TI-89 Titanium / Voyage™ 200

Erste Schritte

- Ein/Aus
- Eine Anwendung auswählen
- Grafische Darstellung einer Funktion
- Menüs
- Modi
- Zahlen eingeben

Erstellen...

- Tabellen
- Matrizen
- Statistik und Datenplots
- Split Screens

Für Fortgeschrittene

- Aktivitäten
- Funktions und Anweisurngen
- Archivieren
- Programmieren

Weitere Informationen

- Rechnerkopplung
- Zahlensysteme
- Texteditor
- Kundendienst und Service

© 2004 Texas Instruments

Wichtig

Texas Instruments übernimmt keine Gewährleistung, weder ausdrücklich noch stillschweigend, einschließlich, aber nicht beschränkt auf implizierte Gewährleistungen bezüglich der handelsüblichen Brauchbarkeit und Geeignetheit für einen speziellen Zweck, was sich auch auf die Programme und Handbücher bezieht, die ohne eine weitere Form der Gewährleistung zur Verfügung gestellt werden.

In keinem Fall haftet Texas Instruments für spezielle, begleitende oder zufällige Beschädigungen in Verbindung mit dem Kauf oder der Verwendung dieser Materialien. Die einzige und ausschließliche Haftung von Texas Instruments übersteigt unabhängig von ihrer Art nicht den geltenden Kaufpreis des Gegenstandes bzw. des Materials. Darüber hinaus übernimmt Texas Instruments keine Haftung gegenüber Ansprüchen Dritter.

© 2004 Texas Instruments Incorporated
Windows, Macintosh sind Warenzeichen der jeweiligen Eigentümer.

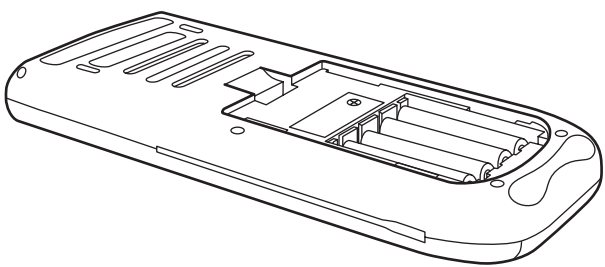
Erste Schritte

In Gang kommen

Einlegen der AAA Batterien

Der TI-89 Titanium verwendet vier AAA Alkali-Batterien und eine Silberoxid-Batterie (SR44SW oder 303) als Backup. Der Voyage™ 200 -verwendet vier AAA Alkali-Batterien und eine Lithium-Batterie als Backup (CR1616 oder CR1620). Die Backup-Batterien sind bereits eingesetzt, die AAA Batterien werden mit den Geräten geliefert.

1. Nehmen Sie die Batteriefachabdeckung an der Rückseite des Handhelds ab.
2. Packen Sie die vier AAA Batterien, die mit Ihrem Gerät geliefert wurden, aus und legen Sie sie in das Batteriefach ein. Ordnen Sie die Batterien im Batteriefach gemäß der schematischen Darstellung der Polarität (+ und –) an.



3. Bringen Sie die Batteriefachabdeckung des Handhelds wieder an. Sie sollte einrasten.

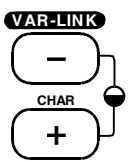
Drücken Sie nach dem Einlegen der zum Gerät mitgelieferten Batterien [ON]. Die Apps-Arbeitsfläche wird eingeblendet.

Hinweis: Wenn Ihr Handheld die vorinstallierten Apps initialisiert, wird ein Fortschrittsbalken mit der Meldung "`Installation in progress . . . Do not interrupt!`" an Stelle der Apps-Arbeitsfläche eingeblendet. Entnehmen Sie während der Initialisierung nicht die Batterien, um zu vermeiden, dass Apps verloren gehen. (Sie können Apps von der Ressource-CD-ROM oder von education.ti.com neu installieren.)

Fortschrittsbalken

Anpassen des Kontrasts

- Zum heller stellen der Anzeige drücken und halten Sie [♦] und tippen auf [−].
- Zum dunkler stellen der Anzeige drücken und halten Sie [♦] und tippen auf [+].

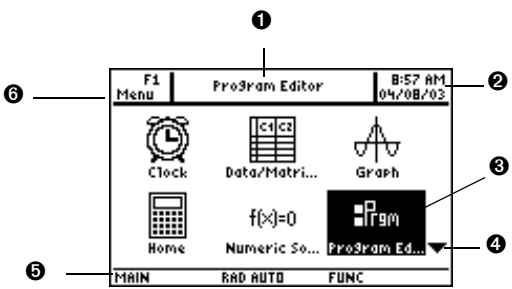


Die Apps-Arbeitsfläche

Die Apps-Arbeitsfläche ist der Ausgangspunkt für die Bedienung Ihres TI-89 Titanium oder Voyage™ 200. Die installierten Apps erscheinen auf dem Desktop als Symbole, die

für einfachen Zugriff in Kategorien organisiert sind. Auf der Apps-Arbeitsfläche können Sie:

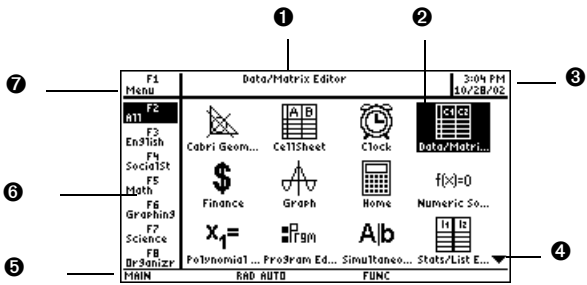
- Apps öffnen.
- Kategorien von Apps auswählen und bearbeiten.
- Alle installierten Apps auf Ihrem Handheld anzeigen.
- Den vollständigen Namen der markierten App anzeigen.
- Zeit und Datum anzeigen und bearbeiten.
- Die Informationen in der Statuszeile prüfen.
- Split-Screen-Modus-Informationen anzeigen.



TI-89 Titanium Apps-Arbeitsfläche

❶ Vollständiger Name der markierten App.
❷ Zeit und Datum.
❸ Drüc‘ken Sie [ENTER], um die markierte App zu öffnen.
❹ Blättern Sie nach unten, um weitere Apps anzuzeigen.
❺ Die Informationen der Statuszeile.
❻ Bearbeiten der Kategorien.

Hinweis: Da das Display des TI-89 Titanium kleiner ist als das des Voyage™ 200, unterscheidet sich die Apps-Arbeitsfläche auf beiden Gerätetypen leicht. Die Liste der Apps-Kategorien, die auf der linken Seite des Voyage™ 200 Desktops angezeigt wird, ist auf dem TI-89 Titanium Desktop nicht vorhanden, Kategorien werden jedoch bei beiden Gerätetypen gleich ausgewählt.



Voyage™ 200 Apps desktop

❶ Vollständiger Name der markierten App.
❷ Drücken Sie [ENTER], um die markierte App zu öffnen.
❸ Zeit und Datum.
❹ Blättern Sie nach unten, um weitere Apps anzuzeigen.
❺ Die Informationen der Statuszeile.
❻ Kategorie der Apps auswählen.
❼ Bearbeiten der Kategorien

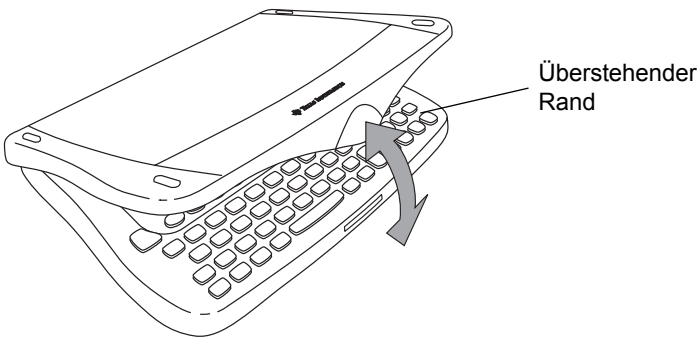
Um jederzeit zur Apps-Arbeitsfläche zurückzukehren, drücken Sie [APPS]. Die letzte gewählte Kategorie und die zuletzt geöffnete App erscheinen markiert.

Entfernen und Aufsetzen der Abdeckung (Voyage™ 200)

Entfernen der Abdeckung:

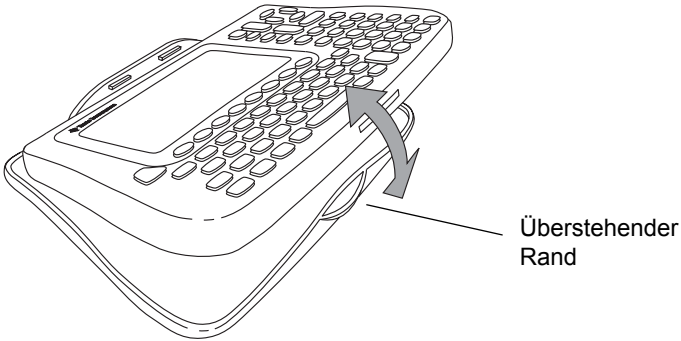
1. Halten Sie das Gerät mit einer Hand fest.
2. Fassen Sie den Deckel mit der anderen Hand am überstehenden Rand.
3. Öffnen Sie den Deckel vom Rand aus.

Beim Aufsetzen des Deckels positionieren Sie ihn mit dem überstehenden Rand an der Vorderseite des Handhelds und rasten ihn wieder ein.



Verstauen des Deckels

Zum Verstauen des Deckels, legen Sie ihn mit dem überstehenden Rand nach vorn umgekehrt unter den Handheld und rasten ihn ein.



Abschalten des Handhelds

Drücken Sie [2nd] [OFF]. Beim nächsten Einschalten des Handhelds liegt die Apps-Arbeitsfläche mit den selben Einstellungen und Speicherinhalten vor. (Wenn Sie die Apps-Arbeitsfläche deaktiviert haben, erscheint der Hauptbildschirm.)

Sie können wahlweise eine der folgenden Tasten benutzen um den TI-89 Titanium / Voyage™ 200 abzuschalten.

Drücken Sie:	Beschreibung
[2nd] [OFF] (Drücken Sie [2nd] und anschließend [OFF])	Einstellungen und Speicherinhalt werden durch die Funktion Constant Memory™ beibehalten. • Sie können [2nd] [OFF] jedoch nicht verwenden, wenn eine Fehlermeldung angezeigt wird. Wenn Sie den TI-89 Titanium / Voyage™ 200 wieder einschalten wird unabhängig von der zuletzt verwendeten Anwendung entweder der Hauptbildschirm oder die Apps-Arbeitsfläche angezeigt.
[♦] [OFF] (Drücken Sie [♦] und anschließend [OFF])	Gleiche Wirkung wie [2nd] [OFF] außer: • Sie können [♦] [OFF] verwenden, wenn eine Fehlermeldung angezeigt wird. • Wenn Sie den TI-89 Titanium / Voyage™ 200 wieder einschalten, entspricht Alles dem Zustand vor dem Ausschalten.

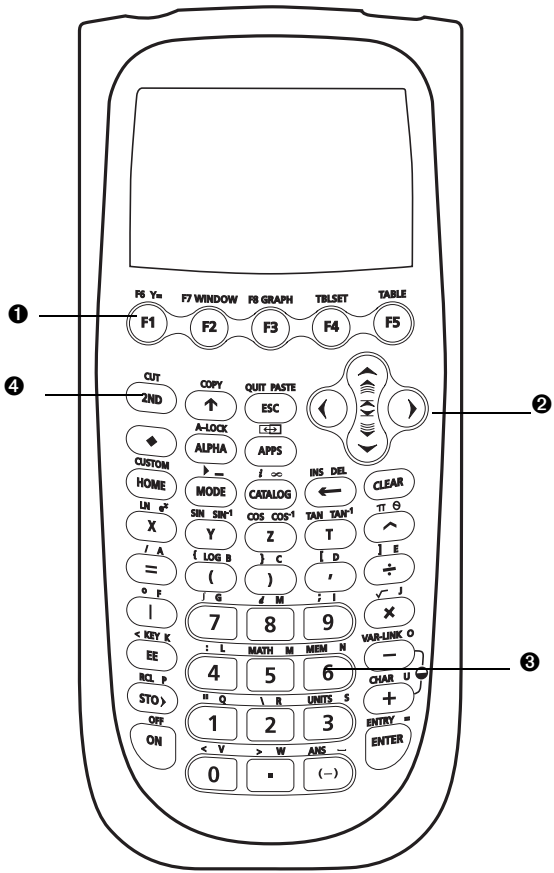
Hinweis: [OFF] ist die sekundäre Funktion der Taste [ON].

Das Leistungsmerkmal Automatic Power Down™ (APD™) des Handhelds verlängert die Lebensdauer der Batterien, indem es den Handheld nach einigen Minuten ohne Aktivitäten automatisch abschaltet. Wenn Sie den Handheld nach APD einschalten:

- Anzeige, Cursor und alle Fehlerzustände sind genau wie vor dem APD.
- Alle Einstellungen und Speicherinhalte bleiben erhalten.

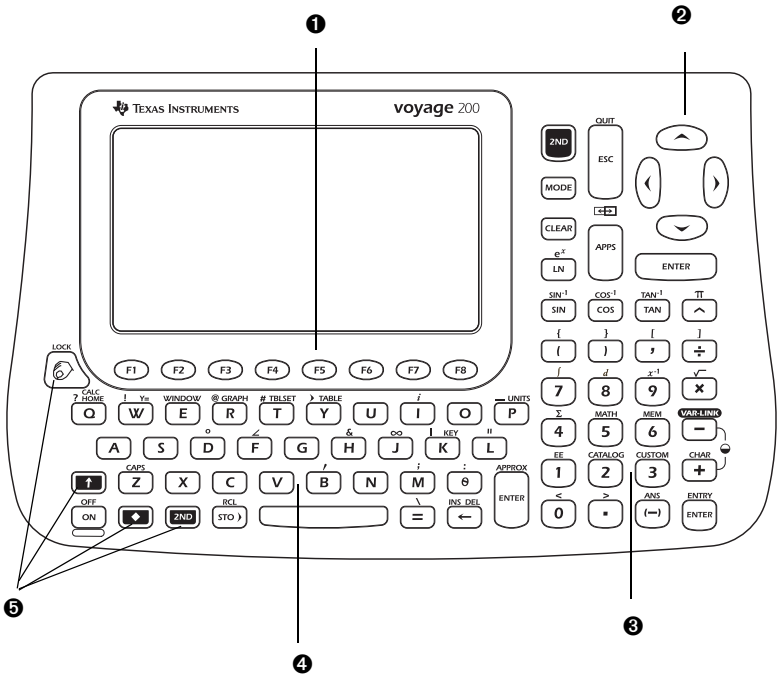
Hinweis: APD funktioniert nicht, wenn eine Berechnung oder ein Programm ausgeführt wird, es sei denn, in der Berechnung bzw. dem Programm wurde eine Pause vorgegeben.

Tasten von TI-89 Titanium und Voyage™ 200



Tasten des TI-89 Titanium

❶ Funktionstasten (F1– F8) öffnen Werkzeugleisten-Menüs und Apps und dienen zum Bearbeiten von Kategorien von Apps.
❷ Cursortasten (◀, ▶, ▲, ▼) steuern den Cursor.
❸ Numerische Tastatur für mathematische und wissenschaftliche Funktionen.
❹ Modifikatortasten (2nd, ♦, ⬆) dienen zum Aufrufen sekundärer und zusätzlicher Funktionen von Tasten.



❶ Funktionstasten (F1 – F8) öffnen Werkzeugleisten-Menüs und Apps und dienen zum Bearbeiten von Kategorien von Apps.
❷ Cursortasten (◀, ▶, ▲, ▼) steuern den Cursor.
❸ Numerische Tastatur für mathematische und wissenschaftliche Funktionen.
❹ QWERTY-Tastatur; ähnlich einer Computertastatur.
❺ Modifikatortasten (2nd, ♦, ⇧, ⎋) dienen zum Aufrufen sekundärer und zusätzlicher Funktionen von Tasten.

QWERTY-Tastatur (nur beim Voyage™ 200)

Wenn Sie mit dem Schreiben am Computer vertraut sind, wird Ihnen die Nutzung der Voyage™ 200 QWERTY-Tastatur nicht schwer fallen. Es gibt aber folgende Unterschiede:

- Zur Eingabe eines einzelnen Großbuchstabens drücken Sie ⇧ und die Taste des Buchstabens.
- Zum Aktivieren des Großbuchstaben-Feststellers drücken Sie 2nd [CAPS]. Zum Deaktivieren des Großbuchstaben-Feststellers drücken Sie 2nd [CAPS] noch einmal.

Eingeben von Sonderzeichen

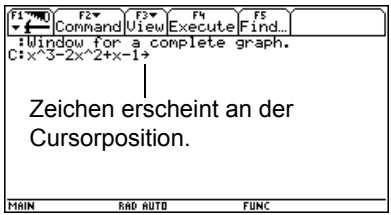
Verwenden Sie zur Eingabe von Sonderzeichen das Menü CHAR (ZEICHEN) und Tastaturbefehle. Über das CHAR (ZEICHEN)-Menü können Sie griechische, mathematische, internationale und andere Sonderzeichen eingeben. Eine Bildschirmdarstellung der Tastatur zeigt die Tastenkürzel zur Eingabe weiterer häufig verwendeter Zeichen.

So wählen Sie Buchstaben aus dem CHAR (ZEICHEN)-Menü:

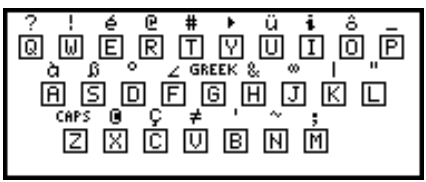
1. Drücken Sie [2nd] [CHAR]. Das CHAR (ZEICHEN)-Menü wird angezeigt.
2. Wählen Sie mit den Cursortasten eine Kategorie. Ein Untermenü führt die Zeichen dieser Kategorie auf.
3. Wählen Sie ein Zeichen mit den Cursortasten und drücken Sie [ENTER].

Beispiel*:* Geben Sie das Zeichen Pfeil-rechts (→) in den Text Editor ein.

Drücken Sie	Ergebnis
[2nd] [CHAR]	
4	

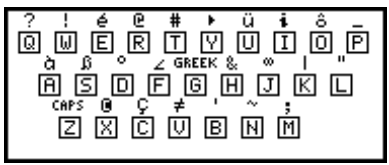
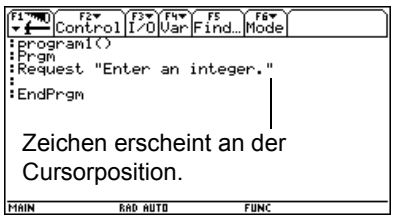
Drücken Sie	Ergebnis
9 – oder – Drücken Sie wiederholt ⊙, um 9:→ auszuwählen und drücken Sie dann ENTER	

Zum Öffnen der Tastaturansicht drücken Sie ♦ [KEY]. Die Tastenbelegung wird eingeblendet.

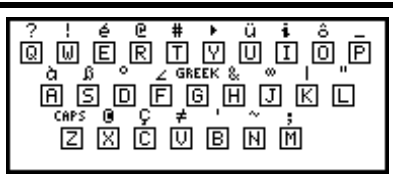


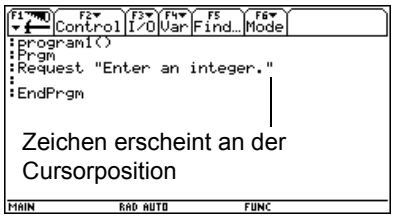
Zur Eingabe der meisten Zeichen, drücken Sie 2nd und die zugehörige Taste. Drücken Sie zum Verlassen der Tastaturansicht ESC.

TI-89 Titanium Beispiel: Verwenden Sie die Tastenbelegungsanzeige, um die Tastenkombination für das Symbol“nicht gleich” (≠) zu bestimmen und das Symbol im Programmeditor einzugeben.

Drücken Sie	Ergebnis
♦ [KEY]	
♦ =	

Voyage™ 200 Beispiel: Verwenden Sie die Tastenbelegungsanzeige, um die Tastenkombination für Anführungszeichen (") zu finden und das Zeichen im Programmeditor einzugeben..

Drücken Sie	Ergebnis
[KEY]	

Drücken Sie	Ergebnis
2nd L	

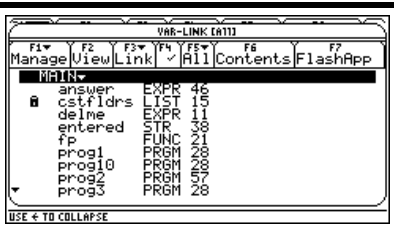
Modifikatortasten

Modifikatortasten fügen Funktionen hinzu, indem Sie die verfügbare Anzahl der Tastaturoperationen vergrößern. Um eine Modifikatorfunktion auszuführen, drücken Sie eine Modifikatortaste und dann die Taste für die zugehörige Operation.

Tasten	Beschreibung
2nd (Zweite)	Zugriffe auf Apps, Menü-Optionen und andere Operationen. Die Zweitfunktionen sind in der gleichen Farbe der Taste 2nd über diese gedruckt.
♦ (Karo)	Zugriffe auf Apps, Menü-Optionen und andere Operationen. Karofunktionen sind in der gleichen Farbe, wie die ♦-Taste über die zugehörigen Tasten gedruckt.
⇧ (Umschalt)	Gibt einen Großbuchstaben für den nächsten eingegebenen Buchstaben ein. Wird mit ◀ und ▶ zum Markieren von Buchstaben bei der Bearbeitung verwendet.

Tasten	Beschreibung
alpha (Alpha; nur beim TI-89 Titanium)	Dient zur Eingabe von Schriftzeichen ohne QWERTY-Tastatur. Das jeweilige Alpha-Zeichen ist über der zugehörigen Taste in der Farbe der Taste alpha angegeben.
(Hand; nur beim Voyage™ 200)	Dient dazu, mit den Cursortasten Geometrieobjekte zu manipulieren. Wird auch zum Zeichnen einer Grafik verwendet.

Beispiel: Gehen Sie in den VAR-LINK Bildschirm, in dem Sie Variablen und Apps verwalten können.

Drücken Sie	Ergebnis
2nd [VAR-LINK]	

Funktionstasten

Verwenden Sie die Funktionstasten um folgende Operationen durchzuführen:

- Auf der Apps-Arbeitsfläche öffnen Sie Apps und wählen oder bearbeiten die Apps-Kategorien.

- Im Hauptbildschirm des Rechners öffnen Sie Menüleisten Menüs, um mathematische Operationen auszuwählen.
- Innerhalb von Apps öffnen Sie Menüleisten Menüs, um Optionen der App auszuwählen.

Cursortasten

Drücken von ◀, ▶, ▲, oder ▼ bewegt den Cursor in der entsprechenden Richtung. Je nach App und abhängig davon, ob die Modifikatortaste [2nd] oder [♦] verwendet wird, steuern die Cursortasten den Cursor in bestimmter Weise.

- ▲ oder ▼ bewegt den Cursor jeweils eine Zeile nach oben oder unten.
- [2nd] ◀ oder [2nd] ▶ bewegt den Cursor zum Anfang bzw. Ende einer Zeile.
- [2nd] ▲ oder [2nd] ▼ bewegt den Cursor jeweils eine Seite nach oben oder unten.
- [♦] ▲ oder [♦] ▼ bewegt den Cursor zum Anfang bzw. Ende der Seite.
- ▲ und ◀, ▲ und ▶, ▼ und ◀, oder ▼ und ▶ bewegt den Cursor diagonal. (Drücken Sie die angegebenen Cursortastenpaare gleichzeitig.)

Numerische Tastatur

Mit der numerischen Tastatur können Sie positive und negative Zahlen eingeben.

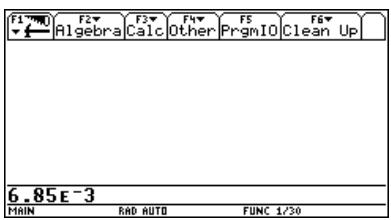
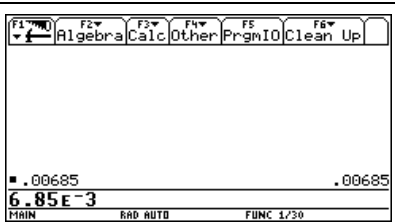
Um eine negative Zahl einzugeben, drücken Sie [(-)] vor Eingabe der Ziffern.

Hinweis: Verwechseln Sie die Vorzeichentaste ([(-)]) nicht mit der Subtraktionstaste ([−]).

Zur Eingabe einer Zahl in wissenschaftlicher Notation:

1. Geben Sie die Zahl (Mantisse) vor dem Exponenten ein. (Dieser Wert kann ein Ausdruck sein.)
2. Drücken Sie [2nd] [EE]. Das Exponentensymbol (E) folgt den eingegebenen Ziffern.
3. Geben Sie den Exponenten als ganze Zahl mit bis zu drei Stellen ein. (Wie das folgende Beispiel zeigt, können Sie auch einen negativen Exponenten verwenden.)

Beispiel*:* Geben Sie auf dem Hauptbildschirm des Rechners 0.00685 in wissenschaftlicher Notation ein.

Drücken Sie	Ergebnis
6 [.] 8 5 [2nd] [EE] [(-)] 3	
[ENTER]	

Weitere wichtige Tasten

Tastenbefehle	Beschreibung
♦ [Y=] nur beim TI-89 Titanium	Zeigt den Y= Editor an.
♦ [WINDOW] nur beim TI-89 Titanium	Zeigt den Window Editor an.
♦ [WINDOW] nur beim TI-89 Titanium	Zeigt den Graphbildschirm an.
♦ [GRAPH] nur beim TI-89 Titanium	Legt Parameter für den Tabellenbildschirm fest.
♦ [WINDOW] nur beim TI-89 Titanium	Zeigt den Tabellenbildschirm an.
♦ TI-89 Titanium: ♦ [CUT] ♦ [COPY] ♦ [PASTE] Voyage™ 200: ♦ X (Ausschneiden) ♦ C (Kopieren) ♦ V (Einfügen)	Mit diesen Tasten können Sie eingegebene Daten durch die Funktionen Ausschneiden, Kopieren und Einfügen bearbeiten.

Tastenbefehle	Beschreibung
[♦] S nur beim Voyage™ 200	Das Dialogfeld SAVE COPY AS (KOPIE SPEICH ALS) wird angezeigt und fordert zur Auswahl eines Verzeichnisses und Eingabe eines Variablennamens auf, wo die auf dem Bildschirm eingegebenen Daten gespeichert werden.
[♦] N nur beim Voyage™ 200	Erstellt eine neue Datei.
[♦] O nur beim Voyage™ 200	Öffnet eine vorhandene Datei, die Sie angeben.
[♦] F nur beim Voyage™ 200	Zeigt das Dialogfeld Formats (Formate) oder Graph Formats (Graphik Formate) an, wo Sie Format-Informationen für die aktive App eingeben.
[APPS]	Anzeigen der Apps-Arbeitsfläche.
[♦] [APPS]	Bei deaktivierter Apps-Arbeitsfläche wird das Menü FLASH APPLICATIONS angezeigt.
[2nd] [⇆]	Schaltet zwischen den beiden zuletzt ausgewählten Apps um.
[2nd] [CUSTOM]	Schaltet das anwenderspezifische Menü ein und aus.
[2nd] [▶]	Konvertiert Maßeinheiten.
TI-89 Titanium: [♦] [_] Voyage™ 200: [2nd] [_]	Kennzeichnet eine Maßeinheit.

Tastenbefehle	Beschreibung
[←]	Löscht das Zeichen links neben dem Cursor (Rücktaste).
[♦] [DEL]	Löscht das Zeichen rechts neben dem Cursor.
[2nd] [INS]	Schaltet zwischen dem Einfüge- und Überschreibmodus um.
[2nd] [MEM]	Zeigt den Memory (Speicher) Bildschirm
TI-89 Titanium: [CATALOG] Voyage™ 200: [2nd] [CATALOG]	Zeigt eine Befehlsliste an.
[2nd] [RCL]	Holt den Inhalt einer Variablen zurück.
[STO▸]	Speichert einen Wert in einer Variablen.
[2nd] [CHAR]	Zeigt das Menü Char (Zeichen) an, in dem Sie griechische Buchstaben, internationale akzentuierte Zeichen und andere Sonderzeichen auswählen können.
[2nd] [QUIT]	• Im Vollbild-Modus wird die Apps-Arbeitsfläche angezeigt. • Im Split-Screen-Modus wird die Vollbildanzeige der aktiven App angezeigt. • Bei deaktivierter Apps-Arbeitsfläche wird der Hauptbildschirm des Rechners angezeigt.

Moduseinstellungen

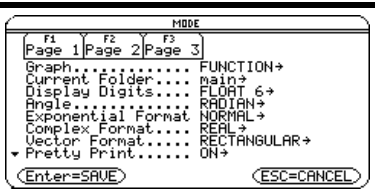
Durch die Modi wird festgelegt, wie der TI-89 Titanium oder Voyage™ 200 Informationen anzeigt und interpretiert. Alle Zahlen, einschließlich der Elemente von Matrizen und Listen werden den aktuellen Moduseinstellungen entsprechend angezeigt. Wenn der TI-89 Titanium oder Voyage™ 200 abgeschaltet wird, werden durch das Leistungsmerkmal Constant Memory™ alle ausgewählten Moduseinstellungen gesichert.

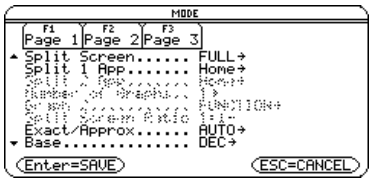
Anzeigen der Moduseinstellungen des TI-89 Titanium oder Voyage™ 200:

1. Drücken Sie MODE. Die Seite 1 des Dialogfelds MODE (MODUS) wird angezeigt.
2. Drücken Sie F2 oder F3, um eine Anzeige der auf Seite 2 oder 3 aufgeführten Modi zu erhalten.

Hinweis: Grau dargestellte Modi sind nur verfügbar, wenn weitere erforderliche Moduseinstellungen gewählt wurden. Beispielsweise ist der Modus Custom Units (Benutzer Einh.) auf Seite 3 nur verfügbar, wenn der Modus Unit System (EinhSystem) auf Custom (Eigene) gesetzt ist.

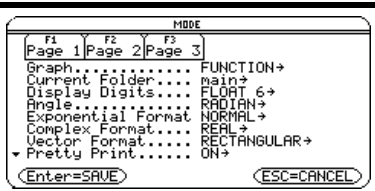
Anzeigemodus-Einstellungen

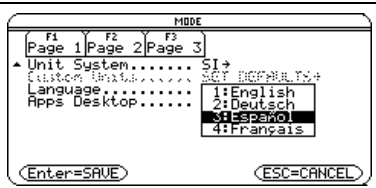
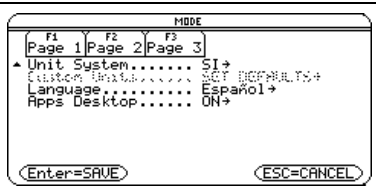
Drücken Sie	Ergebnis
MODE	

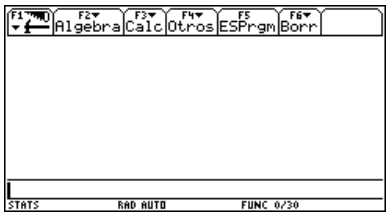
Drücken Sie	Ergebnis
F2	
F3	

Änderung der Moduseinstellungen

Beispiel*:* Ändern des Language (Sprache)-Modus auf Spanish (*Español*).

Drücken Sie	Ergebnis
MODE	

Drücken Sie	Ergebnis
F3	
Blättern Sie nach unten zum Feld Language (Sprache). ▼	
Drücken Sie ▶ und anschließend ▼, bis **3:Español** markiert ist. **Hinweis:** Ihre Menüliste kann in Abhängigkeit von den installierten Sprachen anders aussehen.	
ENTER	

Drücken Sie	Ergebnis
ENTER **Hinweis:** Die zuletzt geöffnete App wird angezeigt (in diesem Beispiel der Hauptbildschirm des Rechners).	

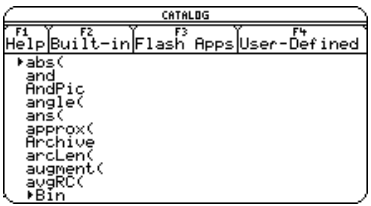
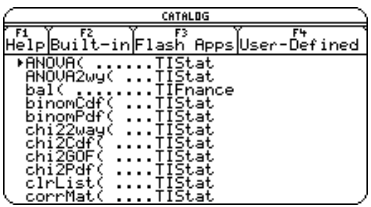
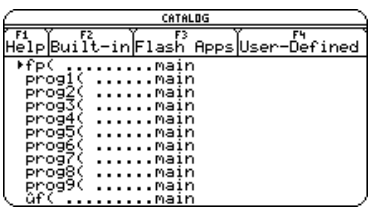
Zur Rückstellung des Language (Sprache) Modus auf English, wiederholen Sie die Schritte und wählen im Language (Sprache) Modusfeld **1:English**.

Verwenden des Catalog (Katalog) für den Befehlszugriff

2nd Mit dem Catalog (Katalog) haben Sie Zugriff auf eine Liste von TI-89 Titanium oder Voyage™ 200-Befehlen, inklusive Funktionen, Anweisungen und benutzerdefinierten Programmen. Befehle werden in alphabetischer Reihenfolge aufgeführt. Befehle, die nicht mit einem Buchstaben beginnen, finden Sie am Ende der Liste (&, /, +, -, usw.).

Momentan nicht verfügbare Optionen erscheinen grau. Beispielsweise wird die Menü-Option Flash Apps (F3) grau dargestellt, wenn keine Flash-Anwendungen auf Ihrem TI-89 Titanium oder Voyage™ 200 installiert sind; die Menü-Option UserDefined (Benutzer) (F4) wird grau dargestellt, wenn Sie noch keine Funktion und kein Programm erstellt haben.

Hinweis: Durch Eingabe eines Buchstabens gelangen Sie zum ersten Befehl in der Liste der mit diesem Buchstaben beginnt.

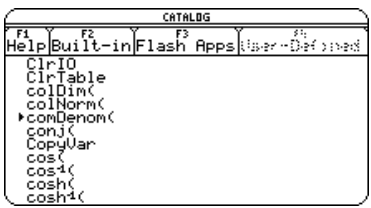
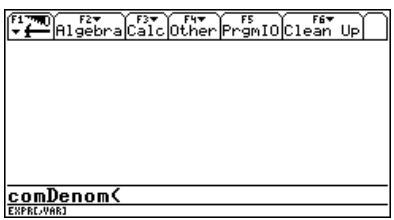
Drücken Sie	Ergebnis
2nd [CATALOG] (zeigt die Built in (Vorhandenen) Befehle an)	
F3 (zeigt die Flash Apps-Befehle an, falls vorhanden)	
F4 (zeigt die User Defined-(Benutzer) Befehle an, falls vorhanden)	

Wählen Sie Befehle aus dem Catalog (Katalog) und fügen Sie sie in die Eingabezeile des Hauptbildschirms des Rechners, oder in andere Apps, wie den Y= Editor, Text Editor oder CellSheet™ Apps ein.

Beispiel: Einfügen des **comDenom((gemNenn()** Befehls in die Eingabezeile des Hauptbildschirms des Rechners.

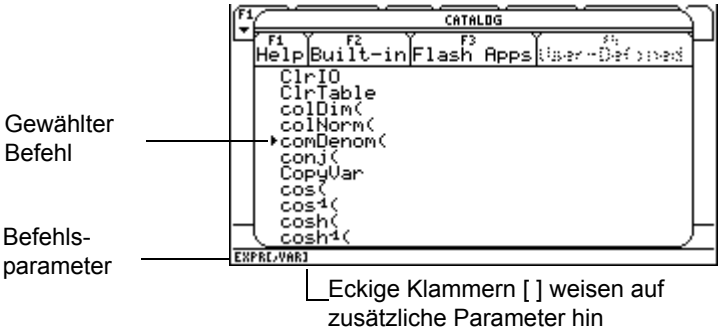
Hinweis: Positionieren Sie den Cursor vor der Auswahl eines Befehls an der Stelle, an welcher der Befehl erscheinen soll.

Durch Drücken von 2nd ⊙ gelangen Sie zur jeweils nächsten Seite der Katalog-Liste.

Drücken Sie	Ergebnis
TI-89 Titanium: CATALOG Voyage™ 200: 2nd [CATALOG] **C** 2nd ⊙ ⊙ ⊙ ⊙ ⊙	
ENTER	

In der Statuszeile werden alle notwendigen und optionalen Parameter für den gewählten Befehl angezeigt. Optionale Parameter erscheinen in rechteckigen Klammern.

Hinweis: Durch Drücken von F1 werden die Parameter für den gewählten Befehl ebenso angezeigt.



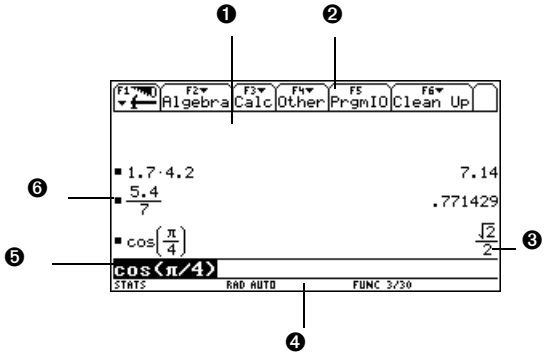
Drücken Sie zum Verlassen des Catalog (Katalog) ohne Auswahl eines Befehls [ESC].

Hauptbildschirm des Rechners

Der Hauptbildschirm ist der Ausgangspunkt des Rechners für mathematische Operationen, inklusive der Ausführung von Anweisungen, dem Auswerten von Ausdrücken und der Anzeige von Ergebnissen.

Drücken Sie zur Anzeige des Hauptbildschirms [♦] [CALC HOME].

Sie gelangen ebenfalls zum Hauptbildschirm des Rechners, wenn Sie auf die Apps-Arbeitsfläche das Symbol Home (Haupt) auswählen und [ENTER] wählen.



❶ Im Protokollbereich werden die eingegebenen Eingabe/Antwort-Paare aufgeführt.

❷ Register zeigen Menüs an, um aus Listen mit Operationen auswählen zu können. Drücken Sie F1, F2 usw., um Menüs anzuzeigen.

❸ Das Ergebnis der letzten Eingabe wird hier angezeigt. (Beachten Sie, dass Ergebnisse nicht in der Eingabezeile angezeigt werden.)

❹ Die Statuszeile zeigt den aktuellen Status des TI-89 Titanium oder Voyage™ 200 an.

❺ Die Eingabezeile zeigt Ihre aktuelle Eingabe an.

❻ Ihre letzte Eingabe wird hier angezeigt.

Sie gelangen zur Apps-Arbeitsfläche zurück, indem Sie im Hauptbildschirm APPS drücken.

Über den Protokollbereich

Je nach Komplexität und Höhe der Ausdrücke zeigt der Protokollbereich bis zu acht Eingabe/Antwort-Paare an. Wenn der Bildschirm voll ist, werden Informationen nach oben aus der Anzeige geschoben. Nutzen Sie den Protokollbereich zum:

- Überarbeiten vorheriger Eingaben und Antworten. Verwenden Sie die Cursortasten, um Eingabe-/Antwort-Paare zu betrachten, die aus dem Bildschirm geschoben wurden.

Holen Sie frühere Eingaben oder Antworten in die Eingabezeile zum nochmaligen Gebrauch oder zum Editieren zurück, oder fügen Sie sie automatisch ein. (Weitere Informationen finden Sie im elektronisch gespeicherten TI-89 Titanium oder Voyage™ 200 Modul - *Operating the Calculator.*)

Der Cursor, der sich normalerweise in der Eingabezeile befindet, kann in den Protokollbereich bewegt werden. In der folgenden Tabelle wird gezeigt, wie Sie den Cursor im Protokollbereich bewegen.

Um...	**Gehen Sie wie folgt vor**
Eingaben/Antworten anzuzeigen, die aus dem Bildschirm geschoben wurden.	Ausgehend von der Eingabezeile drücken Sie ▲, um die letzte Antwort zu markieren. Verwenden Sie ▲ weiterhin, um sich von Antwort zu Eingabe durch den Protokollbereich zu bewegen.
Zum ältesten oder aktuellsten Eingabe/Antwort-Paar wechseln	Wenn sich der Cursor im Protokollbereich befindet, drücken Sie ♦ ▲ oder ♦ ▼.

Um...	Gehen Sie wie folgt vor
Anzeigen einer Eingabe oder einer Antwort, die länger als eine Zeile ist (▶ wird am Ende der Zeile angezeigt)	Bewegen Sie den Cursor zur Eingabe oder zur Antwort. Mit ◀ oder ▶ blättern Sie nach links oder rechts und mit 2nd ◀ oder 2nd ▶ gelangen Sie zum Anfang oder Ende.
Den Cursor in die Eingabezeile zurückbringen	Drücken Sie ESC, oder drücken Sie ▼, bis der Cursor wieder in der Eingabezeile angekommen ist.

Interpretieren der Protokollinformationen in der Statuszeile

Mit Hilfe der Protokollanzeige in der Statuszeile erhalten Sie Informationen zu den Eingabe/Antwort-Paaren. Zum Beispiel:

Wenn sich der Cursor an der Eingabezeile befindet:

Gesamtzahl der momentan gespeicherten Paare _____ 8/30 _____ Maximale Anzahl der speicherbaren Paare

Wenn sich der Cursor im Protokollbereich befindet:

Paarnummer des markierten Eingabe/Antwort-Paares _____ 8/30 _____ Gesamtzahl der momentan gespeicherten Paare

Bearbeiten des Protokollbereichs

Ändern der Anzahl der speicherbaren Paare:

1. Drücken Sie im Hauptbildschirm des Rechners F1 und wählen Sie **9:Format.**
2. Drücken Sie ▶ und markieren Sie die neue Anzahl mit ▲ oder ▼.
3. Drücken Sie ENTER ENTER.

Löschen des Protokollbereichs und aller gespeicherten Paare:

- Drücken Sie im Hauptbildschirm des Rechners F1 und wählen Sie **8:Clear Home (8:LöscheBSchirm)**.

  – oder –

- Geben Sie in der Eingabezeile des Hauptbildschirms **ClrHome** ein.

Bewegen Sie den Cursor zum Löschen eines Eingabe/Antwort-Paares zur Eingabe oder zur Antwort und drücken Sie ← oder CLEAR.

Arbeiten mit Apps

Der TI-89 Titanium oder Voyage™ 200 organisiert Apps auf der Apps-Arbeitsfläche nach Kategorien. Drücken Sie zur Auswahl einer Kategorie eine Funktionstaste (F2 bis 2nd F8 oder F2 bis F8). Die App-Symbole der gewählten Kategorie werden auf der Apps-Arbeitsfläche angezeigt.

Hinweis: Wenn der Name unter einem Apps-Arbeitsfläche-Symbol verkürzt wird, markieren Sie es mit den Cursortasten. Jetzt wird der vollständige Name oben auf der Apps-Arbeitsfläche angezeigt.

Öffnen von Apps

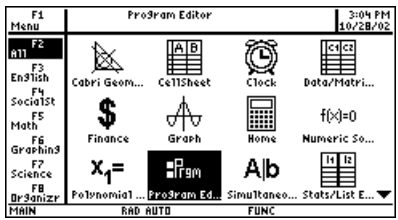
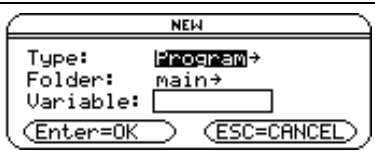
Verwenden Sie die Cursortasten um das Apps-Symbol auf der Arbeitsfläche zu markieren und drücken Sie ENTER. Die App wird entweder direkt geöffnet oder es wird ein Dialogfeld angezeigt. Das häufigste Dialogfeld führt diese Optionen für die App auf:

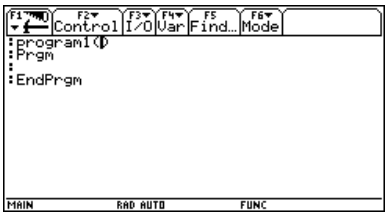
Hinweis: Der TI-89 Titanium oder Voyage™ 200 verwendet den allgemeinen Begriff *Variable* für die App-Dateien, die Sie erstellen.

Option	Beschreibung
Current (Aktuell)	Zeigt den Bildschirm, der angezeigt wurde, als Sie die App zum letzten Mal geöffnet hatten. Wenn keine aktuelle App-Variable vorhanden ist, wird das Dialogfeld New angezeigt.
Open (Öffnen)	Ermöglicht Ihnen eine vorhandene Datei zu öffnen.
New (Neu)	Erstellt eine neue Datei mit dem im Feld angegebenen Namen.

Wählen Sie eine Option, geben Sie die notwendigen Informationen ein und drücken Sie ENTER. Die App wird angezeigt.

Beispiel: Erstellen Sie mit dem Program Editor ein neues Programm.

Drücken Sie	Ergebnis
Markieren Sie mit den Cursortasten prgm Program Ed...	
ENTER	
3	
ENTER	
⊙⊙ p r o g r a m 1	

Drücken Sie	Ergebnis
ENTER ENTER	

Die neu erstellte Programmvariable *program1* ist im Verzeichnis main gespeichert.

Rückkehr von einer App zur Apps-Arbeitsfläche

Drücken Sie APPS. Daraufhin werden die Symbole der zuletzt gewählten Apps-Kategorie auf der Apps-Arbeitsfläche angezeigt, wobei das Symbol der zuletzt geöffneten App markiert erscheint.

Sie können ebenso zur Apps-Arbeitsfläche zurückkehren, indem Sie 2nd [QUIT] im Vollbildmodus drücken. Im Split-Screen-Modus drücken Sie zweimal 2nd [QUIT].

Um zur zuletzt geöffneten App zurück zu kehren, drücken Sie auf der Apps-Arbeitsfläche 2nd [⇔].

Auswählen einer Apps-Kategorie

Auf dem TI-89 Titanium erscheinen die Namen der Apps-Kategorien nur im Menü **F1**. Zur Auswahl einer Apps-Kategorie drücken Sie F1 **2:Select Category**. Verwenden Sie dann die Cursortasten, um eine Apps-Kategorie zu markieren und drücken Sie anschließend ENTER, um die markierte Kategorie auszuwählen. Sie können eine Kategorie auch direkt

über die Tastatur und die zugehörige Tastenkombination auswählen (verwenden Sie bei Bedarf die Taste 2nd). Die App-Symbole für die ausgewählte Kategorie werden auf der Apps-Arbeitsfläche eingeblendet

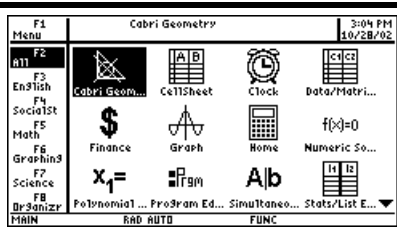
Auf dem Voyage™ 200erscheinen die Namen der Apps-Kategorien links, längs der Apps-Arbeitsfläche. Zur Auswahl einer Apps-Kategorie drücken Sie die zugehörige Funktionstaste (die auf der Apps-Arbeitsfläche oberhalb des Kategorie Namens angezeigt wird).

Die App-Symbole für die ausgewählte Kategorie werden auf der Apps-Arbeitsfläche eingeblendet.

Taste	Beschreibung
F2 All (Alle)	Symbole für alle installierten Apps werden angezeigt. Nicht anpassbar.
F3 English (Englisch)	Anpassbare Kategorie. Die Vorgabe ist English (Englisch).
F4 SocialSt (SozialWiss)	Anpassbare Kategorie. Die Vorgabe ist SocialSt (SozialWiss).
F5 Math (Mathematik)	Anpassbare Kategorie. Die Vorgabe ist Math (Mathematik).
2nd F6 Graphing (Grafik) oder F6 Graphing (Grafik)	Anpassbare Kategorie. Die Vorgabe ist Graphing (Grafik).

Taste	Beschreibung
2nd [F7] Science (Naturwissenschaften) oder F7 Science (Naturwissenschaften)	Anpassbare Kategorie. Die Vorgabe ist Science (Wissenschaft).
2nd [F8] Organizr oder F8 Organizr	Anpassbare Kategorie. Die Vorgabe ist Organizr.

Beispiel: Wählen Sie die Kategorie All (Alle).

Drücken Sie	Ergebnis
F2	

Wenn Sie eine Apps-Kategorie wählen, die keinen Apps enthält, erscheint eine Meldung, die angibt, dass die Kategorie leer ist und verweist Sie auf das F1 **1:Edit Categories**-Menü, wo Sie App-Tastenkürzel zur Kategorie hinzufügen können. (Weitere Informationen zur Anpassung von Apps-Arbeitsflächen-Kategorien finden Sie unten *Anpassen der Apps-Kategorien*).

Drücken Sie ENTER oder ESC, um die Meldung zu löschen und zur Apps-Arbeitsfläche zurückzukehren.

Anpassen der Apps-Kategorien

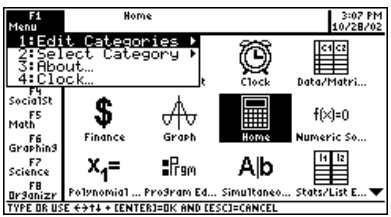
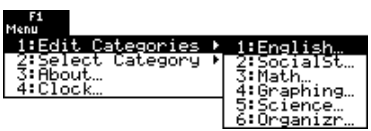
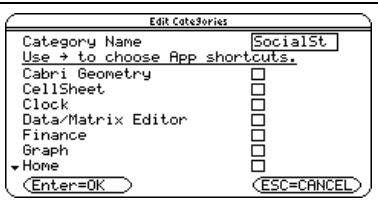
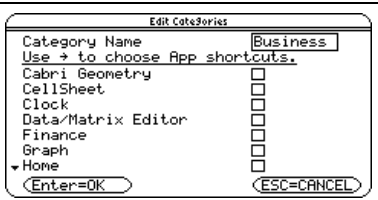
Der TI-89 Titanium oder Voyage™ 200 organisiert Ihre Apps in sieben Kategorien, von denen Sie sechs an Ihre persönlichen Bedürfnisse anpassen können. (Die Kategorie All (Alle) enthält alle installierten Apps und kann nicht bearbeitet werden.)

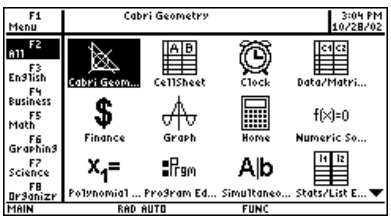
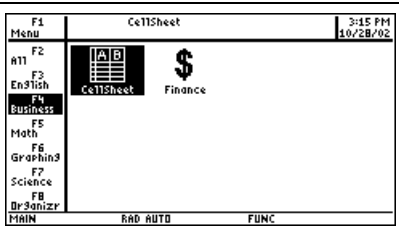
Anpassen der F3 bis 2nd F8 (🖩) oder F8 (⌨) Apps-Kategorien:

1. Wählen Sie F1 **1:Edit Categories**. Die sechs anpassbaren Apps-Kategorienamen werden in einem Untermenü angezeigt. (Die Kategorie All (Alle) wird nicht aufgeführt.)
2. Markieren Sie eine Apps-Kategorie und drücken Sie ENTER. Das Dialogfeld Edit Categories (Kategorien bearbeiten) wird mit einer Liste der installierten Apps und einem Textfeld mit dem hervorgehobenen Kategorienamen angezeigt.
3. Zum Ändern des Apps-Kategorienamens tippen Sie den gewünschten Namen ein.

   Hinweis: Geben Sie einen bis zu acht Zeichen langen Namen ein, der Buchstaben mit oder ohne Groß- und Kleinschreibung, Zahlen, Interpunktionszeichen und Akzentzeichen enthalten kann.
4. Um ein App-Kürzel in einer Kategorie hinzuzufügen oder zu entfernen, drücken Sie ▼ so oft, bis das Feld neben der App ausgewählt ist und drücken Sie dann ▶, um das Häkchen hinzuzufügen oder zu entfernen (✓).
5. Um die Änderungen zu speichern und zur Apps-Arbeitsfläche zurück zu kehren, drücken Sie ENTER.

Beispiel: Ersetzen Sie die Kategorie Social Studies (SozialWiss) durch die Kategorie Business (Geschäft) und fügen Sie die Verweise auf CellSheet™ und die Finance App hinzu.

Drücken Sie	Ergebnis
F1	
⊙	
2 – oder – ⊙ ENTER	
↑ B u s i n e s s	

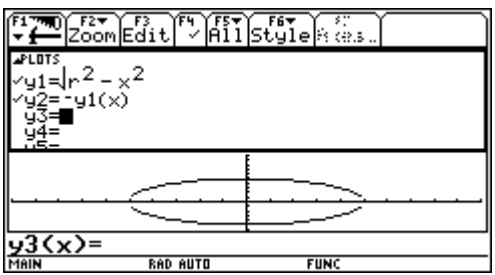
Drücken Sie	Ergebnis
⊖ ⋮ ⊕	
⊖ ⋮ ⊕	
ENTER	
F4	

Öffnen von Apps im Split-Screen-Modus

Ihr TI-89 Titanium oder Voyage™ 200 erlaubt Ihnen die Aufteilung des Bildschirms, so dass Sie zwei Apps gleichzeitig betrachten können. Beispielsweise können Sie den Y= Editor und den Graph-Bildschirm gleichzeitig anzeigen lassen um eine Liste von Funktionen und deren grafische Darstellung zu betrachten.

Wählen Sie den Split Screen-Modus auf Seite 2 des MODE-Bildschirms. Der TI-89 Titanium oder Voyage™ 200 zeigt die gewählten Apps in der Split-Screen-Anzeige wie folgt an. Teilen Sie den Bildschirm horizontal (oben-unten) oder vertikal (links-rechts).

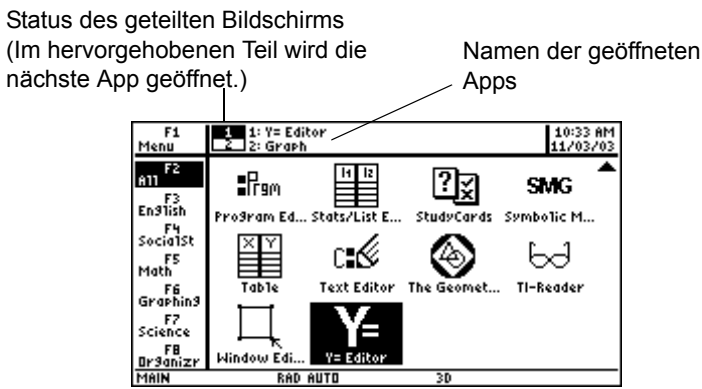
Split-Screen (oben-unten)



Sie gelangen zur Apps-Arbeitsfläche zurück, indem Sie [APPS] drücken. Oberhalb der Apps-Arbeitsfläche wird der Split-Screen-Status mit den Namen der offenen Apps und den Bildschirmabschnitten angezeigt, in denen diese erscheinen. Das Pfeilsymbol (▶) zeigt auf den Bildschirm, in dem die nächste geöffnete App erscheint.

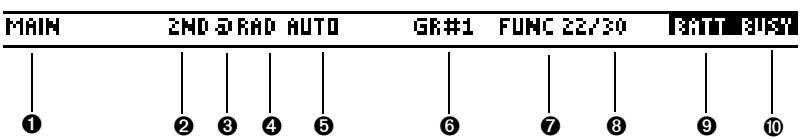
Im Vollbildmodus wird der Split-Screen-Status nicht auf der Apps-Arbeitsfläche angezeigt.

Hinweis: Die Apps-Arbeitsfläche wird immer im Vollbildmodus angezeigt.



Überprüfen der Statusinformationen

In der Statuszeile am unteren Rand des Bildschirms erhalten Sie Informationen über den momentanen Zustand Ihres TI-89 Titanium oder Voyage™ 200.



Anzeige	Bedeutung
❶Aktuelles Verzeichnis	Name des gewählten Verzeichnisses (Main ist das Standard-Verzeichnis).

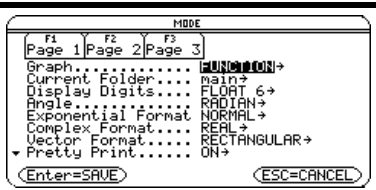
Anzeige	Bedeutung
❷ Modifikatortaste	Gewählte Modifikatortaste (2nd, ♦, ⇧), falls gedrückt.
❸ Hand-Taste (nur beim Voyage™ 200)	Die Modifikatortaste ⇧ wurde ausgewählt (nur beim Voyage™ 200).
❹ Angle (Winkel) Modus	Maßeinheiten, in der Winkelwerte angegeben und interpretiert werden (RAD, DEG (GRD))
❺ Exact/Approx-Modus (Exakt/Approx)	Berechnungs- und Anzeigemodus für Ergebnisse (AUTO, EXACT (EXAKT), APPROX (NÄHERUNG))
❻ Grafiknummer	Aktive von zwei unabhängigen Grafiken im Split-Screen-Modus (GR#1 (GNR1), GR#2 (GNR2))
❼ Graph-Modus	Gewählter Grafiktyp, der angezeigt werden kann (FUNC (FKT), PAR, POL, SEQ (FOLGE), 3D, DE (DGL))
❽ Eingabe-Antwort-Paare	22/30-Anzahl der Eingabe/Antwort-Paare (Vorgabe ist 30, Maximum ist 99) im Protokollbereich des Hauptbildschirms des Rechners.
❾ Batterien ersetzen	Wird angezeigt, wenn die Batterien schwach sind (BATT). Wenn BATT mit einem schwarzen Hintergrund angezeigt wird, sind die Batterien sobald wie möglich zu ersetzen (BATT).

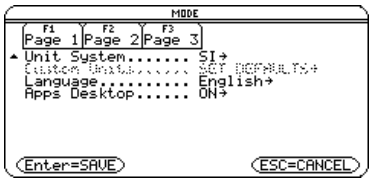
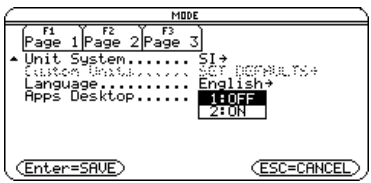
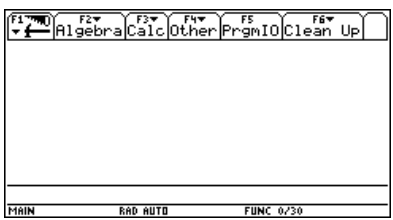
Anzeige	Bedeutung
⓾ Protokoll-Paare, Busy/Pause, Gesperrte Variable	BUSY-Berechnung oder Grafikaufbau wird durchgeführt PAUSE-Sie haben eine Grafik oder ein Programm im Pause-Zustand 🔒-Die im aktuellen Editor geöffnete Variable ist gesperrt oder archiviert und kann nicht verändert werden

Abschalten der Apps-Arbeitsfläche

Sie können die Apps-Arbeitsfläche im Dialogfeld MODE deaktivieren. Wenn Sie das tun, müssen Sie Apps aus dem APPLICATIONS (APPLIKATIONEN) Menü öffnen. Zum Öffnen des Menüs APPLICATIONS (APPLIKATIONEN) drücken Sie APPS.

Beispiel: Deaktivieren Sie die Apps-Arbeitsfläche.

Drücken Sie	Ergebnis
MODE	

Drücken Sie	Ergebnis
F3	
▼ ▼ ▶ ▲	
ENTER ENTER **Hinweis:** Die zuletzt geöffnete App wird angezeigt (in diesem Beispiel der Hauptbildschirm des Rechners).	

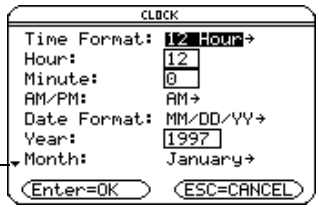
Zum Aktivieren der Apps-Arbeitsfläche wiederholen Sie den Vorgang und wählen im Modusfeld Apps Desktop ON. Um vom Hauptbildschirm des Rechners zur Apps-Arbeitsfläche zurück zu kehren, drücken Sie APPS.

Verwenden der Uhr

Mit Hilfe des Dialogfelds Clock (Uhr) können Sie Datum und Zeit festlegen, das Anzeigeformat der Uhr wählen und diese ein- bzw. ausschalten.

Die Uhr ist per Vorgabe eingeschaltet. Wenn Sie die Uhr deaktivieren, erscheinen alle Einträge des Dialogfelds CLOCK bis auf Clock ON/OFF grau.

▼ weist auf Möglichkeiten des Blätterns zum Ein/Ausschalten der Datums- und Zeitfunktion hin.



Anzeigen des Dialogfelds Clock (Uhr)

1. Mit Hilfe der Cursortasten können Sie das Symbol Clock (Uhr) auf der Apps-Arbeitsfläche auswählen.
2. Drücken Sie ENTER. Das Dialogfeld CLOCK (UHR) wird mit markiertem Feld Time Format (Zeitformat) angezeigt.

Hinweis: Da das Dialogfeld CLOCK (UHR) die aktuellen Einstellungen zum Zeitpunkt des Öffnens anzeigt, kann es notwendig sein, die Zeit vor dem Verlassen zu aktualisieren.

Einstellen der Zeit

1. Drücken Sie ▶, um die Liste mit den Zeitformaten zu öffnen.
2. Drücken Sie ▲ oder ▼, um eine Option zu markieren und drücken Sie dann ENTER. Das ge0wählte Format erscheint im Feld Time Format (Zeitformat).
3. Drücken Sie ▼, um das Feld Hour (Stunde) zu markieren.

4. Geben Sie die Stunde ein und drücken Sie ⊙, um das Feld Minute zu markieren.
5. Geben Sie die Minute(n) ein.
6. Fahren Sie mit Schritt 9 fort, wenn das Zeitformat auf 24 Stunden eingestellt ist.

   – oder –

   Wenn das Zeitformat auf 12 Stunden eingestellt ist, drücken Sie ⊙ um das Feld AM/PM zu wählen.
7. Drücken Sie ⊙, um eine Liste mit den AM/PM-Optionen zu öffnen.
8. Drücken Sie ⊙ oder ⊙, um die Option AM/PM zu markieren und drücken Sie ENTER. Die gewählte Option für AM/PM wird angezeigt.
9. Stellen Sie das Datum ein (siehe auch *Einstellen des Datums*).

   – oder –

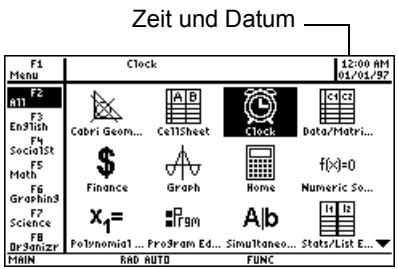
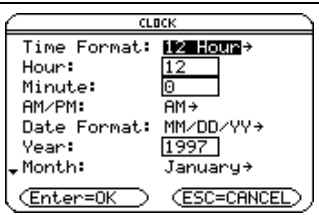
   Drücken Sie zum Speichern und Verlassen des Dialogfelds ENTER. Die Zeit in der oberen rechten Ecke der Apps-Arbeitsfläche wird aktualisiert.

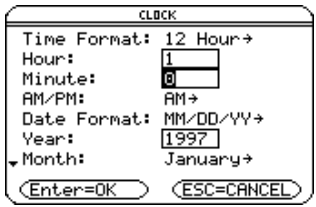
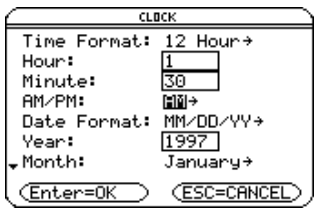
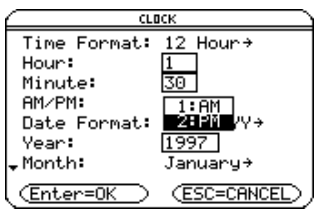
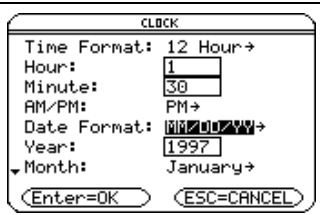
Einstellen des Datums

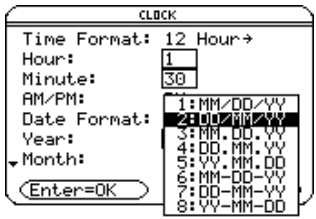
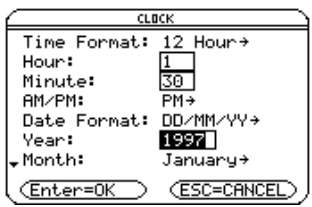
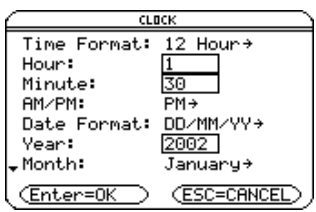
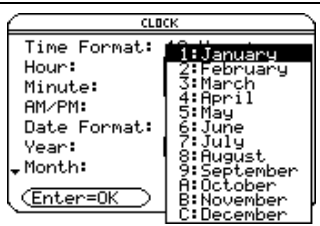
1. Drücken Sie ⊙ oder ⊙ so oft, bis das Feld Date Format (Datumsformat) markiert ist.
2. Drücken Sie ⊙, um eine Liste von Datumsformaten zu öffnen.
3. Drücken Sie ⊙ oder ⊙, um eine Option zu markieren und drücken Sie dann ENTER. Das gewählte Format wird im Feld Date Format (Datumsformat) angezeigt.
4. Drücken Sie ⊙, um das Feld Year (Jahr) zu markieren.
5. Geben Sie das Jahr ein und drücken Sie dann ⊙, um das Feld Month (Monat) zu markieren.
6. Drücken Sie ⊙, um eine Liste der Monate zu öffnen.

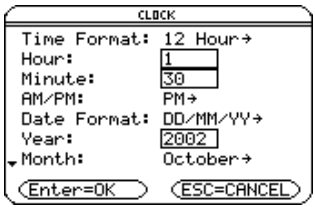
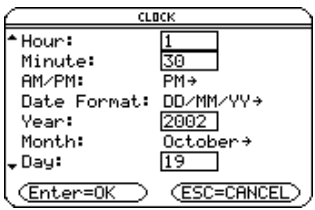
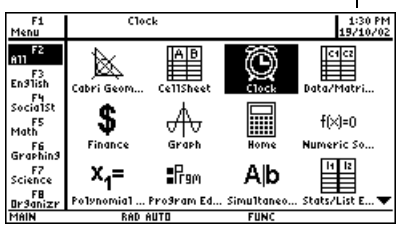
7. Drücken Sie ⊙ oder ⊙, um eine Option zu markieren und drücken Sie dann ENTER. Der gewählte Monat wird im Feld Month (Monat) angezeigt.
8. Drücken Sie ⊙, um das Feld Day (Tag) zu markieren.
9. Geben Sie den Tag ein und drücken Sie dann ENTER ENTER, um Ihre Einstellungen zu speichern und das Dialogfeld zu verlassen. Das Datum in der oberen rechten Ecke der Apps-Arbeitsfläche wird aktualisiert.

Beispiel: Setzen Sie Zeit und Datum auf 19.10.02 (19. Oktober 2002) um 1:30 p.m.

Drücken Sie	Ergebnis
Wählen Sie mit den Cursortasten 	
ENTER	

Drücken Sie	Ergebnis
⊙ 1 ⊙	
3 0 ⊙	
⊙ ⊙	
ENTER ⊙	

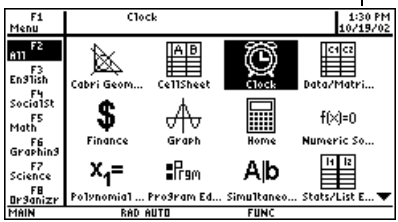
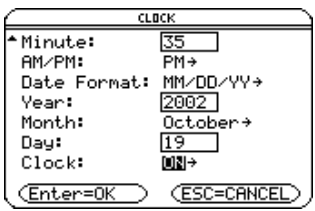
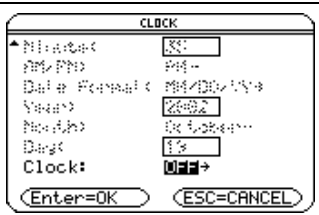
Drücken Sie	Ergebnis
⊙ ⊙	
ENTER ⊙	
2 0 0 2	
⊙ ⊙	

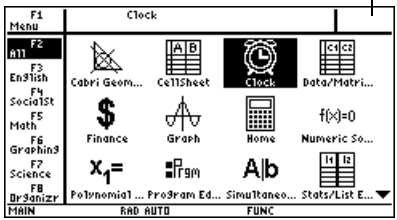
Drücken Sie	Ergebnis
Drücken Sie ⊙ oder ⊙ bis Oktober markiert ist und drücken Sie ENTER	
⊙ 1 9	
ENTER ENTER	Korrigierte Zeit und Datum 

Abschalten der Uhr

Öffnen Sie in der Apps-Arbeitsfläche das Dialogfeld CLOCK (UHR) und wählen Sie OFF (AUS) im Feld Clock (Uhr).

Beispiel: Schalten Sie die Uhr aus.

Drücken Sie	Ergebnis
Wählen Sie mit den Cursortasten 	Uhr an 
ENTER Blättern Sie nach unten zum Feld Clock (Uhr).	
⊙ ⊖ ENTER	

Drücken Sie	Ergebnis
ENTER	Uhr aus 

Wiederholen Sie den Vorgang zum Einschalten der Uhr und wählen Sie ON (AN) im Feld Clock (Uhr). Denken Sie daran, Zeit und Datum nachzustellen.

Verwenden von Menüs

Für die Auswahl der meisten TI-89 Titanium oder Voyage™ 200 Menüs drücken Sie die Funktionstasten, die mit den Menüleisten am oberen Rand des Rechner Hauptbildschirms und den meisten App Bildschirmen korrespondieren. Andere Menüs wählen Sie mit Tastenkommandos aus.

Menüleisten-Menüs

Im Hauptbildschirm des Rechners, dem Ausgangspunkt des TI-89 Titanium oder Voyage™ 200 für mathematische Operationen, können Sie aus einem mathematik-orientierten Menüleisten-Menü auswählen. (Siehe auch das Beispiel auf der folgenden Seite).

Menüleisten-Menüs erscheinen auch am oberen Rand der meisten App Bildschirme. Diese Menüs führen häufig genutzte Funktionen der aktiven App auf.

Weitere Menüs

Wählen Sie die folgenden Menüs mit Tastenbefehlen aus. Diese Menüs enhalten, unabhängig vom angezeigten Bildschirm oder der aktiven App, die selben Optionen.

Drücken Sie	Zur Anzeige
2nd [CHAR]	CHAR (ZEICHEN)-Menü. Enthält Zeichen, die nicht auf der Tastatur verfügbar sind. Diese sind nach Kategorien organisiert (griechische, mathematische, Interpunktions-, Sonder- und internationale Zeichen).
2nd [MATH]	MATH (MATHEMATIK)-Menü. Enthält nach Kategorien sortierte mathematische Operationen.
APPS	APPLICATIONS (APPLIKATIONEN)-Menü. Enthält die installierten Apps. (Das Menü ist nur bei deaktivierter Apps-Arbeitsfläche verfügbar; Apps erreichen Sie normalerweise über die Apps-Arbeitsfläche.)
♦ APPS	FLASH APPLICATIONS-Menü. Listet die installierten Flash Apps auf. (Das Menü ist nur bei deaktivierter Apps-Arbeitsfläche verfügbar; Flash Apps erreichen Sie normalerweise über die Apps- Arbeitsfläche.)

Auswählen von Menüoptionen

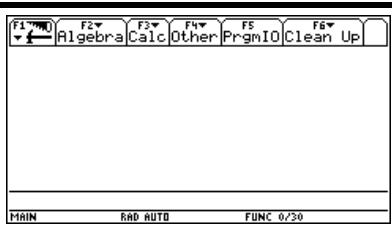
- Drücken Sie die Ziffer oder den Buchstaben links neben der Option, die Sie auswählen wollen.

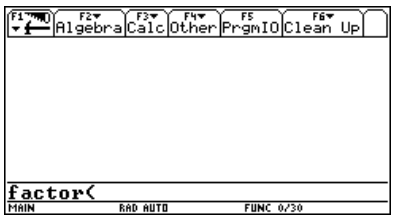
  – oder –

- Drücken Sie ⊖ oder ⊕, um die Option auszuwählen und drücken Sie [ENTER].

Hinweis: Wenn die erste Menüoption gewählt wurde, drücken Sie ⊖, um die letzte Option im Menü zu wählen. Wenn die letzte Menüoption gewählt wurde, drücken Sie ⊕, um die erste Option im Menü zu wählen.

Beispiel: Wählen Sie **factor(** aus dem Menü Algebra des Hauptbildschirms des Rechners.

Drücken Sie	Ergebnis
Drücken Sie [♦] [CALC HOME] – oder – Wählen Sie auf dem Apps-Bildschirm mit den Cursortasten und drücken Sie [ENTER]	

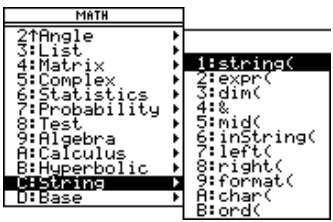
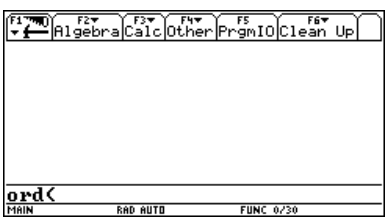
Drücken Sie	Ergebnis
F2	▼ deutet an, dass sich das Algebra Menü öffnet, wenn Sie F2 drücken.
2 – oder – ▼ ENTER	

Auswählen von Untermenüoptionen

Ein kleiner Pfeil (▶) rechts neben einer Menüoption weist darauf hin, dass sich bei Auswahl des Menüpunkts ein Untermenü öffnet.

↓ weist auf zusätzliche Optionen hin.

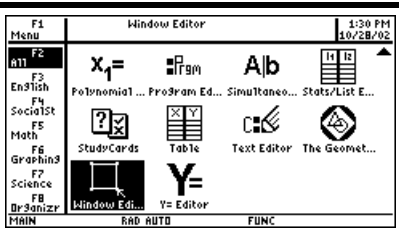
Beispiel: Wählen Sie **ord(** aus dem Menü **MATH (MATHEMATIK)** des Hauptbldschirms.

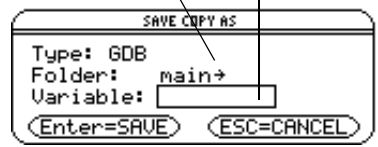
Drücken Sie	Ergebnis
2nd [MATH]	
C – oder – ▲ ▲ ▶	
B – oder – ▲ ENTER	

Gebrauch von Dialogfeldern

Drei Punkte (...) am Ende eines Menüpostens signalisieren, dass bei Auswahl der Option ein Dialogfeld geöffnet wird. Wählen Sie den Posten und drücken Sie ENTER.

Beispiel: Öffnen Sie das Dialogfeld SAVE COPY AS (KOPIE SPEICH ALS) im Window Editor.

Drücken Sie	Ergebnis
APPS Wählen Sie mit den Cursortasten und drücken Sie ENTER	
F1	

Drücken Sie	Ergebnis
2 – oder – ▼ ENTER ESC	Drücken Sie ▶ für eine Liste von Verzeichnissen. Geben Sie den Namen der Variablen ein.  Drücken Sie zwei Mal ENTER zum Sichern und Schließen des Dialogfelds.

Hinweis: Durch Drücken des ♦ S-Tastenkürzels wird das Dialogfeld SAVE COPY AS (KOPIE SPEICH ALS) in den meisten Apps ebenfalls geöffnet.

Verlassen eines Menüs

Zum Verlassen eines Menüs, ohne eine Auswahl zu treffen, drücken Sie ESC.

Wechseln zwischen Menüleisten-Menüs

So wechseln Sie zwischen den Menüs der Menüleisten, ohne einen Menüposten zu wählen:

- Drücken Sie eine der Funktionstasten (F1 bis F8) eines Menüleisten-Menüs.

- Drücken Sie eine Funktionstaste, danach ▶ oder ◀, um von einem Menüleisten-Menü zu nächsten zu gelangen. Drücken Sie im letzten Menü ▶, um zum ersten Menü zu gelangen und umgekehrt.

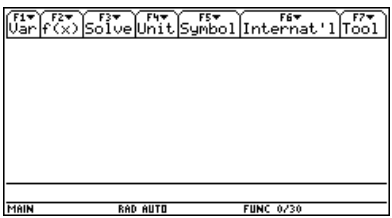
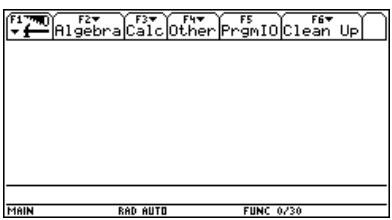
Hinweis: Wenn Sie ▶ drücken, während ein Menüposten mit einem Untermenü gewählt ist, dann wird anstelle des nächsten Menüs das Untermenü angezeigt. Drücken Sie ▶ noch einmal, um zum nächsten Menü zu gelangen.

Anwendermenü

Über das Anwendermenü erhalten Sie schnellen Zugriff auf die von Ihnen am häufigsten genutzten Optionen. Sie können das vorgegebene Anwendermenü verwenden oder mit dem Program Editor Ihr eigenes erstellen. Sie können alle verfügbaren Befehle und Zeichen des TI-89 Titanium oder Voyage™ 200 hinzufügen.

Das Anwendermenü ersetzt das Standard-Menüleisten-Menü auf dem Hauptbildschirm des Rechners. (Weitere Details zur Erstellung von Anwendermenüs finden Sie im TI-89 Titanium oder Voyage™ 200 Online-Modul *Programmieren*). Über Anwendermenüs sind weitere Informationen verfügbar.

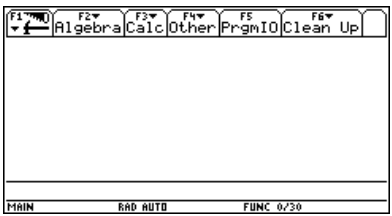
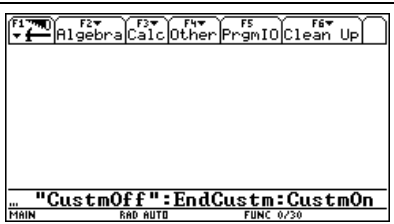
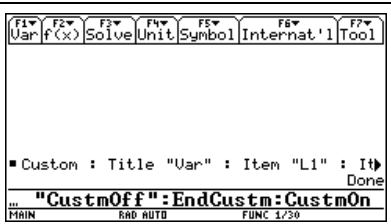
Beispiel: Aktivieren und deaktivieren Sie das Anwendermenü vom Hauptbildschirm des Rechners aus.

Drücken Sie	Ergebnis
2nd [CUSTOM]	Vorgegebenes Anwendermenü 
2nd [CUSTOM]	Normales Menüleisten-Menü 

Beispiel: Vorgegebenes Anwendermenü wiederherstellen

Hinweis: Das Wiederherstellen des vorgegebenen Anwendermenüs löscht das vorherige Anwendermenü. Wenn Sie das vorherige Anwendermenü mit einem Programm erstellt

haben, können Sie dieses Programm noch einmal ausführen, um das Menü wieder zu verwenden.

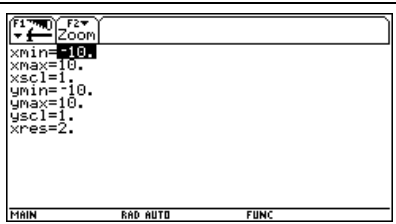
Drücken Sie	Ergebnis
2nd [CUSTOM] (um das kundenspezifische Menü zu deaktivieren und das Standard-Menüleisten-Menü zu verwenden)	
F6	
3 – oder – ▼ ▼ ENTER	
ENTER	

Öffnen von Apps bei deaktivierter Apps-Arbeitsfläche

Wenn Sie die Apps-Arbeitsfläche deaktivieren, verwenden Sie das APPLICATIONS (APPLIKATIONEN) Menü um Apps zu öffnen. Zum Öffnen des APPLICATIONS (APPLIKATIONEN) Menüs bei deaktivierter Apps-Arbeitsfläche drücken Sie APPS.

Hinweis: Wenn Sie APPS bei aktivierter Apps-Arbeitsfläche drücken, erscheint die Apps-Arbeitsfläche anstelle des Menüs APPLICATIONS (APPLIKATIONEN).

Beispiel: Öffen Sie bei deaktivierter Apps-Arbeitsfläche den Window Editor über das Menü APPLICATIONS (APPLIKATIONEN).

Drücken Sie	Ergebnis
APPS	
3 – oder – ▼ ▼ ENTER	

Wenn Sie Apps verwenden wollen, die nicht im Menü APPLICATIONS (APPLIKATIONEN) aufgeführt werden, wählen Sie **1:FlashApps**.

Verwalten von Apps und Betriebssystem-Versionen

Mit Hilfe der Verbindungsmöglichkeiten des TI-89 Titanium oder Voyage™ 200 können Sie Apps herunterladen, u.a. von:

- der TI Educational & Productivity Solutions (E&PS) Website unter: education.ti.com/latest
- der CD-ROM, die mit Ihrem TI-89 Titanium oder Voyage™ 200 ausgeliefert wurde.
- einem kompatiblen Graphik-Handheld.

Das Hinzufügen von Apps zu Ihrem TI-89 Titanium oder Voyage™ 200 funktioniert wie das Laden von Software auf einen Computer. Sie benötigen dazu lediglich die TI Connect™-Software und das USB-Kabel, das mit Ihrem TI-89 Titanium oder Voyage™ 200 mitgeliefert wurde.

Die Systemanforderungen und Anweisungen zum Verbinden kompatibler Handhelds und zum Herunterladen der TI Connect-Software, von Apps und BS-Versionen finden Sie auf der TI E&PS-Website.

Lesen Sie bitte vor dem Herunterladen von Apps auf Ihren TI-89 Titanium oder Voyage™ 200 die Lizenzvereinbarung auf der CD-ROM oder TI-Website.

Finden der BS-Version und Identifikations-(ID)-Nummern

Wenn Sie Software auf der TI E&PS-Website erwerben oder die Kundendienstnummer anrufen, werden Sie um Informationen zu Ihrem TI-89 Titanium oder Voyage™ 200 gebeten. Sie finden diese Informationen auf dem **ABOUT (INFO)**-Bildschirm.

Sie erhalten eine Anzeige des **ABOUT (INFO)**-Bildschirms, indem Sie F1 **3:About (3:Info)** auf der Apps-Arbeitsfläche drücken. Der **ABOUT (INFO)**-Bildschirm zeigt die folgenden Informationen über Ihren TI-89 Titanium oder Voyage™ 200:

❶ BS-Version

❷ Hardware-Version

❸ Geräte-ID (wird benötigt, um Zertifikate zur Installation gekaufter Apps zu erhalten). Ähnlich einer Seriennummer. Schreiben Sie sich diese Nummer auf und bewahren Sie sie an einem sicheren Ort auf, falls der Handheld einmal verloren geht oder gestohlen wird.

❹ Revisionsnummer des Apps-Zertifikats (Cert. Rev.)

❺ Produkt-ID (Product ID), ähnlich einer Typennummer.

Beachten Sie, dass Ihre Bildschirmanzeige sich von der oben gezeigten unterscheiden wird.

Löschen einer Applikation

Durch das Löschen einer Applikation wird diese vom TI-89 Titanium oder Voyage™ 200 entfernt und Platz für andere Applikationen geschaffen. Überlegen Sie vor dem Löschen einer Applikation, ob Sie diese zur späteren Neuinstallation auf einem Computer speichern wollen.

1. Beenden Sie die Applikation.
2. Drücken Sie 2nd [VAR-LINK], um den **VAR-LINK (All)**-Bildschirm anzuzeigen.
3. Drücken Sie 2nd [F7] (TI-89 Titanium) oder F7, um eine Liste der installierten Applikationen zu erhalten.
4. Wählen Sie die zu löschende Applikation indem Sie F4 drücken. (Drücken Sie F4 noch einmal, um die Auswahl aufzuheben.)
5. Drücken Sie F1 **1:Delete (1:Löschen)**. Das Dialogfeld **VAR-LINK** zur Bestätigung der Löschung wird angezeigt.
6. Drücken Sie ENTER, um die Applikation zu löschen.

Hinweis: Es können nur Flash-Apps gelöscht werden.

Anschließen des TI-89 Titanium oder Voyage™ 200 an andere Geräte

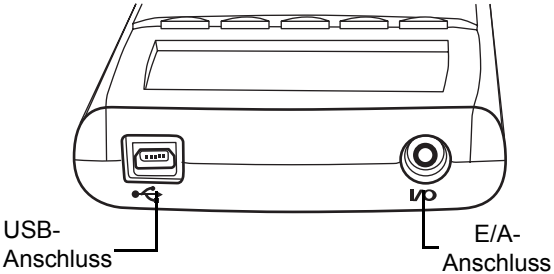
Der TI-89 Titanium besitzt einen Mini-USB-Anschluss. Sowohl der TI-89 Titanium als auch der Voyage™ 200 besitzen einen E/A-Anschluss. Diese Anschlüsse dienen dazu, zwei kompatible Graphik-Handhelds zu verbinden, oder einen Computer oder periphere Geräte anzuschließen.

Außerdem besitzen das Lehrermodell des TI-89 Titanium und alle Voyage™ 200 Handhelds einen Zubehöranschluss. Dieser Anschluss dient zur Ausgabe von Anzeigedaten, so dass die Klasse das Display des Handhelds über ein Videogerät oder als Overhead-Projektion sehen kann.

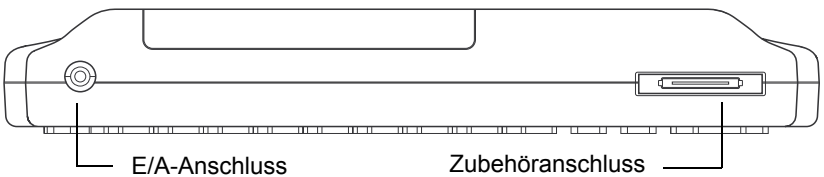
So verbinden Sie Ihren Handheld mit einem Computer – Schließen Sie den USB-Anschluss des TI-89 Titanium mit dem beiliegenden USB cable an, oder verbinden Sie den E/A-Anschluss des Voyage™ 200 mit dem beiliegenden TI Connectivity USB-Kabel.

So verbinden Sie Ihren Handheld mit einem anderen Handheld – Verwenden Sie das USB-Geräteverbindungskabel oder ein Standard-Geräteverbindungskabel, um den TI-89 Titanium oder Voyage™ 200 mit einem kompatiblen Graphik-Handheld oder einem peripheren Gerät, wie z.B. einem TI-89 oder TI-92 Plus Graphik-Handheld oder den Systemen CBL 2™ und CBR™ zu verbinden.

So machen Sie die Anzeige des Handheld-Displays für die Klasse sichtbar – Mit der Zubehör-Schnittstelle können Sie den TI-Presenter™-Videoadapter am Voyage™ 200 anschließen. Der TI-Presenter-Videoadapter stellt eine Video-Schnittstelle zwischen dem Voyage™ 200 und Video-Anzeige- oder Aufzeichnungsgeräten bereit. Oder verwenden Sie den Zubehöranschluss, um das TI ViewScreen™ OHP-Display an Ihren Handheld anzuschließen. Das TI ViewScreen OHP-Display vergrößert das Display und projiziert es, so dass für die ganze Klasse sichtbar ist. Weitere Informationen zum Videoadapter TI-Presenter und dem TI ViewScreen OHP-Display finden Sie auf der TI E&PS Website unter education.ti.com.



TI-89 Titanium Anschlüsse



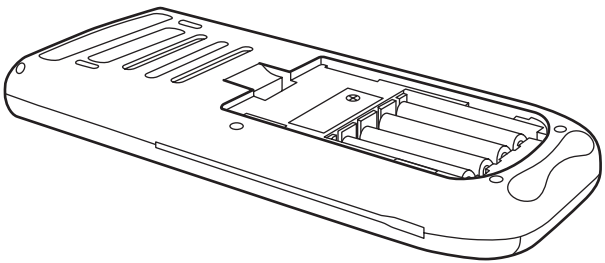
Voyage™ 200 Anschlüsse

Batterien

Der TI-89 Titanium verwendet vier AAA Alkali-Batterien und eine Silberoxid-Batterie (SR44SW oder 303) als Backup. Der Voyage™ 200 verwendet vier AAA Alkali-Batterien und eine Lithium-Batterie als Backup (CR1616 oder CR1620). Die Backup-Batterien sind bereits eingesetzt, die AAA Batterien werden mit den Geräten geliefert.

Einsetzen der AAA Batterien

1. Nehmen Sie die Batteriefachabdeckung an der Rückseite des Handhelds ab.
2. Packen Sie die vier AAA Batterien, die mit Ihrem Gerät geliefert wurden, aus und legen Sie sie in das Batteriefach ein. Ordnen Sie die Batterien im Batteriefach gemäß der schematischen Darstelellung der Polarität (+ und –) an.



3. Bringen Sie die Batteriefachabdeckung des Handhelds wieder an. Sie sollte einrasten.

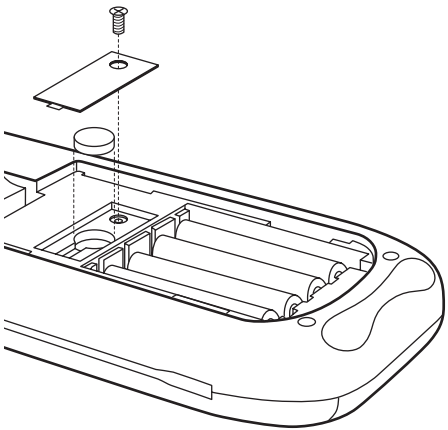
Austauschen der AAA Alkali-Batterien

Wenn die Batterieleistung nachlässt, verringert sich der Kontrast der Anzeige, insbesondere bei Berechnungen. Wenn Sie feststellen, dass Sie öfters den Kontrast nachstellen müssen, sollten Sie die AAA Alkali-Batterien ersetzen.

In der Statuszeile finden Sie auch Batterieinformationen. Schalten Sie den TI-89 Titanium oder Voyage™ 200 vor dem Austauschen der Batterien aus, indem Sie [2nd] [OFF] drücken, um einen Verlust der im Speicher abgelegten Informationen zu vermeiden. Nehmen Sie die Batterie und die AAA Alkali-Batterien nicht gleichzeitig heraus.

Austauschen der Backup- (Silberoxid-) Batterie

1. Um die Silberoxid Backup-Batterie auszutauschen, nehmen Sie den Batteriefachdeckel ab und lösen die kleine Schraube, mit der die Abdeckung BACK UP BATTERY befestigt ist.



2. Entnehmen Sie die alte Batterie und setzen Sie eine neue des Typs SR44SW oder 303 mit dem Pluspol (+) nach oben ein. Bringen Sie die Abdeckung und die Schraube wieder an.

Wichtige BS-Download-Information

Vor dem Herunterladen eines BS sollten neue Batterien eingelegt werden.

Im BS-Lademodus funktioniert das Leistungsmerkmal APD™ nicht. Wenn Sie Ihren Taschenrechner für längere Zeit im Download-Modus lassen, bevor Sie mit dem Download beginnen, können Ihre Batterien viel Energie verlieren. Dies kann einen Austausch der leeren Batterien vor dem Download notwendig machen.

Sie können mit einem Geräte-Verbindungskabel das BS auch auf einen anderen TI-89 Titanium oder Voyage™ 200 übertragen. Wenn Sie die Übertragung versehentlich vor dem Ende unterbrechen, müssen Sie das BS über einen Computer neu installieren. Vergessen Sie auf keinen Fall vor dem Download neue Batterien einzulegen.

Bitte wenden Sie sich an Texas Instruments, wie unter Hinweise zu TI Produktservice und Garantieleistungen beschrieben, falls ein Problem auftritt.

Vorsichtsmassnahmen im Umgang mit Batterien

Beachten Sie beim Austausch der Batterien die folgenden Vorsichtsmassnahmen.

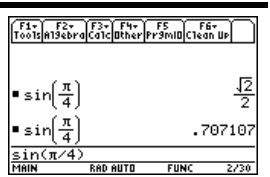
- Bewahren Sie Batterien außer Reichweite von Kindern auf.
- Verwenden Sie neue und alte Batterien nicht zusammen. Verwenden Sie Batterien unterschiedlicher Marken oder Typen nicht zusammen.
- Verwenden Sie Batterien und Akkumulatoren nicht zusammen.
- Legen Sie die Batterien mit der vorgegebenen Polarität (+ und -) ein.
- Legen Sie nicht-aufladbare Batterien nicht in ein Akku-Ladegerät ein.
- Entsorgen Sie verbrauchte Batterien vorschriftgemäß und so bald wie möglich.
- Batterien dürfen nicht ins Feuer geworfen oder geöffnet werden.

Vorschau

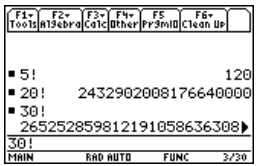
Berechnungen durchführen

In diesem Kapitel werden mehrere Beispiele behandelt, die vom Hauptbildschirm des Handhelds ausgehen und einige Rechenfunktionen des TI-89 Titanium / Voyage™ 200 demonstrieren. Vor der Ausführung jedes Beispiels wurde der Protokoll-Bereich des Bildschirms gelöscht, indem F1 gedrückt und **8:Clear Home** gewählt wurde. So wird lediglich das Ergebnis der Tastenfolgen in den Beispielen angezeigt.

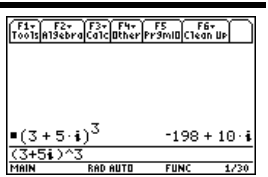
Berechnungen anzeigen

Schritte und Tastenfolgen	Anzeige
Berechnen Sie **sin(π/4)**, und zeigen Sie das Ergebnis in symbolischem und numerischem Format an. Zum Löschen des Protokoll-Bereichs vorheriger Berechnungen drücken Sie F1 und wählen **8:Clear Home**. 2nd [SIN] 2nd [π] ÷ 4) ENTER ◆ ENTER SIN 2nd [π] ÷ 4) ENTER ◆ ENTER	

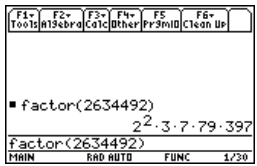
Ermittlung von Fakultäten

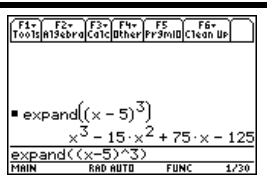
Schritte und Tastenfolgen	Anzeige
Berechnen Sie die Fakultät verschiedener Zahlen, um zu sehen, wie der TI-89 Titanium / Voyage™ 200 mit besonders großen Ganzzahlen umgeht. Zum Auswählen des Fakultät-Operators (!) drücken Sie [2nd] [MATH], wählen Sie **7:Probability** und dann **1:!**. 5 [2nd] [MATH] 7 1 [ENTER] 20 [2nd] [MATH] 7 1 [ENTER] 30 [2nd] [MATH] 7 1 [ENTER] 5 [2nd] [!] [ENTER] 20 [2nd] [!] [ENTER] 30 [2nd] [!] [ENTER]	

Rechnen mit komplexen Zahlen

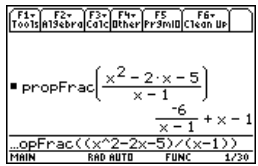
Schritte und Tastenfolgen	Anzeige
Berechnen Sie $(3+5i)^3$ um zu sehen, wie der TI-89 Titanium / Voyage™ 200 Berechnungen mit komplexen Zahlen ausführt. Drücken Sie [(] 3 [+] 5 [2nd] [i] [)] [^] 3 [ENTER]	

Ermittlung von PrimfaktorenTerme entwickeln

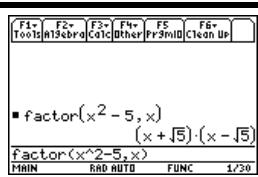
Schritte und Tastenfolgen	Anzeige
Berechnen Sie die Faktorzerlegung von 2634492. Sie können “factor” in der Eingabezeile eingeben, indem Sie über die Tastatur **FACTOR** schreiben oder die Taste F2 betätigen und **2:factor(** wählen. Drücken Sie F2 2 2634492) ENTER (*Optional*) Geben Sie selbst andere Zahlen ein.	

Schritte und Tastenfolgen	Anzeige
Entwickeln Sie den Term **(x−5)**^{3}. Sie können “expand” in der Eingabezeile eingeben, indem Sie über die Tastatur **EXPAND** schreiben oder die Taste F2 betätigen und **3:expand(** wählen. Drücken Sie F2 3 (X − 5) ^ 3) ENTER (*Optional*) Geben Sie selbst andere Terme ein.	

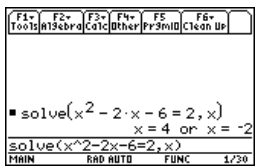
Terme vereinfachen

Schritte und Tastenfolgen	Anzeige
Vereifachen Sie den Term $(x^2-2x-5)/(x-1)$ als Partialbruch. Sie können “propFrac” in der Eingabezeile eingeben, indem Sie **PROPFRAC** über die Tastatur schreiben oder die Taste F2 betätigen und **7:propFrac(** wählen. Drücken Sie F2 7 (X ^ 2 − 2 X − 5) ÷ (X − 1)) ENTER	

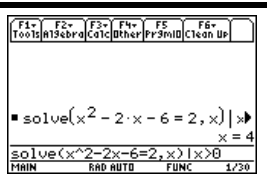
Faktorzerlegung von Polynomen

Schritte und Tastenfolgen	Anzeige
Zerlegen Sie das Polynom (x^2-5). Sie können “factor” in der Eingabezeile eingeben, indem Sie **FACTOR** über die Tastatur schreiben oder die Taste F2 betätigen und **2:factor(** wählen. Drücken Sie F2 2 X ^ 2 − 5 , X) ENTER	

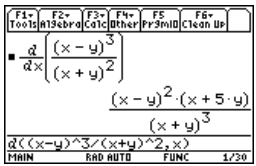
Gleichungen lösen

Schritte und Tastenfolgen	Anzeige
Lösen Sie die Gleichung $x^2-2x-6=2$ nach x auf. Sie können "**solve(**" in der Eingabezeile eingeben, indem Sie im Catalog-Menü "**solve(**" wählen, über die Tastatur **SOLVE(** schreiben oder die Taste F2 betätigen und **1:solve(** wählen. Der Statuszeilenbereich zeigt die für die markierte Position im **Catalog**-Menü erforderliche Syntax an. Drücken Sie F2 1 X ^ 2 − 2 X − 6 = 2 , X) ENTER	 

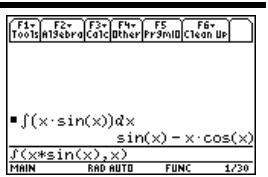
Gleichungen mit einer Einschränkung des Wertebereichs lösen

Schritte und Tastenfolgen	Anzeige
Lösen Sie die Gleichung **x2−2x−6=2** nach x auf, wobei x größer als Null sein soll. Mit dem Operator "*with*" (|) wird die Lösungsmenge weiter eingeschränkt. F2 1 X ^ 2 − 2 X − 6 = 2 , X)	X 2nd [>] 0 ENTER F2 1 X ^ 2 − 2 X − 6 = 2 , X) 2nd [|] X 2nd [>] 0 ENTER	

Ableitungen von Funktionen ermitteln

Schritte und Tastenfolgen	Anzeige
Ermitteln Sie die Ableitung von **(x−y)^3/(x+y)^2** bezüglich x. In diesem Beispiel wird der Differential-Operator und seine Anzeige in “Pretty Print” im Protokoll-Bereich dargestellt. Drücken Sie [2nd] [d] [(] X [−] Y [)] [^] 3 [÷] [(] X [+] Y [)] [^] 2 [,] X [)] [ENTER]	

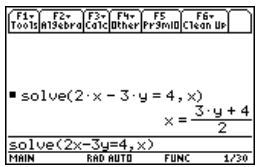
Integrale von Funktionen ermitteln

Schritte und Tastenfolgen	Anzeige
Ermitteln Sie das Integral von **x∗sin(x)** bezüglich x. Dieses Beispiel zeigt das Integrieren. [2nd] [∫] X [×] [2nd] [SIN] X [)] [,] X [)] [ENTER] [2nd] [∫] X [×] [SIN] X [)] [,] X [)] [ENTER]	

Vorschau auf symbolisches Rechnen

Lösen Sie das Gleichungssystem **2x − 3y = 4** and **−x + 7y = −12**. Lösen Sie die erste Gleichung so, dass x durch y ausgedrückt wird. Setzen Sie den Term für x in die zweite

Gleichung ein und berechnen Sie y. Setzen Sie dann den für y gefundenen Wert wieder in die erste Gleichung ein, um damit den Wert von x zu berechnen.

Schritte und Tastenfolgen	Anzeige
1. Gehen Sie in den Hauptbildschirm und löschen Sie die Eingabezeile. Lösen Sie die Gleichung $2x - 3y = 4$ nach x auf. F2 **1** wählt **solve(** aus dem Algebra-Menü. Sie können auch **solve(** direkt über die Tastatur eingeben bzw. die Funktion aus dem Dialogfeld **Catalog** (Katalog) auswählen. HOME CLEAR CLEAR F2 1 2 X − 3 Y = 4 , X) ENTER ◆ [CALC HOME] CLEAR CLEAR F2 1 2 X − 3 Y = 4 , X) ENTER	
2. Beginnen Sie mit dem Lösen der Gleichung **−x + 7y = −12** für y, aber drücken Sie ENTER noch nicht. Drücken Sie F2 1 (-) X + 7 Y = (-) 12 , Y)	

Schritte und Tastenfolgen	Anzeige

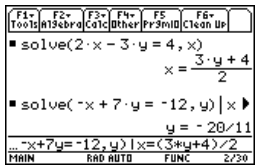
3. Verwenden Sie den "*with*"-Operator, um den Term für x einzusetzen, der aus der ersten Gleichung errechnet wurde. Sie erhalten den Wert für y.

   Der "*with*"-Operator erscheint als **|** auf dem Bildschirm.

   Verwenden Sie die automatische Einfüge-Funktion, um die letzte Antwort im Protokoll-Bereich zu markieren und diese in die Eingabezeile einzufügen.

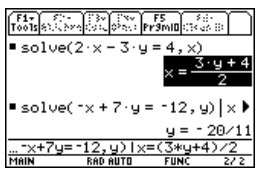
   [|] ▲ [ENTER] [ENTER]

   [2nd] [|] ▲ [ENTER] [ENTER]



4. Markieren Sie die Gleichung für *x* im Protokoll-Bereich.

   Drücken Sie ▲ ▲ ▲

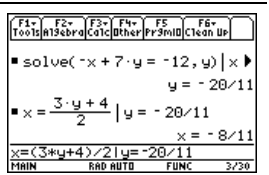


5. Fügen Sie den markierten Term in die Eingabezeile ein. Setzen Sie dann den aus der zweiten Gleichung errechneten Wert für y ein.

   [ENTER] [|] ▲ [ENTER] [ENTER]

   [ENTER] [2nd] [|] ▲ [ENTER] [ENTER]

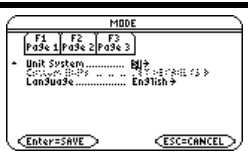
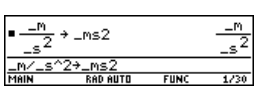
   Die Lösung lautet:
   x = ⁻8/11 and y = ⁻20/11



Dies ist ein Beispiel für symbolisches Rechnen. Zum Lösen von Gleichungssystemen ist eine eigene Funktion verfügbar.

Konstanten und Maßeinheiten

Berechnen Sie mit der Gleichung f = m∗a die Kraft, wenn m = 5 kg und a = 20 m/s^2 sind. Berechnen Sie die Kraft, wenn a = 9,8 m/s^2 beträgt. (Bei dieser Konstanten handelt es sich um die Erdbeschleunigung _g). Konvertieren Sie das Ergebnis von Newton in Kraftkilogramm.

Schritte und Tastenfolgen	Anzeige
1. Öffnen Sie das Dialogfeld **MODE**, Page 3. Wählen Sie **SI** für den Modus **Unit System**, um das internationale Einheitensystem einzustellen. Die Ergebnisse werden in diesen Standardeinheiten angezeigt. Drücken Sie MODE F3 ▶ **1** ENTER	
2. Erstellen Sie für m/s^2 die Beschleunigungseinheit namens _ms2. Im Dialogfeld für Maßeinheiten können Sie Einheiten aus einer alphabetisch sortierten Kategorienliste auswählen. Mit 2nd ▼ und 2nd ▲ können Sie die Kategorien Seite für Seite durchsuchen.	 

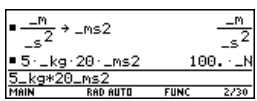
Schritte und Tastenfolgen	Anzeige

Nun können Sie _ms2 verwenden, anstatt jedesmal _m/_s^2 neu eingeben zu müssen. Außerdem kann _ms2 nun mit 2nd [UNITS] aus der Kategorie Acceleration (Beschleunigung) gewählt werden. Bei Auswahl einer Maßeinheit im Dialogfeld **UNITS** wird das Zeichen _ automatisch eingesetzt.

2nd [UNITS] ▼ ▶ M ENTER ÷ 2nd [UNITS] ▼ ▼ ▼ ▼ ▶ S ENTER ^ 2 STO▸ ♦ [_] 2nd [a-lock] MS alpha 2 ENTER

♦ [UNITS] ▼ ▶ M ENTER ÷ ♦ [UNITS] ▼ ▼ ▼ ▼ ▶ S ENTER ^ 2 STO▸ 2nd [_] MS2 ENTER

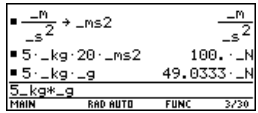
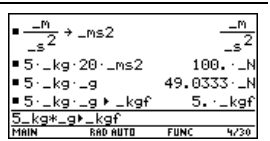
3. Berechnen Sie die Kraft, wenn m = 5 kg (_kg) und a = 20 m/s² (_ms2) ist.



Kennen Sie die Abkürzung einer Einheit, so können Sie diese direkt über die Tastatur eingeben.

5 ♦ [_] 2nd [a-lock] KG alpha × 20 ♦ [_] 2nd [a-lock] MS alpha 2 ENTER

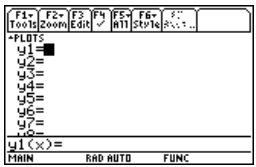
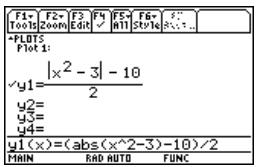
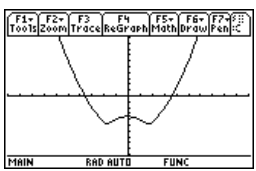
5 2nd [_] KG × 20 2nd [_] MS2 ENTER

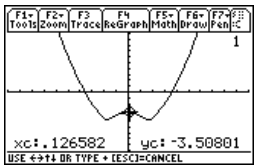
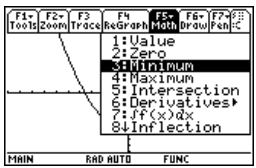
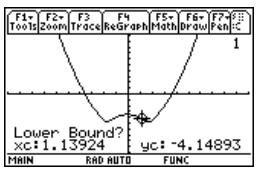
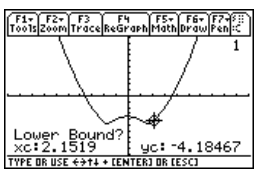
Schritte und Tastenfolgen	Anzeige
4. Berechnen Sie unter Beibehaltung von m die Gewichtskraft (mit Hilfe der Konstanten _g). Für _g können Sie die vordefinierte Konstante aus dem Dialogfeld **UNITS** einsetzen oder _g eingeben. 5 ♦ [_] 2nd [a-lock] KG alpha × 2nd [UNITS] ▶ alpha G ENTER ENTER 5 2nd [_] KG × ♦ [UNITS] ▶ G ENTER ENTER	
5. Konvertieren Sie in Kraftkilogramm (_kgf). 2nd [▶] zeigt den Konvertierungsoperator ▸ an. ▶ 2nd [▶] ♦ [_] 2nd [a-lock] KGF alpha ENTER ▶ 2nd [▶] 2nd [_] KGF ENTER	

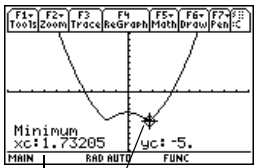
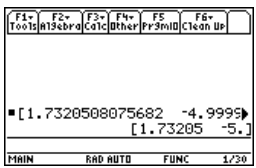
Grundlagen der graphischen Darstellung von Funktionen I

In diesem Beispiel werden Ihnen einige Graphikfunktionen des TI-89 Titanium / Voyage™ 200 vorgestellt. Es wird gezeigt, wie Funktionen mit Hilfe des **Y= Editor** graphisch dargestellt werden können. Sie erfahren, wie Sie eine Funktion eingeben, einen Graph einer Funktion erstellen, eine Kurve tracen, einen Tiefpunkt ermitteln und die Minimum-Koordinaten in den Hauptbildschirm übertragen können.

Entdecken Sie die graphischen Fähigkeiten von TI-89 Titanium / Voyage™ 200, indem Sie die Funktion **y=(|x^2−3|−10)/2** graphisch darstellen.

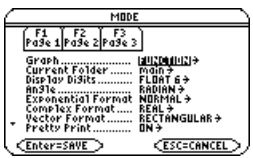
Schritte und Tastenfolgen	**Anzeige**
1. Rufen Sie den **Y= Editor** auf. Drücken Sie ♦ [Y=]	
2. Geben Sie die Funktion **(abs(x^2−3)−10)/2** ein. "Pretty- Print" -Anzeige der Funktion bei y1=. (CATALOG A ENTER X ^ 2 − 3) − 1 0) ÷ 2 ENTER (2nd [CATALOG] A ENTER X ^ 2 − 3) − 1 0) ÷ 2 ENTER	
3. Lassen Sie den Graph der Funktion anzeigen. Wählen Sie **6:ZoomStd**, indem Sie **6** drücken oder den Cursor auf **6:ZoomStd** setzen und ENTER drücken. Drücken Sie F2 6	

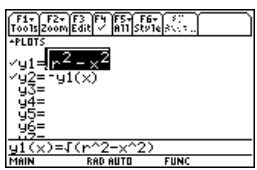
Schritte und Tastenfolgen	Anzeige
4. Aktivieren Sie die **Trace**-Funktion. Der Trace Cursor sowie die x- und y-Koordinaten werden angezeigt. Drücken Sie F3	
5. Öffnen Sie das **MATH** -Menü, und wählen Sie **3:Minimum.** Drücken Sie F5 ▼ ▼ ENTER	
6. Setzen Sie die untere Grenze. Drücken Sie ▶ (Cursor rechts) zum Bewegen des Zeichen-Cursors so lange, bis er sich knapp links neben dem Minimum befindet. Drücken Sie dann erneut ENTER. Drücken Sie ▶ ... ▶ ENTER	
7. Setzen Sie die obere Grenze. Drücken Sie ▶ (Cursor rechts) zum Bewegen des Zeichen-Cursors so lange, bis er sich knapp rechts neben dem Minimum befindet. Drücken Sie ▶ ... ▶	

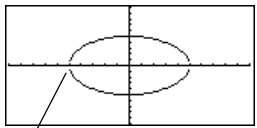
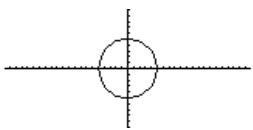
Schritte und Tastenfolgen	Anzeige
8. Ermitteln Sie den Tiefpunkt des Graphen zwischen unterer und oberer Grenze. Drücken Sie ENTER	
9. Übertragen Sie das Ergebnis in den Hauptbildschirm, und blenden Sie den Hauptbildschirm ein. ◆ (-) HOME ◆ H ◆ [CALC HOME]	

Grundlagen der graphischen Darstellung von Funktionen II

Stellen Sie einen Kreis mit Radius 5 und Mittelpunkt im Ursprung des Koordinatensystems graphisch dar. Schauen Sie sich den Kreis im Standard-Ansichtfenster (**ZoomStd**) an. Verwenden Sie dann **ZoomSqr** zum Anpassen des Ansichtfensters.

Schritte und Tastenfolgen	Anzeige
1. Öffnen Sie das **MODE**-Dialogfeld an. Wählen Sie **FUNCTION** für den **Graph**-Modus. Drücken Sie MODE ▶ 1 ENTER	
2. Wählen Sie den Hauptbildschirm. Speichern Sie dann den Radius 5 in der Variablen r. HOME 5 STO▸ alpha R ENTER ◆ [CALC HOME] 5 STO▸ R ENTER	5→r

Schritte und Tastenfolgen	Anzeige
3. Zeigen Sie den Y= Editor an und löschen Sie ihn. Definieren Sie dann y1(x) = $\sqrt{(r^2 - x^2)}$, für die obere Kreishälfte. Bei der graphischen Darstellung von Funktionen müssen Sie für die obere und untere Hälfte eines Kreises unterschiedliche Funktionen definieren. [♦] [Y=] [F1] 8 [ENTER] [ENTER] [2nd] [√] [alpha] R [^] 2 [−] X [^] 2 [)] [ENTER] [♦] [Y=] [F1] 8 [ENTER] [ENTER] [2nd] [√] R [^] 2 [−] X [^] 2 [)] [ENTER]	
4. Definieren Sie y2(x) = $-\sqrt{r^2 - x^2}$, für die untere Kreishälfte. Die obere Hälfte ist das Negative der oberen Hälfte; Sie können also y2(x) = ⁻y1(x) definieren. Den vollen Funktionsnamen **y1(x)** benutzen, nicht nur y1. Drücken Sie [ENTER] [(-)] Y 1 [(] X [)] [ENTER]	

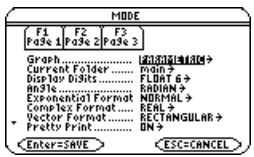
Schritte und Tastenfolgen	Anzeige
5. Wählen Sie das ZoomStd-Ansichtfenster, in welchem die Funktionen automatisch graphisch dargestellt werden. Im Standard-Ansichtfenster reicht sowohl die x- als auch die y-Achse von ⁻10 bis 10. Der Bereich ist bei der x-Achse auf eine längere Strecke aufgetragen als bei der y-Achse. Der Kreis erscheint deshalb als eine Ellipse. Drücken Sie F2 **6**	 Beachten Sie die kleine Lücke zwischen der oberen und der unteren Hälfte.
6. Wählen Sie **ZoomSqr**. **ZoomSqr** vergrößert die Bereiche der x-Achsen, so dass die Proportionen von Kreisen und Quadraten korrekt angezeigt werden. Drücken Sie F2 5	

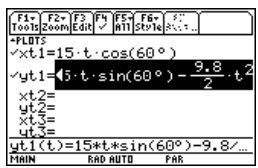
Hinweis: Da jede Kreishälfte durch eine eigene Funktion bestimmt wird, befindet sich zwischen oberer und unterer Hälfte eine Lücke. Die mathematischen Endpunkte jeder Hälfte sind (⁻5,0) und (5,0). Je nach Ansichtfenster können die gezeichneten Endpunkte leicht von den mathematischen Endpunkten abweichen.

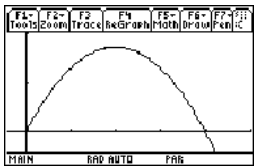
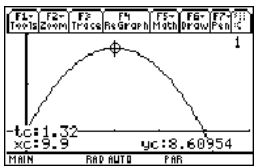
Parameterdarstellungen

Erzeugen Sie die Parameterdarstellung der Kurve, die ein Ball, der unter einem Winkel (θ) 60° und mit der Anfangsgeschwindigkeit (v_0) 15 m/s getreten wird, beschreibt. Die

Gravitationskonstante g beträgt 9,8 m/s^2. Welche maximale Höhe erreicht der Ball und wann trifft er auf, wenn Luftwiderstand und sonstige Verzögerungskräfte unberücksichtigt bleiben?

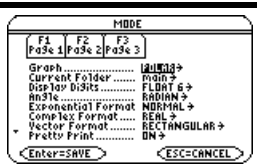
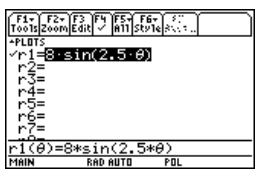
Schritte und Tastenfolgen	Anzeige
1. Öffnen Sie das Dialogfeld **MODE**. Als **Graph**-Modus wählen Sie **PARAMETRIC**. Drücken Sie MODE ▶ 2 ENTER	
2. Rufen Sie den Y= Editor auf, und löschen Sie seinen Inhalt. Definieren Sie nun die horizontale Komponente xt1(t) = v_0t cos θ. Geben Sie die Werte für v_0 und θ ein. Drücken Sie T × 2nd [COS], nicht T 2nd [COS]. Drücken Sie T × COS, nicht T COS.	xt1(t)=15t∗cos(60°)

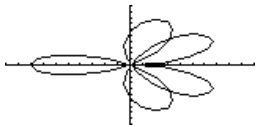
Schritte und Tastenfolgen	Anzeige
Geben Sie das Gradsymbol ° ein: entweder durch Drücken von 2nd [°] oder 2nd [MATH] 2 1 ein. So wird sichergestellt, dass eine Zahl unabhängig vom Winkelmodus als Grad interpretiert wird. ◆ [Y=] F1 8 ENTER ENTER 15T × 2nd [COS] 60 2nd [°]) ENTER ◆ [Y=] F1 8 ENTER ENTER 15T × COS 60 2nd [°]) ENTER	
3. Definieren Sie die vertikale Komponente **yt1(t) = v_0t sin θ – (g/2)t²**. Geben Sie die Werte für v_0, θ und g ein. ENTER 15T × 2nd [SIN] 60 2nd [°]) − (9.8 ÷ 2) T ^ 2 ENTER ENTER 15T × SIN 60 2nd [°]) − (9.8 ÷ 2) T ^ 2 ENTER	
4. Rufen Sie den Window Editor auf. Geben Sie für dieses Beispiel passende Fenstervariablen ein. Drücken Sie ⊙ oder ENTER, um einen Wert einzugeben und zur nächsten Variablen zu gelangen. Drücken Sie ◆ [WINDOW] 0 ⊙ 3 ⊙ .02 ⊙ (-) 2 ⊙ 25 ⊙ 5 ⊙ (-) 2 ⊙ 10 ⊙ 5	tmin=0. tmax=3. tstep=.02 xmin=-2. xmax=25. xscl=5. ymin=-2. ymax=10. yscl=5.

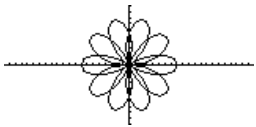
Schritte und Tastenfolgen	Anzeige
5. Stellen Sie die Flugbahn des Balls graphisch dar. Drücken Sie ♦ [GRAPH]	
6. Wählen Sie **Trace**. Bewegen Sie den Cursor entlang der Bahn, um folgende Werte zu ermitteln: • y-Wert bei maximaler Höhe • t-Wert bei dem der Ball auf dem Boden auftrifft. Drücken Sie F3 ⊙ oder ⊙ so weit erforderlich	

Polardarstellungen

Der Graph der polaren Gleichung r1(θ) = A sin B θ hat die Form einer Rose. Zeichnen Sie die Rose mit A=8 und B=2.5. Setzen Sie nun andere Werte für A und B ein, und beobachten Sie, wie sich das Aussehen der Rose verändert.

Schritte und Tastenfolgen	Anzeige
1. Öffnen Sie das Dialogfeld **MODE**. Wählen Sie als **Graph**-Modus **POLAR**. Wählen Sie als 0 **Angle**-Modus **RADIAN**. Drücken Sie MODE ▷ 3 ▽ ▽ ▽ ▷ 1 ENTER	
2. Aktivieren und löschen Sie den Y= Editor. Definieren Sie nun die Polardarstellung r1(θ) = A sin Bθ. Geben Sie für A 8 und für B 2,5 ein. ▭ ◆ [Y=] F1 8 ENTER ENTER 8 2nd [SIN] 2.5 ◆ [θ]) ENTER ⌨ ◆ [Y=] F1 8 ENTER ENTER 8 SIN 2.5 θ) ENTER	

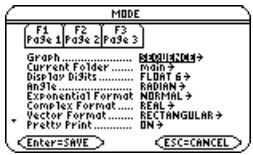
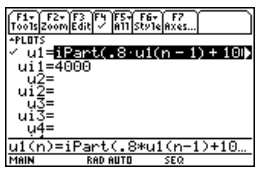
Schritte und Tastenfolgen	Anzeige
3. Wählen Sie das **ZoomStd**-Ansichtfenster, um die Darstellung zu erzeugen. • Die Darstellung zeigt nur fünf "Blätter". - Im Standard-Ansichtfenster ist die Window-Variable θmax = 2π. Die übrigen Blätter haben θ Werte größer als 2π.	
• Die Rose erscheint unsymmetrisch. - Der Wertebereich beträgt sowohl für die x- als auch die y-Achse ⁻10 bis 10. Diese Werte verteilen sich jedoch über eine längere Strecke entlang der x-Achse, als das bei der y-Achse der Fall ist. Drücken Sie F2 6	
4. Aktivieren Sie den Window Editor, und ändern Sie θmax auf 4π. 4π wird als Zahl berechnet, wenn Sie den Window Editor verlassen. Drücken Sie ♦ [WINDOW] ⊖ 4 2nd [π]	θmin=0. θmax=4π θstep=.13089969389957 xmin=-10. xmax=10. xscl=1. ymin=-10. ymax=10. yscl=1.

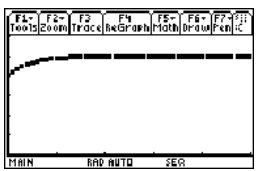
Schritte und Tastenfolgen	Anzeige
5. Wählen Sie **ZoomSqr**, um die Darstellung neu zu zeichnen. **ZoomSqr** vergrößert den Bereich entlang der x-Achse, so dass die Darstellung im richtigen Größenverhältnis erfolgt. Drücken Sie F2 5	
Sie können die Zahlenwerte für *A* und *B* ändern und die Darstellung erneut zeichnen lassen.	

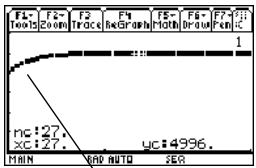
Graphische Darstellung von Folgen

Ein kleiner Wald besteht aus 4000 Bäumen. Jedes Jahr werden 20% der Bäume gefällt (80% bleiben erhalten) und 1000 neue Bäume gepflanzt. Verwenden Sie eine Folge, um die Anzahl der im Wald stehenden Bäume am Ende jedes Jahres zu berechnen. Wird die Zahl der Bäume bei einer bestimmten Zahl konstant?

Anfang	Nach 1 Jahr	Nach 2 Jahren	Nach 3 Jahren	. . .
4000	.8 x 4000 + 1000	.8 x (.8 x 4000 + 1000) + 1000	.8 x (.8 x (.8 x 4000 + 1000) + 1000) + 1000	. . .

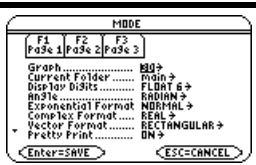
Schritte und Tastenfolgen	Anzeige
1. Öffnen Sie das Dialogfeld **MODE**. Wählen Sie als **Graph**-Modus **SEQUENCE**. Drücken Sie MODE ▶ 4 ENTER	
2. Aktivieren und löschen Sie den **Y= Editor**. Definieren Sie nun die Folge als u1(n) = iPart(.8∗u1(n−1)+1000). Verwenden Sie **iPart**, um den ganzzahligen Teil des Ergebnisses zu erhalten. Schließlich werden keine Bruchteile von Bäumen gefällt.	
iPart(wird wie folgt aufgerufen: Mit 2nd [MATH], durch Eingabe über die Tastatur oder durch Auswahl aus dem Dialogfeld **CATALOG**. ◆ [Y=] F1 8 ENTER ENTER 2nd [MATH] 14.8 alpha U1 (alpha N − 1) + 1000) ENTER ◆ [Y=] F1 8 ENTER ENTER 2nd [MATH] 14.8 U1 (N − 1) + 1000) ENTER	
3. Definieren Sie ui1 als Anfangswert, der als erstes Glied verwendet wird. Drücken Sie ENTER 4000 ENTER	

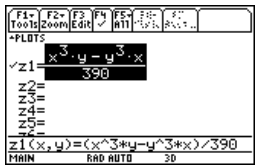
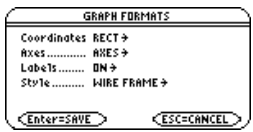
Schritte und Tastenfolgen	Anzeige
4. Rufen Sie den Window Editor auf. Stellen Sie die n- und plot-Window-Variablen ein. Mit nmin=0 und nmax=50 wird die Größe des Waldes über einen Zeitraum von 50 Jahren berechnet. Drücken Sie [♦] [WINDOW] 0 ⊖ 50 ⊖ 1 ⊖ 1 ⊖	nmin=0. nmax=50. plotStrt=1. plotStep=1. xmin=0. xmax=50. xscl=10. ymin=0. ymax=6000. yscl=1000.
5. Setzen Sie die Window-Variablen x und y für dieses Beispiel auf passende Werte. Drücken Sie 0 ⊖ 50 ⊖ 10 ⊖ 0 ⊖ 6000 ⊖ 1000	
6. Aktivieren Sie den Graphikbildschirm. Drücken Sie [♦] [GRAPH]	

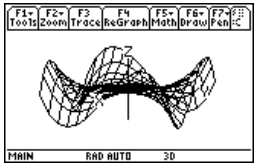
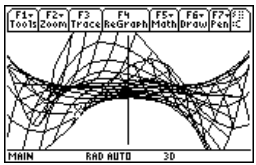
Schritte und Tastenfolgen	Anzeige
7. Wählen Sie **Trace**. Bewegen Sie den Cursor, um die Entwicklung jahresweise zu verfolgen. Nach wieviel Jahren (nc) hat sich die Zahl der Bäume (yc) stabilisiert? Das Tracen beginnt bei nc=0. nc ist die Anzahl der Jahre. xc = nc, da n auf der x-Achse abgebildet wird. yc = u1(n), die Anzahl der Bäume im Jahr n. Drücken Sie F3 ▶ und ◀ wie benötigt	 Als Standard wird bei Folgen der Anzeigestil "Quadrat" verwendet.

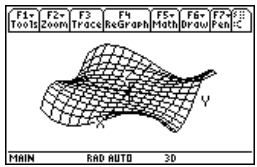
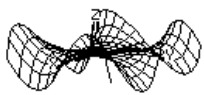
3D-Darstellungen

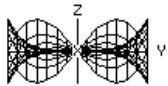
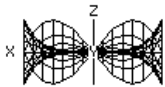
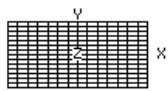
Zeichnen Sie die Fläche mit $z(x,y) = (x^3y - y^3x) / 390$. Zeigen Sie den Graph animiert an, indem Sie die Werte der für den Blickwinkel verantwortlichen “eye”-Fenstervariablen mit dem Cursor interaktiv ändern. Zeigen Sie den Graph dann in verschiedenen Graphikformat-Stilen an.

Schritte und Tastenfolgen	Anzeige
1. Öffnen Sie das Dialogfeld **MODE**. Als **Graph**-Modus wählen Sie **3D**. Drücken Sie MODE ▶ 5 ENTER	

Schritte und Tastenfolgen	Anzeige
2. Öffnen und löschen Sie den **Y= Editor**. Definieren Sie nun die 3D-Gleichung **z1(x,y) = ($x^3y - y^3x$) / 390.** Beachten Sie, wie bei den Tastenfolgen die implizite Multiplikation verwendet wird. Drücken Sie ♦ [Y=] F1 8 ENTER ENTER (X ^ 3 Y − Y ^ 3 X) ÷ 390 ENTER	
3. Ändern Sie das Grafikformat, um die Achsen anzuzeigen und zu beschriften. Nehmen Sie außerdem die Einstellung **Style = WIRE FRAME** vor. Eine Animation ist mit jedem Graphikformat-Stil möglich, die schnellste Methode ist allerdings mit **WIRE FRAME.** ♦ | ⊖ ⊕ 2 ⊖ ⊕ 2 ⊖ ⊕ 1 ENTER ♦ F ⊖ ⊕ 2 ⊖ ⊕ 2 ⊖ ⊕ 1 ENTER	

Schritte und Tastenfolgen	Anzeige
4. Wählen Sie die **ZoomStd**-Ansicht. Dadurch wird die Gleichung automatisch graphisch dargestellt. Während der Berechnung der Gleichung (bevor der Graph abgebildet wird) erscheint in der oberen linken Bildschirmecke die Anzeige des bereits berechneten Anteils in Prozent. Drücken Sie [F2] 6	 
Hinweis: Haben Sie die 3D-Darstellung bereits verwendet, wird der Graph möglicherweise in erweiterter Ansicht angezeigt. Bei der Erzeugung einer Animation des Graphen kehrt die Anzeige automatisch zur normalen Ansicht zurück. (In der normalen und der erweiterten Ansicht sind, außer dem Anzeigen von Animationen, dieselben Vorgänge möglich.) Drücken Sie [×] (drücken Sie [×] um zwischen normaler und erweiterter Ansicht umzuschalten)	

Schritte und Tastenfolgen	Anzeige
5. Zeigen Sie den Graph bewegt an, indem Sie den Wert der Fenstervariablen eyeϕ verkleinern. ⊖ oder ⊕ haben einen weniger starken Einfluß auf eyeθ und eyeψ als eyeϕ. Um den Graph kontinuierlich zu bewegen, drücken Sie den Cursor ca. 1 Sekunde lang und lassen ihn dann los. Zum Anhalten drücken Sie ENTER. Drücken Sie ⊖ acht mal.	
6. Stellen Sie die ursprüngliche Ausrichtung des Graphen wieder her. Bewegen Sie dann den Betrachtungswinkel entlang der den Graph umgebenden “Ansichtsspur”. Drücken Sie 0 (Null, nicht Buchstabe O) ◀ ◀ ◀	

Schritte und Tastenfolgen	Anzeige
7. Zeigen Sie den Graph entlang der x-Achse, der y-Achse und dann entlang der z-Achse an. Drücken Sie X Dieser Graph weist entlang der y- und der x-Achse dieselbe Form auf. Drücken Sie Y Drücken Sie Z	  
8. Kehren Sie zur ursprünglichen Ausrichtung zurück. Drücken Sie 0 (Null)	

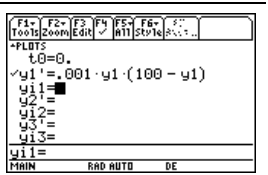
Schritte und Tastenfolgen	Anzeige
9. Zeigen Sie den Graph in verschiedenen Graphikformat-Stilen an. [I] (drücken Sie [I] um von einem Stil auf den nächsten umzuschalten) F (drücken Sie F drücken Sie F, um von einem Stil auf den nächsten umzuschalten)	 HIDDEN SURFACE CONTOUR LEVELS (die Berechnung der Konturen kann zusätzliche Zeit beanspruchen) WIRE AND CONTOUR WIRE FRAME

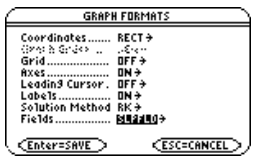
Hinweis: Sie können den Graph auch als impliziten Plot anzeigen. Verwenden Sie hierzu das Dialogfeld **GRAPH FORMATS** ([F1] 9 oder TI-89 Titanium: [◆] [I];

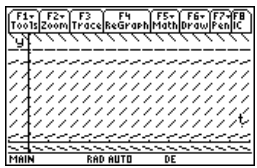
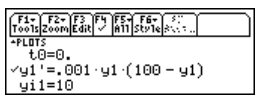
Voyage™ 200: ♦ F). Wenn Sie zum Umschalten zwischen den verschiedenen Stilen TI-89 Titanium: I; Voyage™ 200: F drücken, wird der implizite Plot nicht angezeigt.

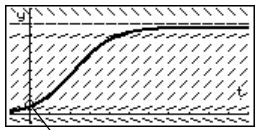
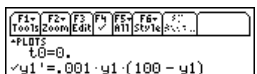
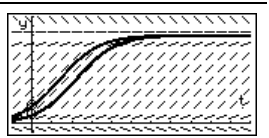
Graphische Darstellung von Differentialgleichungen

Stellen Sie die Lösung für die logistische Differentialgleichung erster Ordnung y' = .001y∗(100−y) graphisch dar. Beginnen Sie mit dem Richtungsfeld. Geben Sie dann im Y= Editor und interaktiv über den Graphikbildschirm die Anfangsbedingungen ein.

Schritte und Tastenfolgen	Anzeige
1. Öffnen Sie das Dialogfeld **MODE**. Als **Graph**-Modus wählen Sie **DIFF EQUATIONS**. Drücken Sie MODE ▷ 6 ENTER	
2. Öffnen und löschen Sie den **Y= Editor**. Definieren Sie nun die Differentialgleichung erster Ordnung: **y1'(t)=.001y1∗(100−y1)** Drücken Sie ×, um das Zeichen ∗ einzugeben. Verwenden Sie keine implizite Multiplikation zwischen Variablen und Klammern, sonst wird die Eingabe als Funktionsaufruf behandelt.	

Schritte und Tastenfolgen	Anzeige
Lassen Sie die Anfangsbedingung **yi1** frei. **Hinweis:** Wenn y1' ausgewählt ist, zeichnet der TI-89 Titanium / Voyage™ 200 die Graphik der Lösung für y1, nicht die Graphik der Differentialgleichung y1'. Drücken Sie ♦ [Y=] F1 8 ENTER ENTER .001 Y1 × (100 − Y1) ENTER	
3. Öffnen Sie das Dialogfeld **GRAPH FORMATS**, und nehmen Sie folgende Einstellungen vor: **Axes = ON**, **Labels = ON**, **Solution Method = RK** und **Fields = SLPFLD**. **Hinweis:** Soll eine einzige Differentialgleichung dargestellt werden, muss Fields auf **SLPFLD** oder **FLDOFF** eingestellt sein. Bei **Fields=DIRFLD** tritt während des Zeichnens ein Fehler auf. ♦ I ▼ ▼ ▶ 2 ▼ ▼ ▶ 2 ▼ ▶ 1 ▼ ▶ 1 ENTER ♦ F ▼ ▼ ▶ 2 ▼ ▼ ▶ 2 ▼ ▶ 1 ▼ ▶ 1 ENTER	

Schritte und Tastenfolgen	Anzeige
4. Rufen Sie den Window Editor auf, und stellen Sie die Fenstervariablen wie hier angegeben ein. Drücken Sie ♦ [WINDOW] 0 ▼ 10 ▼ .1 ▼ 0 ▼ (-) 10 ▼ 110 ▼ 10 ▼ (-) 10 ▼ 120 ▼ 10 ▼ 0 ▼ .001 ▼ 20	t0=0. tmax=10. tstep=.1 tplot=0. xmin=-10. xmax=110. xscl=10. ymin=-10. ymax=120. yscl=10. ncurves=0. diftol=.001 fldres=20.
5. Öffnen Sie den Grafikbildschirm. Da Sie keine Anfangsbedingung angegeben haben, wird nur das Richtungsfeld dargestellt (entsprechend der Einstellung **Fields=SLPFLD** im Dialogfeld **GRAPH FORMATS**). Drücken Sie ♦ [GRAPH]	
6. Kehren Sie zurück zum Y= Editor, und geben Sie eine Anfangsbedingung ein: yi1=10 Drücken Sie ♦ [Y=] ENTER 10 ENTER	

Schritte und Tastenfolgen	Anzeige
7. Rufen Sie den Grafikbildschirm wieder auf. Im Y= Editor eingegebene Anfangsbedingungen treten immer bei t_0 auf. Der Graph beginnt am Anfangswert und wird zunächst nach rechts und dann nach links gezeichnet. Drücken Sie ♦ [GRAPH]	 Die Anfangsbedingung ist mit einem Kreis gekennzeichnet.
8. Kehren Sie zum Y= Editor zurück. Ändern Sie yi1, indem Sie eine Liste von zwei Anfangsbedingungen eingeben: yi1={10,20} Drücken Sie ♦ [Y=] ⊙ ENTER 2nd [{] 10 , 20 2nd [}] ENTER	
9. Öffnen Sie den Graphikbildschirm erneut. Drücken Sie ♦ [GRAPH]	

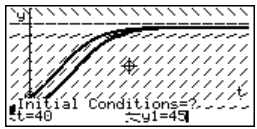
Schritte und Tastenfolgen	Anzeige

10. Eine Anfangsbedingung wird wie folgt interaktiv ausgewählt:

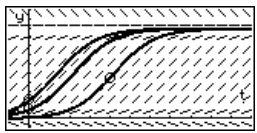
    [2nd] [F8]

    [F8]

    Geben Sie auf die entsprechenden Aufforderungen hin t=40 und y1=45 ein.



    Bei der interaktiven Wahl einer Anfangsbedingung können Sie für t einen anderen Wert angeben als den im Y= Editor oder Fenster-Editor eingegebenen Wert t_0.



    Anstatt zuerst

    [2nd] [F8]

    [F8]

    zu drücken und dann t und y1 einzugeben, können Sie den Cursor an eine gewünschte Stelle auf dem Bildschirm bewegen und dann [ENTER] drücken.

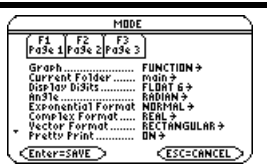
    Sie können mit [F3] Kurven für die im Y= Editor angegebenen Anfangsbedingungen verfolgen (tracen) lassen. Die Kurve für eine interaktiv gewählte Anfangsbedingung können Sie nicht verfolgen (tracen) lassen.

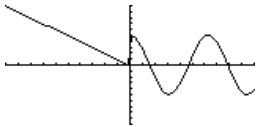
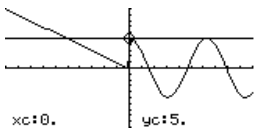
    [2nd] [F8] 40 [ENTER] 45 [ENTER]

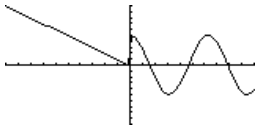
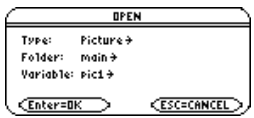
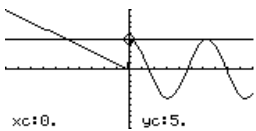
    [F8] 40 [ENTER] 45 [ENTER]

Weitere Darstellungsarten

Stellen Sie im Hauptbildschirm folgende stückweise definierte Funktion graphisch dar: y = ⁻x für x < 0 und y = 5 cos(x), für x ≥ 0. Zeichnen Sie eine waagrechte Linie über den Hochpunkten der Cosinuskurve. Speichern Sie das Bild des dargestellten Graphen.

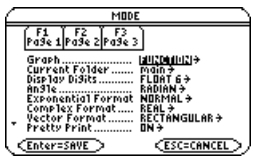
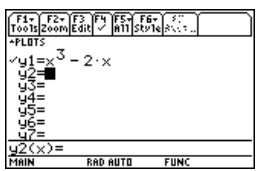
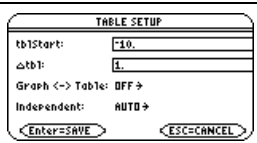
Schritte und Tastenfolgen	Anzeige
1. Öffnen Sie das Dialogfeld **MODE**. Als **Graph**-Modus wählen Sie **FUNCTION**. Als **Angle**-Modus wählen Sie **RADIAN**. Drücken Sie MODE ▷ 1 ▽ ▽ ▽ ▷ 1 ENTER	
2. Wechseln Sie zum Hauptbildschirm. Verwenden Sie den **Graph**-Befehl und die **when** -Funktion, um die stückweise definierte Funktion zu bestimmen. F4 **2** wählt **Graph** aus dem Menüleisten-Menü **Other** und fügt automatisch ein Leerzeichen hinzu. HOME F4 2 2nd [a-lock] WHEN alpha ( ) X 2nd [<] 0 , (-) X , 5 × 2nd [COS] X)) ◆ [CALC HOME] F4 2 WHEN ( ) X 2nd [<] 0 , (-) X , 5 × COS X))	Graph when(x<0,⁻x, 5∗cos(x))

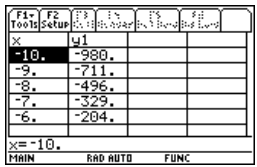
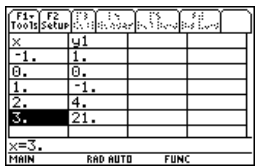
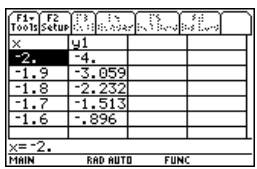
Schritte und Tastenfolgen	Anzeige
3. Führen Sie den **Graph**-Befehl aus, wodurch automatisch der Graphikbildschirm aufgerufen wird. Der **Graph** verwendet die aktuellen Window-Variablen, wobei in diesem Beispiel deren Standardwerte ([F2] **6**) angenommen werden. Drücken Sie [ENTER]	
4. Zeichnen Sie eine waagrechte Linie über der Cosinuskurve. Der Taschenrechner behält den Zeilenmodus bis zur Auswahl eines neuen Befehls bzw. bis zum Drücken von [ESC] bei. [2nd] [F7] 5 ⊙ (bis die Linie plaziert ist) [ENTER] [F7] 5 ⊙ (bis die Linie plaziert ist) [ENTER]	
5. Speichern Sie diese Grafik. Benutzen Sie **PIC1** als Variablenname für das Bild. Stellen Sie auf jeden Fall Type = Picture ein. Standardmäßig ist GDB eingestellt. [F1] 2 ⊙ 2 ⊙ ⊙ PIC [alpha] 1 [ENTER] [ENTER] [F1] 2 ⊙ 2 ⊙ ⊙ PIC1 [ENTER] [ENTER]	

Schritte und Tastenfolgen	Anzeige
6. Löschen Sie die gezeichnete waagrechte Linie. Sie können auch F4 zum Neuzeichnen drücken. 2nd F6 1 F6 1	
7. Laden Sie das gespeicherte Bild, um den Graphen mit der Linie erneut anzuzeigen. Stellen Sie auf jeden Fall Type = Picture ein. Standardmäßig ist GDB eingestellt. Drücken Sie F1 1 ▶ 2 (falls noch nicht angezeigt, stellen Sie auch Variable = pic1 ein) ENTER	 

Tabellen

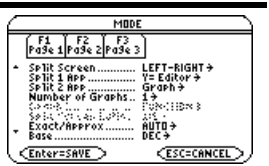
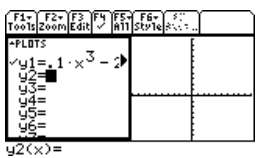
Berechnen Sie die Funktion y= x^3−2x für jede ganze Zahl zwischen ⁻10 und 10. Wie häufig und an welchen Stellen wird das Vorzeichen gewechselt?

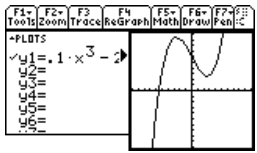
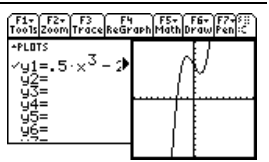
Schritte und Tastenfolgen	Anzeige
1. Öffnen Sie das **MODE**-Dialogfeld. Wählen Sie für den **Graph**-Modus **FUNCTION**. Drücken Sie MODE ▶ 1 ENTER	
2. Öffnen Sie den Y= Editor, und löschen Sie den Inhalt. Definieren Sie dann **y1(x) = x^3 − 2x.** Drücken Sie ♦ [Y=] F1 8 ENTER ENTER X ^ 3 − 2 X ENTER	
3. Stellen Sie die Tabellenparameter auf: **tblStart = ⁻10** **Δtbl = 1** **Graph < - > Table = OFF** **Independent = AUTO** Drücken Sie ♦ [TBLSET] (-) 10 ▼ 1 ▼ ▶ 1 ▼ ▶ 1 ENTER	

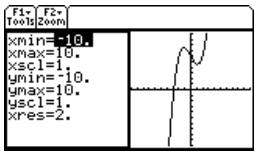
Schritte und Tastenfolgen	Anzeige
4. Öffnen Sie den Tabellen-Bildschirm. Drücken Sie ♦ [TABLE]	
5. Durchlaufen Sie die Tabelle. Beachten Sie, dass y1 bei x = ⁻1, 1, und 2 das Vorzeichen ändert. Verwenden Sie 2nd ⊖ und 2nd ⊕ zum seitenweisen Scrollen. Drücken Sie ⊖ und ⊕ nach Bedarf	
6. Schauen Sie sich die Vorzeichenänderung zwischen x = ⁻2 und x = ⁻1 näher an, indem Sie die Tabellenparameter auf: **tblStart = ⁻2** **∆tbl = .1** Drücken Sie F2 (-) 2 ⊖ .1 ENTER ENTER	

Geteilte Bildschirme

Unterteilen Sie den Bildschirm, um sowohl den Y= Editor als auch den Graphik-Bildschirm anzuzeigen. Untersuchen Sie dann das Verhalten eines Polynoms bei Änderung seiner Koeffizienten.

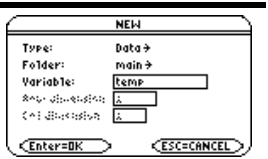
Schritte und Tastenfolgen	Anzeige
1. Öffnen Sie das **MODE** -Dialogfeld. Wählen Sie **FUNCTION** für **Graph**. Wählen Sie **LEFT-RIGHT** für **Split Screen**. Wählen Sie **Y= Editor** für **Split 1 App.** Wählen Sie Graph für **Split 2 App.** Drücken Sie [MODE] ▶ 1 [F2] ▶ 3 ▼ ▶ 2 ▼ ▶ 4 [ENTER]	
2. Löschen Sie den Inhalt des **Y= Editor**, und schalten Sie etwaige Statistik-Daten-Plots aus. Definieren Sie dann **y1(x) = .1x^3−2x+6**. Ein dicker Rahmen um den Y= Editor zeigt an, dass dieser aktiv ist. Ist er aktiv, erstreckt sich dessen Eingabezeile über die gesamte Anzeige. Drücken Sie [F1] 8 [ENTER] [F5] 5 [ENTER] .1 X [^] 3 [−] 2 X [+] 6 [ENTER]	

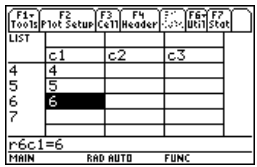
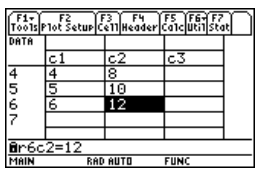
Schritte und Tastenfolgen	Anzeige
3. Wählen Sie das **ZoomStd**-Ansichtfenster, wodurch auf den Graphik-Bildschirm umgeschaltet und die Funktion graphisch dargestellt wird. Der dicke Rahmen befindet sich nun um den Graphik-Bildschirm. Drücken Sie F2 6	
4. Schalten Sie auf den Y= Editor um. Bearbeiten Sie dann y1(x), um $.1x^3$ in $.5x^3$ zu ändern. 2nd [+] ist die Zweitfunktion von APPS. Der dicke Rahmen befindet sich um den **Y= Editor**. Drücken Sie 2nd [+] ▲ ENTER ◀ ▶ ▶ ← 5 ENTER	
5. Schalten Sie auf den Graphik-Bildschirm um, wodurch die bearbeitete Funktion neu erstellt wird. Der dicke Rahmen befindet sich um den Graphik-Bildschirm. Drücken Sie 2nd [+]	

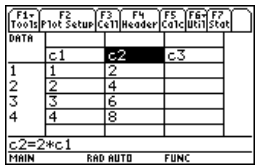
Schritte und Tastenfolgen	Anzeige
6. Schalten Sie auf den Y= Editor um. Öffnen Sie dann an dessen Stelle den Window Editor. Drücken Sie [2nd] [⊞] [♦] [WINDOW]	
7. Öffnen Sie den Hauptbildschirm. Beenden Sie dort die Teilung, und zeigen Sie den Hauptbildschirm in voller Größe an. Drücken Sie [2nd] [QUIT] [2nd] [QUIT]	

Daten/Matrix-Editor

Erstellen Sie mit dem Daten/Matrix-Editor eine einspaltige Listenvariable. Fügen Sie dann eine zweite Informationsspalte hinzu. Beachten Sie, wie die Listenvariable (die nur eine Spalte haben kann) automatisch in eine Datenvariable (die mehrere Spalten haben kann) umgewandelt wird.

Schritte und Tastenfolgen	Anzeige
1. Starten Sie den Daten/Matrix-Editor, um eine neue Listenvariable namens TEMP zu erstellen. Drücken Sie [APPS] … 3 ▶ 3 ▼ ▼ TEMP [ENTER] [ENTER]	

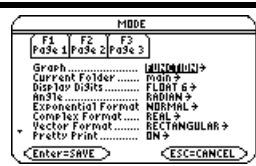
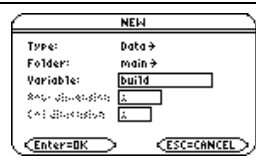
Schritte und Tastenfolgen	Anzeige
2. Geben Sie eine Spalte mit Zahlen ein. Bewegen Sie den Cursor ein Feld nach oben (um zu sehen, daß der Wert eines markierten Feldes in der Eingabezeile angezeigt wird). LIST erscheint in der oberen linken Ecke und weist auf eine Listenvariable hin. Sie können statt [ENTER] auch ⊙ verwenden, um Informationen in ein Feld einzugeben. Drücken Sie 1 [ENTER] 2 [ENTER] 3 [ENTER] 4 [ENTER] 5 [ENTER] 6 [ENTER] ⊙	
3. Gehen Sie zu Spalte 2, und definieren Sie die Kopfzeile der Spalte so, daß Spalte 2 den doppelten Wert von Spalte 1 annimmt. DATA erscheint in der oberen linken Ecke und weist darauf hin, daß die Listenvariable in eine Datenvariable umgewandelt wurde. ▶ ⊙ [F4] 2 [×] [alpha] C 1 [ENTER] ▶ ⊙ [F4] 2 [×] C 1 [ENTER]	 bedeutet, daß sich die Zelle in einer definierten Spalte

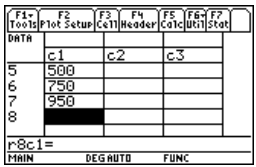
Schritte und Tastenfolgen	Anzeige
4. Gehen Sie zur Kopfzeilenzelle von Spalte 2, damit deren Definition in der Eingabezeile angezeigt wird. Wenn sich der Cursor in der Kopfzeilenzelle befindet, braucht F4 nicht gedrückt werden, um sie zu definieren. In diesem Fall brauchen Sie nur den gewünschten Term einzugeben. Drücken Sie 2nd ▲ ▲	
5. Löschen Sie den Inhalt der Variablen. Einfaches Löschen der Daten wandelt die Datenvariable nicht wieder in eine Listenvariable um. Drücken Sie F1 8 ENTER	

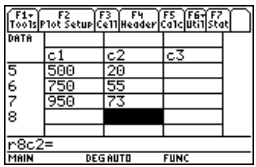
Hinweis: Falls Sie die aktuelle Variable nicht zu speichern brauchen, verwenden Sie sie als Notizblock. Wenn Sie das nächste Mal eine Variable für temporäre Daten brauchen, löschen Sie den Inhalt der aktuellen Variablen und verwenden Sie sie wieder. So können temporäre Daten eingegeben werden, ohne daß jedesmal eine neue Variable erstellt werden muß, die nur Speicherplatz im Rechner belegt.

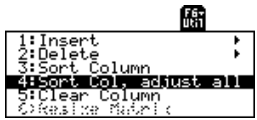
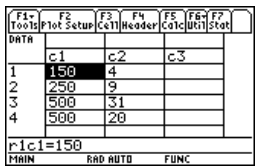
Statistiken und Darstellung von Daten

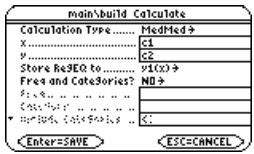
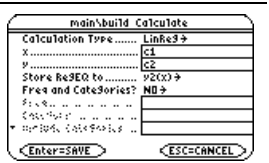
Auf der Grundlage eines Beispiels mit sieben Städten geben Sie Daten ein, die die Bevölkerungszahl mit der Anzahl von Gebäuden mit mehr als 12 Stockwerken in Bezug setzen. Berechnen und zeichnen Sie Gleichungen, die zu diesen Daten passen, mit Hilfe von Med-Med und linearer Regression. Geben Sie für jede Regressionsgleichung die Prognose ab, wieviele Gebäude mit mehr als 12 Stockwerken Sie in einer Stadt mit 300.000 Einwohnern vorfinden würden.

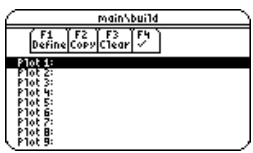
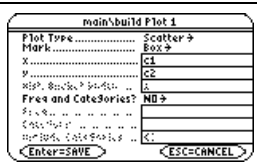
Schritte und Tastenfolgen	Anzeige
1. Aktivieren Sie das Dialogfeld **MODE**. Wählen Sie für **Graph** die Einstellung **FUNCTION**. Drücken Sie [MODE] ▶ 1 [ENTER]	
2. Starten Sie den Daten/Matrix-Editor, um eine neue Datenvariable namens BUILD zu erstellen. Drücken Sie [APPS] … 3 ▼ ▼ BUILD [ENTER] [ENTER]	

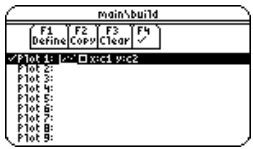
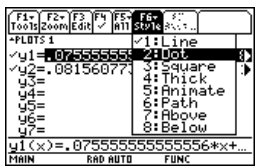
Schritte und Tastenfolgen	Anzeige
3. Geben Sie nach untenstehendem Muster die Bevölkerungszahlen in Spalte 1 ein. Bev. (in 1000) — Gebäude > 12 Stockw. 150 — 4 500 — 31 800 — 42 250 — 9 500 — 20 750 — 55 950 — 73 Drücken Sie 150 [ENTER] 500 [ENTER] 800 [ENTER] 250 [ENTER] 500 [ENTER] 750 [ENTER] 950 [ENTER]	

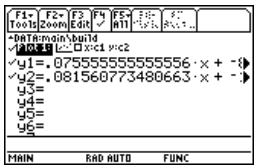
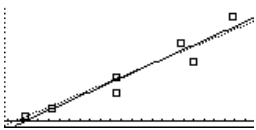
Schritte und Tastenfolgen	Anzeige
4. Bewegen Sie den Cursor auf Zeile 1 in Spalte 2 (r1c2). Geben Sie nun die entsprechende Gebäudezahl ein. ◆ ▲ bewegt den Cursor zum Anfang der Seite. Nach der Eingabe von Daten für eine Zelle können Sie ENTER oder ▼ drücken, um die Daten in die Zelle zu übernehmen und den Cursor um eine Zelle nach unten zu bewegen. Durch Drücken von ▲ werden die Daten übernommen und der Cursor um eine Zelle nach oben bewegt. ▶ ◆ ▲ 4 ENTER 31 ENTER 42 ENTER 9 ENTER 20 ENTER 55 ENTER 73 ENTER ▶ 2nd ▲ 4 ENTER 31 ENTER 42 ENTER 9 ENTER 20 ENTER 55 ENTER 73 ENTER	

Schritte und Tastenfolgen	Anzeige
5. Gehen Sie mit dem Cursor auf Zeile 1 in Spalte 1 (r1c1). Sortieren Sie nun auf Grundlage der Bevölkerungszahl die Daten in aufsteigender Reihenfolge. Dadurch wird Spalte 1 sortiert. Anschließend werden alle anderen Spalten angeglichen, so daß sie die gleiche Reihenfolge wie Spalte 1 beibehalten. Dies ist grundlegend für die Beibehaltung der richtigen Zuordnung zwischen den Datenspalten.	 
Um Spalte 1 zu sortieren, kann sich der Cursor an jeder beliebigen Stelle in Spalte 1 befinden. In diesem Fall drücken Sie ▦ ♦ ⊙; ▦ 2nd ⊙, um die ersten vier Zeilen sehen zu können. ▦ ◀ ♦ ⊙ 2nd [F6] 4 ▦ ◀ 2nd ⊙ F6 4	

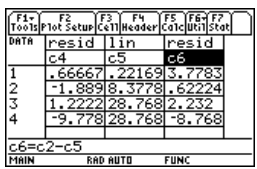
Schritte und Tastenfolgen	Anzeige
6. Aktivieren Sie das Dialogfeld **Calculate**. Setzen Sie: **Calculation Type = MedMed** **x = C1** **y = C2** **Store RegEQ to = y1(x)** [F5] ▷ 7 ▽ C [alpha] 1 ▽ [alpha] C2 ▽ ▷ ▽ [ENTER] [F5] ▷ 7 ▽ C1 ▽ C2 ▽ ▷ ▽ [ENTER]	
7. Führen Sie die Berechnung durch, um die Med-Med Regressionsgleichung darzustellen. Wie im Dialogfeld **Calculate** festgelegt, wird diese Gleichung in y1(x**)** gespeichert. Drücken Sie [ENTER]	
8. Schließen Sie den Bildschirm **STAT VARS**. Der Daten/Matrix-Editor wird angezeigt. Drücken Sie [ENTER]	
9. Rufen Sie das Dialogfeld **Calculate** auf. Setzen Sie: **Calculation Type = LinReg** **x = C1** **y = C2** **Store RegEQ to = y2(x)** Drücken Sie [F5] ▷ 5 ▽ ▽ ▽ ▷ ▽ [ENTER]	

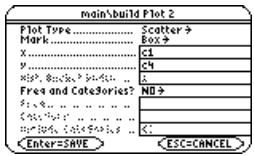
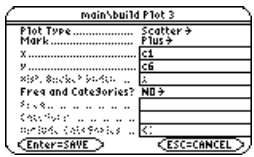
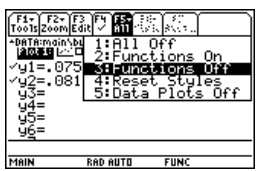
Schritte und Tastenfolgen	Anzeige
10. Führen Sie die Berechnung durch, um die LinReg-Regressionsgleichung darzustellen. Diese Gleichung wird in y2(x) gespeichert. Drücken Sie ENTER	
11. Schließen Sie den Bildschirm **STAT VARS**. Der Daten/Matrix-Editor wird angezeigt. Drücken Sie ENTER	
12. Rufen Sie den Plot Setup-Bildschirm auf. Plot 1 ist standardmäßig markiert. Mit F3 können Sie markierte Plot-Einstellungen löschen. Drücken Sie F2	
13. Legen Sie für Plot 1 fest: **Plot Type = Scatter** **Mark = Box** **x = C1** **y = C2** Beachten Sie die Ähnlichkeit zum Dialogfeld **Calculate**. F1 ▶ 1 ▼ ▶ 1 ▼ C alpha 1 ▼ alpha C2 F1 ▶ 1 ▼ ▶ 1 ▼ C1 ▼ C2	

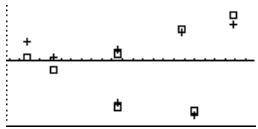
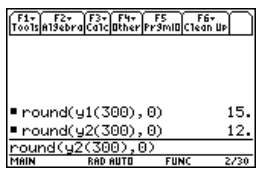
Schritte und Tastenfolgen	Anzeige
14. Speichern Sie die Plotdefinition, und kehren Sie zum Plot Setup-Bildschirm zurück. Beachten Sie die Kurzschreibweise für die Definition von Plot 1. Drücken Sie [ENTER] [ENTER]	
15. Aktivieren Sie den Y= Editor. Setzen Sie für y1(x), die Med-Med Regressionsgleichung, den Anzeigestil auf Dot. **Hinweis:** Je nach vorherigem Inhalt Ihres Y= Editor muss der Cursor eventuell auf y1 bewegt werden.	
PLOTS 1 am oberen Bildschirmrand bedeutet, daß Plot 1 ausgewählt wurde. Beachten Sie, daß y1(x) und y2(x) ausgewählt waren, als die Regressionsgleichungen gespeichert wurden. [♦] [Y=] [2nd] [F6] 2 [♦] [Y=] [F6] 2	

Schritte und Tastenfolgen	Anzeige
16. Scrollen Sie nach oben, um Plot 1 zu markieren. Die abgebildete Kurzdefinition entspricht der des Plot Setup-Bildschirms. Drücken Sie ⊙	
17. Verwenden Sie **ZoomData**, um Plot 1 und die Regressionsgleichungen y1(x) und y2(x) zu zeichnen. **ZoomData** untersucht die Daten aller ausgewählten Statistikplots und passt das Ansichtsfenster an, so daß alle Punkte eingeschlossen sind. Drücken Sie F2 9	
18. Kehren Sie zur aktuellen Sitzung des Daten/Matrix-Editors zurück. Drücken Sie 2nd[⇔]	

Schritte und Tastenfolgen	Anzeige
19. Geben Sie einen Titel für Spalte 3 ein. Definieren Sie den Kopf von Spalte 3 als die von der Med-Med-Geraden vorausgesagten Werte. Um einen Titel eingeben zu können, muss die Titelzelle der Spalte (die oberste Zelle) mit dem Cursor markiert werden. Mit F4 kann von jeder Stelle in einer Spalte aus der Spaltenkopf definiert werden. Befindet sich der Cursor in einer Kopfzelle, muss F4 nicht gedrückt werden. ▶ ▶ ▲ ▲ 2nd [a-lock] MED alpha ENTER F4 Y1 (alpha C1) ENTER ▶ ▶ ▲ ▲ MED ENTER F4 Y1 (C1) ENTER	
20. Geben Sie einen Titel für Spalte 4 ein. Definieren Sie den Kopf von Spalte 4 als Restfehler (Differenz zwischen beobachteten und vorausgesagten Werten, daher der Spaltentitel "resid") für Med-Med. ▶ ▲ 2nd [a-lock] RESID alpha ENTER alpha C2 − alpha C3 ENTER ▶ ▲ RESID ENTER F4 C2 − C3 ENTER	F1 Tools F2 Plot Setup F3 Cell F4 Header F5 Calc F6 Util F7 Stat DATA med resid c2 c3 c4 1 4 3.3333 .66667 2 9 10.889 -1.889 3 31 29.778 1.2222 4 20 29.778 -9.778 c4=c2-c3 MAIN RAD AUTO FUNC

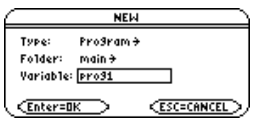
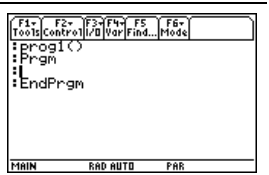
Schritte und Tastenfolgen	Anzeige
21. Geben Sie einen Titel für Spalte 5 ein. Definieren Sie den Kopf von Spalte 5 als die von der LinReg-Geraden vorausgesagten Werte. ▣ ▶ ▲ ▲ 2nd [a-lock] LIN alpha ENTER F4 Y2 (alpha C1) ENTER ⌨ ▶ ▲ LIN ENTER F4 Y2 (C1) ENTER	
22. Geben Sie einen Titel für Spalte 6 ein. Definieren Sie den Kopf von Spalte 6 als Restfehler für LinReg. ▣ ▶ ▲ 2nd [a-lock] RESID alpha ENTER F4 alpha C2 − alpha C5 ENTER ⌨ ▶ ▲ RESID ENTER F4 C2 − C5 ENTER	
23. Rufen Sie den Plot Setup-Bildschirm auf, und heben Sie die Auswahl von Plot 1 auf. Drücken Sie F2 F4	

Schritte und Tastenfolgen	Anzeige
24. Markieren Sie Plot 2, und definieren Sie: **Plot Type = Scatter** **Mark = Box** **x = C1** **y = C4** (MedMed-Restfehler) ▼ F1 ▼ ▼ C alpha 1 ▼ alpha C4 ENTER ENTER ▼ F1 ▼ ▼ C1 ▼ C4 ENTER ENTER	
25. Markieren Sie Plot 3, und definieren Sie: **Plot Type = Scatter** **Mark = Plus** **x = C1** **y = C6** (LinReg-Restfehler) ▼ F1 ▼ ▶ 3 ▼ C alpha 1 ▼ alpha **C6** ENTER ENTER ▼ F1 ▼ ▶ 3 ▼ C1 ▼ C6 ENTER ENTER	
26. Aktivieren Sie den Y= Editor, und schalten Sie alle y(x)-Funktionen aus. Wählen Sie aus F5 **3:Functions Off**, nicht **1:All Off**. Die Plots 2 und 3 sind nach wie vor gewählt. Drücken Sie ♦ [Y=] F5 3	

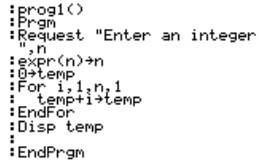
Schritte und Tastenfolgen	Anzeige
27. Verwenden Sie **ZoomData**, um die Restfehler graphisch darzustellen. □ sind die Med-Med-Restfehler; + marks the LinReg residuals. Drücken Sie F2 9	
28. Rufen Sie den Hauptbildschirm auf. HOME ◆ [CALC HOME]	
29. Verwenden Sie die Med-Med (y1(x))- und LinReg (y2(x))-Regressionsgleichungen, um die Werte für x = 300 (Bevölkerung 300.000) zu berechnen. Mit der **round**-Funktion 2nd [MATH] 1 3) wird sichergestellt, daß die resultierende Gebäudeanzahl als ganze Zahl angezeigt wird. Nach der Berechnung des ersten Ergebnisses ändern Sie in der Eingabezeile y1 in y2. Drücken Sie 2nd [MATH] 1 3 Y1 (300) , 0) ENTER ▶ ◀ (acht mal) ← 2 ENTER	

Programmieren

Verfassen Sie ein Programm, das den Benutzer zur Eingabe einer ganzen Zahl auffordert, alle Zahlen von 1 bis zur eingegebenen Zahl addiert und das Ergebnis anzeigt.

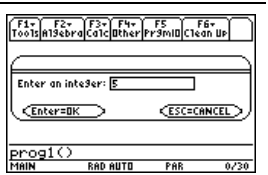
Schritte und Tastenfolgen	Anzeige
1. Beginnen Sie im Program Editor (Programmeditor) ein neues Programm. Drücken Sie [APPS] … 3	
2. Geben Sie PROG1 (ohne Leerzeichen) als Namen der neuen Programmvariablen ein. ▼ ▼ PROG [alpha] 1 ▼ ▼ PROG 1	
3. Öffnen Sie die “Schablone” für ein neues Programm. Der Programmname sowie **Prgm** und **EndPrgm** werden automatisch angezeigt. Zum Abschluss der Eingabe in ein Eingabefeld, z. B. in Variable, müssen Sie [ENTER] zweimal drücken. Drücken Sie [ENTER] [ENTER]	

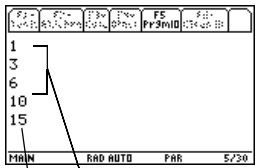
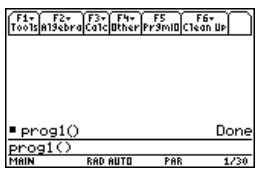
Schritte und Tastenfolgen	Anzeige

4. Geben Sie die folgenden Programmzeilen ein.
 	
`Request "Enter an integer",n`
Zeigt ein Dialogfeld an, das zur Eingabe einer Ganzzahl auffordert, die Eingabe abwartet und die Zahl (als String) in der Variablen n ablegt.
 	
`expr(n)!n`
Wandelt den String in einen numerischen Term um.
 	
`0!temp`
Erzeugt eine Variable mit dem Namen temp und initialisiert sie mit dem Wert 0.
 	
`For i,1,n,1`
Beginnt eine For-Schleife auf der Basis der Variablen i. Beim ersten Schleifendurchlauf ist i = 1. Am Schleifenende wird i um 1 erhöht. Die Schleife wird so oft durchlaufen, bis i > n.
 	
`temp+i!temp`
Addiert den aktuellen Wert von i zu temp.
	

Schritte und Tastenfolgen	Anzeige

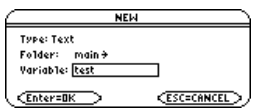
`EndFor`
Markiert das Ende der For-Schleife.
`Disp temp`
Zeigt den Endwert von temp an.
Geben Sie die Programmzeilen wie abgebildet ein. Drücken Sie am Ende jeder Zeile ENTER.	

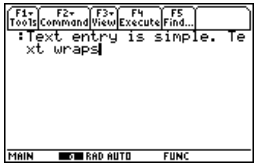
5. Wechseln Sie zum Hauptbildschirm. Geben Sie den Programmnamen gefolgt von einem Paar runder Klammern ein. Sie müssen ( ) stets angeben, auch wenn keine Argumente an das Programm übergeben werden. Das Programm zeigt nun das Dialogfeld mit dem zuvor eingegebenen Aufforderungstext an. HOME 2nd [a-lock] PROG alpha 1 ( ) ENTER ◆ [CALC HOME] PROG1 ( ) ENTER	prog1()
6. Geben Sie ins Dialogfeld 5 ein. Drücken Sie 5	

Schritte und Tastenfolgen	Anzeige
7. Setzen Sie die Programmausführung fort. Der Befehl Disp zeigt das Ergebnis auf dem Programm-I/O-Bildschirm an. Das Ergebnis ist die Summe der ganzen Zahlen von 1 bis 5. Der Programm-I/O-Bildschirm ähnelt zwar dem Hauptbildschirm, dient jedoch nur für Ein-/Ausgabevorgänge im Rahmen eines Programms. Mit dem Programm-I/O-Bildschirm können Sie keine normalen Rechenvorgänge vornehmen. Drücken Sie [ENTER] [ENTER]	
8. Verlassen Sie den Programm-I/O-Bildschirm, und kehren Sie zum Hauptbildschirm zurück. Sie können auch [ESC], [2nd] [QUIT], oder [HOME] [◆] [CALC HOME] drücken, um zum Hauptbildschirm zurückzukehren. Drücken Sie [F5]	

Texteditor

Starten Sie eine neue Texteditor-Sitzung. Üben Sie den Umgang mit dem Texteditor, indem Sie einen beliebigen Text eingeben. Bewegen Sie beim Eingeben den Textcursor, und verbessern Sie eventuelle Tippfehler.

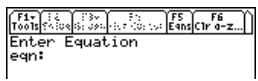
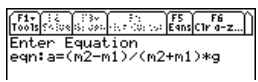
Schritte und Tastenfolgen	Anzeige
1. Starten Sie eine neue Texteditor-Sitzung. Drücken Sie [APPS] … 3	
2. Erstellen Sie eine Textvariable mit dem Namen TEST. In ihr wird automatisch der gesamte Text abgelegt, den Sie während dieser Sitzung eingeben. Benutzen Sie das Verzeichnis MAIN, das im Dialogfeld NEW voreingestellt ist. Zum Abschluss der Eingabe in ein Eingabefeld, z. B. in **Variable**, müssen Sie [ENTER] zweimal drücken. Drücken Sie ▼ TEST [ENTER] [ENTER]	

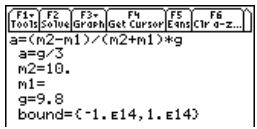
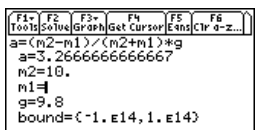
Schritte und Tastenfolgen	Anzeige
3. Geben Sie einen Beispieltext ein. • Um einen einzelnen Großbuchstaben einzugeben, drücken Sie [⇧] und dann den Buchstaben. Nur beim TI-89 Titanium: - Für die Eingabe eines Leerzeichens drücken Sie [alpha] [␣] (Alpha-Funktion der Taste [(-)] key). - Zur Eingabe eines Punktes drücken Sie [alpha], um die Feststellfunktion auszuschalten, dann [.] und erneut [2nd] [a-lock], um die Feststellfunktion wieder einzuschalten.	
Üben Sie das Bearbeiten des Texts wie folgt: • Verschieben Sie den Textcursor mit Hilfe der Cursortasten. • Drücken Sie [←] bzw. [♦] [DEL], um ein Zeichen links bzw. rechts des Cursors zu löschen. [2nd] [a-lock] Geben Sie einen beliebigen Text ein Geben Sie einen beliebigen Text ein	

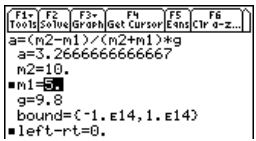
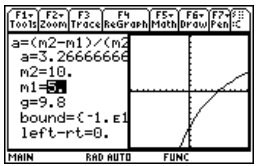
Schritte und Tastenfolgen	Anzeige
4. Beenden Sie den Texteditor, und rufen Sie den Hauptbildschirm auf. Ihre Text-Sitzung wurde bei der Eingabe automatisch gespeichert. Sie brauchen deshalb die Sitzung nicht manuell zu speichern, bevor Sie den Text- Editor beenden. [HOME] [◆] [CALC HOME]	
5. Kehren Sie zur aktuellen Texteditor-Sitzung zurück. Der Text wird exakt so angezeigt wie vor dem Beenden des Editors. Drücken Sie [2nd][⇔]	

Numerischer Gleichungslöser

Ermitteln Sie für die gegebene Gleichung a=(m2−m1)/(m2+m1)∗g, in welcher die bekannten Werte m2=10 und g=9.8 sind, den Wert für m1. Gehen Sie von a=1/3 g aus.

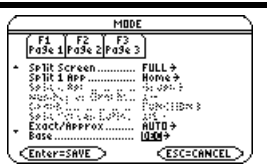
Schritte und Tastenfolgen	Anzeige
1. **Öffnen Sie den** Numeric Solver (Numerischer Gleichungslöser. Drücken Sie [APPS]	
2. Geben Sie die Gleichung ein. Wenn Sie [ENTER] oder ⊙ drücken, werden die in der Gleichung verwendeten Variablen auf dem Bildschirm aufgelistet. [alpha] A [=] [(] [alpha] M2 [−] [alpha] M1 [)] [÷] [(] [alpha] M2 [+] [alpha] M1 [)] [×] [alpha] G [ENTER] A [=] [(] M2 [−] M1 [)] [÷] [(] M2 [+] M1 [)] [×] G [ENTER]	

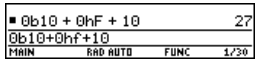
Schritte und Tastenfolgen	Anzeige
3. Geben Sie außer für die unbekannte Variable m1 für alle Variablen Werte ein. Definieren Sie zunächst m2 und g. Definieren Sie anschließend a. (Sie müssen zuerst g definieren, bevor Sie a bezüglich g definieren können.) Akzeptieren Sie die Vorgabe für bound. Wurde eine Variable zuvor bereits definiert, so wird ihr Wert als Standardwert angezeigt. ▼ 10 ▼ ▼ 9.8 ▲ ▲ ▲ alpha G ÷ 3 ▼ 10 ▼ ▼ 9.8 ▲ ▲ ▲ G ÷ 3	
4. Setzen Sie den Cursor auf die unbekannte Variable m1. Sie können auch einen Schätzwert für m1 eingeben. Auch wenn Sie für alle Variablen einen Wert eingeben, wird die numerische Auflösungsfunktion nach der durch den Cursor markierten Variablen auswerten. Drücken Sie ▼ ▼	 g/3 wird ausgewertet, wenn der Cursor von der Linie weg bewegt wird.

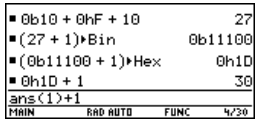
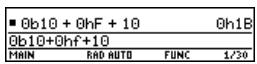
Schritte und Tastenfolgen	Anzeige
5. Lösen Sie nach der unbekannten Variablen auf. Zur Kontrolle der Genauigkeit der Lösung werden die rechte und die linke Seite getrennt ausgewertet. Die Differenz wird als left-rt angezeigt. Bei einer exakten Lösung ist left-rt=0. Drücken Sie F2	 ■ markiert die berechneten Werte
6. Stellen Sie die Lösung in einem **ZoomStd**-Ansichtfenster graphisch dar. Der Graph wird im geteilten Bildschirm angezeigt. Sie können ihn durch Tracen, Zoomen etc. untersuchen. Die vom Cursor markierte Variable (Unbekannte m1) befindet sich auf der x-Achse und left-rt auf der y-Achse. Drücken Sie F3 3	
7. Kehren Sie zum Numerischen Gleichungslöser zurück, und beenden Sie die geteilte Displaydarstellung. Sie können ENTER oder ⊖ drücken, um die Variablenliste erneut anzuzeigen. Drücken Sie 2nd [⇔] F3 2	

Zahlensysteme

Berechnen Sie 10 binär (Basis 2) + F hexadezimal (Basis 16) + 10 dezimal (Basis 10). Verwenden Sie dann den Operator ▶, um eine ganze Zahl von einem Zahlensystem in ein anderes zu überführen. Betrachten Sie dann, wie sich die Änderung der Basis auf das angezeigte Ergebnis auswirkt.

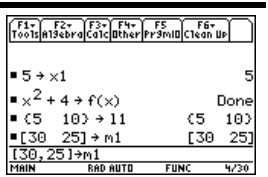
Schritte und Tastenfolgen	Anzeige
1. Zeigen Sie das Dialogfeld **MODE**, Seite 2 an. Wählen Sie für den **Base**-Modus **DEC** als Standardzahlensystem. Ganzzahlige Ergebnisse werden entsprechend dem **Base**-Modus angezeigt. Bruch- und Fließkomma-Ergebnisse werden stets in dezimaler Form angezeigt. Drücken Sie MODE F2 (gehen Sie mit ⊙ zu Base-Modus über) ▶ 1 ENTER	

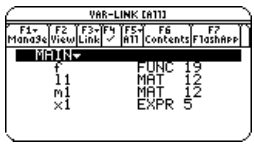
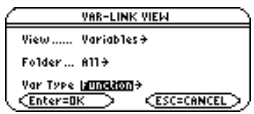
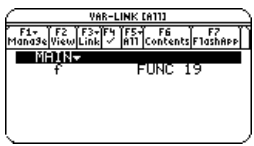
Schritte und Tastenfolgen	Anzeige
2. Berechnen Sie 0b10+0hF+10. Zur Eingabe einer Dual- oder Hexadezimalzahl muß das Präfix 0b bzw. 0h verwendet werden (Null und Buchstabe B bzw. H). Anderenfalls wird die Eingabe als Dezimalzahl behandelt. **Wichtig:** Das Präfix 0b bzw. 0h besteht aus Null (nicht Buchstabe O) und B bzw. H. 0 alpha B 10 + 0 2nd [a-lock] HF alpha + 10 ENTER 0 B 10 + 0 HF + 10 ENTER	
3. Addieren Sie 1 zum Ergebnis, und konvertieren Sie es in binäre Form. 2nd [▶] zeigt den Konvertierungsoperator ▶ an. + 1 2nd [▶] 2nd [a-lock] BIN alpha ENTER + 1 2nd [▶] BIN ENTER	
4. Addieren Sie 1 zum Ergebnis, und konvertieren Sie es in hexadezimale Form. + 1 2nd [▶] 2nd [a-lock] HEX alpha ENTER + 1 2nd [▶] HEX ENTER	

Schritte und Tastenfolgen	Anzeige
5. Addieren Sie 1 zum Ergebnis, und behalten Sie die standardmäßige Dezimalbasis bei. In den Ergebnissen wird zur Kennzeichnung des Zahlensystems das Präfix 0b bzw. 0h verwendet. Drücken Sie + 1 ENTER	
6. Schalten Sie den **Base**-Modus auf **HEX** um. Wenn **Base = HEX** oder **BIN**, dann wird der Absolutwert eines Ergebnisses durch bestimmte Grenzen beschränkt. Drücken Sie MODE F2 (gehen Sie mit ⊙ zu Base-Modus über) ⊙ 2 ENTER	
7. Berechnen Sie 0b10+0hF+10. 0 alpha B 10 + 0 2nd [a-lock] HF alpha + 10 ENTER 0 B 10 + 0 HF + 10 ENTER	
8. Schalten Sie den **Base**-Modus auf **BIN** um. Drücken Sie MODE F2 (gehen Sie mit ⊙ zu Base-Modus über) ⊙ 3 ENTER	
9. Geben Sie erneut 0b10+0hF+10. Drücken Sie ENTER	

Speicher- und Variablenverwaltung

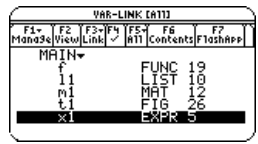
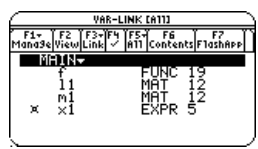
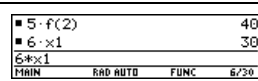
Weisen Sie Variablen Werte mit verschiedenen Datentypen zu. Lassen Sie sich mit dem **VAR-LINK** Bildschirm eine Aufstellung der definierten Variablen anzeigen. Verschieben Sie Variablen dann in den Benutzerarchiv-Speicher, und finden Sie heraus, wie Sie auf die archivierten Variablen zugreifen können (archivierte Variablen werden automatisch gesperrt). Entnehmen Sie dann die Variablen aus dem Archiv, und löschen Sie unbenutzte Variablen, so daß sie nicht unnötig Speicherplatz belegen.

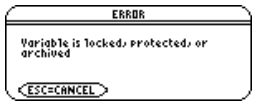
Schritte und Tastenfolgen	Anzeige
1. Weisen Sie im Hauptbildschirm Variablen Werte folgender Variablentypen zu. Term: 5 →x1 Function: x^2+4 →f(x) Liste: {5,10} →l1 Matrix: [30,25] →m1 [HOME] [CLEAR] 5 [STO▸] X1 [ENTER] X [^] 2 [+] 4 [STO▸] [alpha] F [(] X [)] [ENTER] [2nd] [{] 5 [,] 10 [2nd] [}] [STO▸] [alpha] L1 [ENTER] [2nd] [[] 30 [,] 25 [2nd] []] [STO▸] [alpha] M1 [ENTER] [◆] [CALC HOME] [CLEAR] 5 [STO▸] X1 [ENTER] X [^] 2 [+] 4 [STO▸] F [(] X [)] [ENTER] [2nd] [{] 5 [,] 10 [2nd] [}] [STO▸] L1 [ENTER] [2nd] [[] 30 [,] 25 [2nd] []] [STO▸] M1 [ENTER]	

Schritte und Tastenfolgen	Anzeige
2. Nehmen sie an, Sie haben eine Operation begonnen, für die Sie eine Funktionsvariable benötigen, aber deren Namen vergessen. Drücken Sie 5 [×]	5∗
3. Rufen Sie den **VAR-LINK** Bildschirm auf. Dieses Beispiel geht davon aus, dass die oben zugewiesenen Variablen die einzigen bislang definierten sind. Drücken Sie [2nd] [VAR-LINK]	
4. Ändern Sie die Ansicht, so dass nur Variable vom Typ “function” angezeigt werden. Das mag in einem Beispiel mit nur vier Variablen nicht sonderlich nützlich erscheinen, überlegen Sie jedoch, wie sinnvoll es ist, wenn eine Vielzahl Variablen unterschiedlicher Typen definiert sind. Drücken Sie [F2] ▼ ▼ ▶ 5 [ENTER]	 

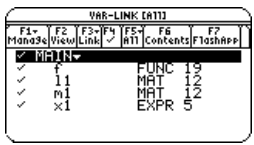
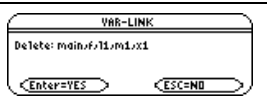
Schritte und Tastenfolgen	Anzeige
5. Markieren Sie die Funktionsvariable f, und zeigen Sie deren Inhalt an. Beachten Sie, dass die Funktion mit f(x) zugewiesen wurde, hier jedoch als **f** angezeigt wird. ▼ 2nd [F6] ▼ F6	x^2+4
6. Schließen Sie das Inhaltsfenster. Drücken Sie ESC	
7. Lassen Sie die Variable **f** weiterhin markiert, schließen Sie den **VAR-LINK** Bildschirm, und kopieren Sie den Namen der Variablen in die Eingabezeile. Beachten Sie, dass " (" eingefügt wurde. Drücken Sie ENTER	5∗f(
8. Schließen Sie die Rechenoperation ab. Drücken Sie 2) ENTER	5∗f(2)

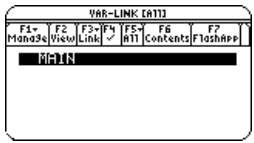
Eine Variable archivieren

Schritte und Tastenfolgen	Anzeige
1. Rufen Sie erneut den **VAR-LINK** Bildschirm auf, und markieren Sie die zu archivierende Variable. Die zuvor vorgenommene Ansichtänderung ist nicht mehr wirksam. Der Bildschirm zeigt die Aufstellung aller definierten Variablen. Drücken Sie [2nd] [VAR-LINK] (verwenden Sie ⊙ zum Markieren von x1)	
2. Archivieren Sie die Variable mit Hilfe des Menüs [F1] **Manage** aus der Menüleiste. ⌑ zeigt an, dass die Variable archiviert ist. Drücken Sie [F1] 8	 
3. Kehren Sie zum Hauptbildschirm zurück, und verwenden Sie die archivierte Variable in einer Rechnung. [HOME] 6 [×] X1 [ENTER] [♦] [CALC HOME] 6 [×] X1 [ENTER]	

Schritte und Tastenfolgen	Anzeige
4. Versuchen Sie, in der archivierten Variablen einen anderen Wert zu speichern. Drücken Sie 10 STO▸ X1 ENTER	
5. Löschen Sie die Fehlermeldung. Drücken Sie ESC	
6. Entnehmen Sie die Variable mit **VAR-LINK** aus dem Archiv. Drücken Sie 2nd [VAR-LINK] (verwenden Sie ⊙ zum Markieren von x1) F1 9	
7. Kehren Sie zum Hauptbildschirm zurück, und speichern Sie einen anderen Wert in der aus dem Archiv entnommenen Variablen. HOME ENTER ◆ [CALC HOME] ENTER	

Variable löschen

Schritte und Tastenfolgen	Anzeige
1. Rufen Sie **VAR-LINK** auf, und wählen Sie mit dem Menüleisten-Menü F5 **All** alle Variablen aus. Das Häkchen ✓ zeigt an, daß ein Eintrag ausgewählt ist. Beachten Sie auch, dass damit gleichzeitig das Verzeichnis **MAIN** ausgewählt wurde. **Hinweis:** Statt F5 zu benutzen (wenn Sie nicht alle Variablen löschen möchten), können Sie auch einzelne Variable auswählen. Markieren Sie jede Variable, die Sie löschen wollen, und drücken Sie F4. Drücken Sie 2nd STO▸ F5 1	 
2. Löschen Sie mit F1. **Hinweis:** Um die markierten Variablen zu löschen, können Sie ← drücken (anstatt F1 **1**). Drücken Sie F1 1	
3. Bestätigen Sie den Löschvorgang. Drücken Sie ENTER	

Schritte und Tastenfolgen	Anzeige
4. Da mit F5 **1** auch das Verzeichnis **MAIN** ausgewählt wurde, erhalten Sie eine Fehlermeldung mit dem Hinweis, dass Sie das Verzeichnis **MAIN** nicht löschen können. Bestätigen Sie die Meldung. Erscheint **VAR-LINK** erneut, werden die gelöschten Variablen nicht mehr angezeigt. Drücken Sie ENTER	
5. Schließen Sie den **VAR-LINK** Bildschirm, und kehren Sie zur aktuellen Anwendung zurück (in diesem Fall zum Hauptbildschirm). Wenn Sie den **VAR-LINK** Bildschirm mit ESC (statt mit ENTER) verlassen, wird der markierte Name nicht in die Eingabezeile übernommen. Drücken Sie ESC	

Die Bedienung des Handheld

Den Handheld ein- und ausschalten

Sie können den TI-89 Titanium / Voyage™ 200 mit den Tasten [ON] und [2nd] [OFF] (oder [♦] [OFF]) manuell ein- und ausschalten. Zur Schonung der Batterien sorgt die APD™-Funktion (Automatic Power Down™) für das automatische Ausschalten des TI-89 Titanium / Voyage™ 200.

So schalten Sie den Handheld ein

Drücken Sie [ON].

- Wurde das Gerät zuvor durch Betätigen von [2nd] [OFF] ausgeschaltet, kehrt der Handheld entweder zum Hauptbildschirm oder zur Apps-Arbeitsfläche zurück.
- Wurde das Gerät zuvor durch Betätigen von [♦] [OFF] oder durch die APD-Funktion automatisch ausgeschaltet, kehrt der Handheld zu der Anwendung zurück, die Sie zuletzt benutzt haben.

So schalten Sie den Handheld aus

Sie können den Handheld mit folgenden Tasten ausschalten.

Drücken Sie:	Beschreibung
2nd [OFF] (drücken Sie 2nd und danach [OFF])	Einstellungen und Speicherinhalte werden durch die Constant Memory™-Funktion erhalten. Beachten Sie jedoch: • Bei Fehlermeldungen können Sie 2nd [OFF] nicht verwenden. • Schalten Sie den TI-89 Titanium / Voyage™ 200 wieder ein, wird stets entweder der Hauptbildschirm angezeigt oder die Apps-Arbeitsfläche (unabhängig von der zuletzt verwendeten Anwendung).
♦ [OFF] (drücken Sie ♦ und danach [OFF])	Wie für 2nd [OFF], außer: • Bei Fehlermeldungen können Sie ♦ [OFF] verwenden. • Schalten Sie den TI-89 Titanium / Voyage™ 200 wieder ein, wird sich das Gerät in demselben Zustand wie vor dem Ausschalten befinden.

Hinweis: [OFF] ist die Zweitfunktion der ON-Taste.

APD (Automatic Power Down / Abschaltautomatik)

Nach einigen Minuten Stillstand wird der Handheld automatisch ausgeschaltet. Diese Funktion heißt APD.

Drücken Sie ON, kehrt der Handheld in genau denselben Zustand zurück, in dem er sich vor dem Abschalten befunden hat.

- Das Display, der Cursor und etwaige Fehlerzustände sind vollkommen unverändert.
- Sämtliche Einstellungen und Speicherinhalte werden beibehalten.

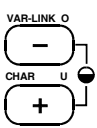
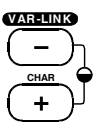
Die APD-Funktion wird während des Ablaufs einer Berechnung oder eines Programms nicht aktiv, es sei denn, das Programm wurde unterbrochen.

Kontrast der Anzeige einstellen

Die Helligkeit und der Kontrast der Anzeige hängen von der Raumbeleuchtung, der Ladung der Batterien, dem Sichtwinkel und der Einstellung des Anzeige-Kontrastes ab. Die Kontrasteinstellung wird bei ausgeschaltetem Gerät im Speicher gehalten.

Kontrast der Anzeige einstellen

Sie können den Anzeige-Kontrast an Ihren Sichtwinkel und die Lichtbedingungen anpassen.

Wirkung:	Drücken und gedrückt halten:
Verstärkung des Kontrastes (dunkler)	♦ und +
Verringerung des Kontrasts (heller)	♦ und −
Kontrast- Tasten 	Kontrast- Tasten 

Sollten Sie ♦ + oder ♦ − zu lange gedrückt halten, könnte das Display vollkommen schwarz oder weiß erscheinen. Eine feinere Regulierung erzielen Sie, indem Sie ♦ gedrückt halten und dann + oder − antippen.

Wann müssen die Batterien ersetzt werden

Mit zunehmender Erschöpfung der Batterien wird die Anzeige heller (insbesondere bei Berechnungen), und Sie müssen den Kontrast verstärken. Wenn Sie den Kontrast häufig nachstellen müssen, sollten die vier Alkalibatterien ausgetauscht werden.

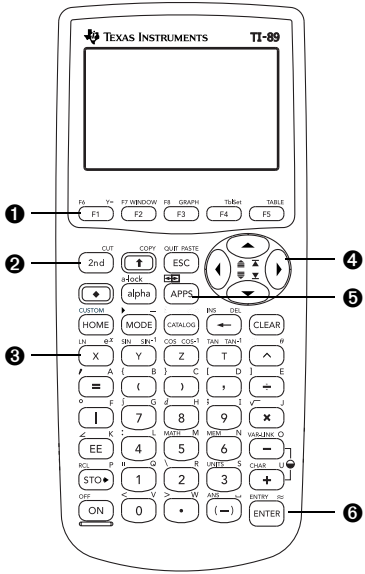
Hinweis: Das Display kann nach dem Auswechseln der Batterien sehr dunkel erscheinen. Stellen Sie es durch ♦ − heller.

Die Statuszeile am unteren Rand des Displays gibt ebenfalls Auskunft über den Zustand der Batterien.

Anzeige in der Statuszeile	Beschreibung
BATT	Batterien schwach.
BATT	Batterien so bald wie möglich ersetzen.

Die TI-89 Titanium Tastatur

Die Funktion der einzelnen Tasten hängt davon ab, ob Sie zuerst eine Modifikatortaste drücken.

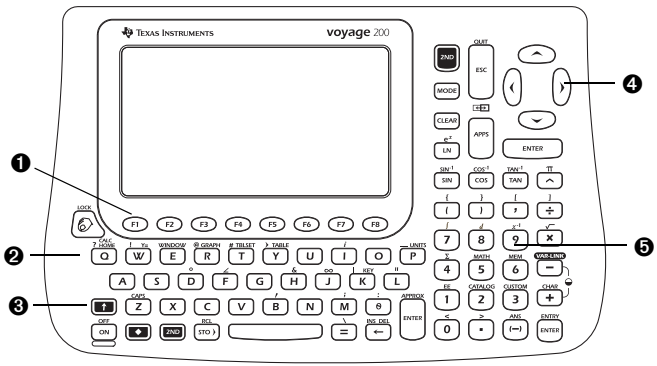


❶ F1 – 2nd F8 offene Symbolleistenmenüs. Wählt Anwendungen (zusammen mit ♦).
❷ 2nd, ♦, ⬆, und alpha vergrößern den Funktionsumfang durch Ergänzen weiterer Tastenbefehle.
❸ X, Y und Z werden häufig in symbolischen Berechnungen verwendet.

❹ ◀, ▶, ▲, und ▼ bewegen den Cursor.
❺ APPS dient zum Wählen einer Anwendung.
❻ ENTER wertet einen Term aus, führt eine Anweisung aus, wählt einen Menüpunkt aus, etc.

Die Voyage™ 200 Tastatur

Dank der handfreundlichen Form und der Tastenanordnung können Sie, selbst wenn Sie das Gerät in beiden Händen halten, schnell auf jedes beliebige Feld der Tastatur zugreifen. Die Tastatur ist in verschiedene Felder zusammengehöriger Tasten aufgeteilt.



❶ F1 – F8 offene Symbolleistenmenüs.
❷ QWERTY-Tastatur funktioniert genau wie eine Computer-Tastatur.
❸ ◆ , 2nd, ⇧ und ⎋ erweitern den Funktionsumfang durch Vergrößerung der Anzahl der verfügbaren Tastenbefehle.

❹ ◀, ▶, ▲ und ▼ verschieben den Cursor.
❺ Die numerische Tastatur liefert mathematische und wissenschaftliche Funktionen.

Den Cursor bewegen

Um den Cursor in eine bestimmte Richtung zu bewegen, drücken Sie die entsprechende Cursortaste (◀, ▶, ▲ oder ▼).

In einigen Anwendungen können Sie außerdem:

- 2nd ◀ oder 2nd ▶ verwenden, um zum Anfang oder zum Ende einer Zeile zu gelangen.
- 2nd ▲ oder 2nd ▼ verwenden, um einen Bildschirm nach oben oder unten zu gelangen.
- ♦ ▲ oder ♦ ▼ verwenden, um an den Anfang oder das Ende einer Seite zu gelangen.
- ▲ und ◀, ▲ und ▶, ▼ und ◀ oder ▼ und ▶ für diagonale Cursorsteuerung. (Beide Cursortasten müssen gleichzeitig gedrückt werden.)

Modifikatortasten

Modifikatortasten

Modifikatortaste	Beschreibung
2nd (zweite)	Ermöglicht den Zugriff auf die Zweitfunktion der anschließend betätigten Taste. Auf der Tastatur sind die Zweitfunktionen in der Farbe der 2nd-Taste angegeben.
♦ (Karo)	Aktiviert Tasten, durch welche Sie bestimmte Anwendungen, Menüpunkte und andere Operationen direkt über die Tastatur wählen können. Auf der Tastatur haben diese Tasten dieselbe Farbe wie die ♦-Taste.

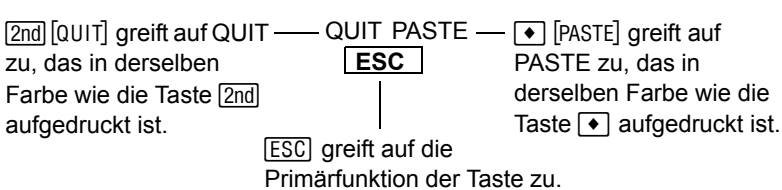
Modifikatortaste	Beschreibung
[↑] (Shift)	Schreibt den nächsten Buchstaben, den Sie drücken, als Großbuchstabe. [↑] wird zusammen mit ▶ und ◀ auch zum Markieren von Zeichen in der Eingabezeile verwendet.
[alpha] (nur 🖩)	Dient zum Eingeben von Buchstaben und Leerzeichen. Auf der Tastatur haben diese dieselbe Farbe wie die Taste [alpha].
[✋] (Hand) (nur 🖳)	In Kombination mit den Cursortasten zur Manipulation geometrischer Objekte. [✋] wird auch zum Zeichnen auf Graphen eingesetzt.

Hinweis: Die Verwendung von [alpha] und [↑] ist detailliert beschrieben.

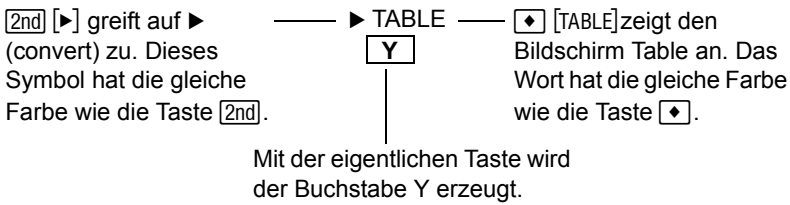
Beispiel für die Modifikatortasten [2nd] und [karo]

Je nachdem, ob Sie zuerst die Taste [2nd] oder [♦] drücken, kann z.B. die Taste [ESC] drei verschiedene Wirkungen haben.

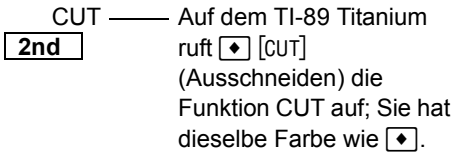
Im folgenden Beispiel für den TI-89 Titanium wird die Verwendung der Modifikatortasten [2nd] oder [♦] mit der Taste [ESC] dargestellt.



Im folgenden Beispiel für den Voyage™ 200 wird die Verwendung der Modifikatortasten [2nd] oder [♦] mit der Buchstabentaste Y dargestellt.



Einige Tasten haben nur eine zusätzliche Funktion, auf die, je nach Farbe des aufgedruckten Funktionsnamens und Position über der Taste, entweder mit der Taste [2nd] oder [♦] zugegriffen wird.



Drücken Sie eine Modifikatortaste wie [2nd] oder [♦], erscheint in der Statuszeile am unteren Fensterrand der Hinweis 2ND bzw. ♦. Wenn Sie versehentlich eine Modifikatortaste betätigen, drücken Sie diese erneut (oder drücken Sie [ESC]), um die Wirkung der Taste aufzuheben.

Weitere wichtige Tasten, mit denen Sie vertraut sein sollten

Taste	Beschreibung
♦ [Y=]	Ruft den Y=Editor auf (*Grundlagen der graphischen Darstellung von Funktionen*).
♦ [WINDOW]	Ruft den Window-Editor auf (*Grundlagen der graphischen Darstellung von Funktionen*).
♦ [GRAPH]	Öffnet den Graphikbildschirm (*Grundlagen der graphischen Darstellung von Funktionen*).
♦ [TBLSET]	Stellt die Parameter für den Tabellenbildschirm ein (*Tabellen*).
♦ [TABLE]	Öffnet den Tabellenbildschirm (*Tabellen*).
: ♦ [CUT] ♦ [COPY] ♦ [PASTE] : ♦ **X** (cut) ♦ **C** (copy) ♦ **V** (paste)	Dient zum Bearbeiten eingegebener Daten durch einen Ausschneide-, Kopier- oder Einfügevorgang.
2nd [⇆]	Wechselt zwischen den letzten beiden Apps oder zwischen den Split-Screen-Abschnitten.
2nd [CUSTOM]	Schaltet das Benutzermenü ein oder aus *(Das Benutzermenü ein- und ausschalten)*.
2nd [▶]	Konvertiert Maßeinheiten (*Konstanten und Maßeinheiten*).

Taste	Beschreibung
[♦] [_] 2nd [_]	Weist eine Maßeinheit zu (*Konstanten und Maßeinheiten*).
←	Löscht das Zeichen links neben dem Cursor (backspace).
2nd [INS]	Schaltet zwischen Einfüge- und Überschreibmodus für die Dateneingabe um (Ein Zeichen einfügen oder überschreiben).
[♦] [DEL]	Löscht das Zeichen rechts neben dem Cursor.
| 2nd [|]	Gibt den "*with* "-Operator ein, der in symbolischen Berechnungen verwendet wird (*Symbolisches Rechnen*).
2nd [∫], 2nd [d]	Führt Integrationen und Ableitungen durch (*Symbolisches Rechnen*).
2nd [∠]	Kennzeichnet einen Winkel bei Polar-, Zylinder- oder Kugelkoordinaten.
2nd [MATH]	Ruft das Menü MATH auf.
2nd [MEM]	Öffnet den MEMORY Bildschirm (Speicher-und Variablen-Verwaltung).
2nd [VAR-LINK]	Öffnet den VAR-LINK Bildschirm für die Verwaltung von Variablen (Speicher-und Variablen-Verwaltung)
2nd [RCL]	Ruft den Inhalt einer Variablen ab (Den Wert einer Variablen abrufen).
2nd [UNITS] [♦] [UNITS]	Zeigt das Dialogfeld UNITS (Einheiten) an (*Konstanten und Maßeinheiten*).

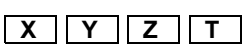
Taste	Beschreibung
2nd [CHAR]	Zeigt das Menü CHAR, über das Sie griechische Buchstaben, international akzentuierte Buchstaben etc. wählen können (*Texteditor*).
2nd [ENTRY], 2nd [ANS]	Ruft die letzte Eingabe bzw. die letzte Antwort ab (Eine frühere Eingabe abrufen und Die letzte Antwort abrufen).

Buchstaben eingeben

Buchstaben werden in Termen wie z.B. $\mathbf{x^2+y^2}$, bei der Eingabe von Variablennamen (Regeln für Variablennamen) und im Texteditor (*Texteditor*) verwendet.

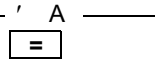
Eingabe von Buchstaben, Satz- und anderen Zeichen auf dem TI-89 Titanium

Die Buchstaben x, y, z und t werden in algebraischen Termen häufig verwendet. Damit Sie diese schnell eingeben können, verfügt die Tastatur über Primärtasten für diese Buchstaben.



Andere Zeichen sind als alpha-Funktionen anderer Tasten verfügbar, für die das zuvor bereits erklärte Prinzip der Modifikatortasten 2nd und ♦ gilt. Beispiel:

2nd [′] gibt ′ ein, das in derselben Farbe wie die Taste 2nd aufgedruckt ist.



alpha [A] gibt ein A ein, das in derselben Farbe wie die Taste alpha aufgedruckt ist.

Eingabe von Buchstaben, Satz- und anderen Zeichen auf TI-89 Titanium / Voyage™ 200

Möchten Sie:	Taste auf :	Taste auf :
Einen einzelnen Kleinbuchstaben eingeben.	alpha und dann die Taste für den jeweiligen Buchstaben (die Statuszeile zeigt a an)	Buchstabentaste
Einen einzelnen Großbuchstaben eingeben.	↑ und dann die Taste für den jeweiligen Buchstaben (die Statuszeile zeigt A an)	↑ und danach die Buchstabentaste (in der Statuszeile steht A)
Ein Leerzeichen eingeben.	alpha [␣] (Alpha-Funktion der Taste ·)	Leertaste
Die Kleinbuchstaben-Feststellfunktion einschalten.	2nd [a-lock] (Statuszeile zeigt a an)	(keine Taste)

Möchten Sie:	Taste auf :	Taste auf :
Die Großbuchstaben-Feststellfunktion einschalten.	[↑] [a-lock] (Statuszeile zeigt A an)	[2nd] [CAPS]
Feststellfunktion ausschalten.	[alpha] (Schaltet Großbuchstabentaste und Kleinbuchstabentaste aus)	[2nd] [CAPS] (schaltet Großbuchstabentaste aus)

Hinweis:

- Um auf dem TI-89 Titanium x, y, z oder t einzugeben, brauchen Sie weder [alpha] noch die Feststellfunktion. Für die Eingabe von X, Y, Z oder T müssen Sie jedoch [↑] oder die Großbuchstaben-Feststellfunktion verwenden.
- Beim TI-89 Titanium ist die Feststellfunktion beim Umschalten zwischen Anwendungen, wie z.B. zwischen Texteditor und Hauptbildschirm stets ausgeschaltet.

Wenn eine der beiden Feststellfunktionen eingeschaltet ist:

- Müssen Sie zum Eingeben eines Punktes, Kommas oder anderer Zeichen, welche die Primärfunktion einer Taste darstellen, die Feststellfunktion ausschalten.
- Müssen Sie zum Eingeben eines Zweitfunktions-Zeichens wie z.B. [2nd] [{] die Feststellfunktion nicht ausschalten. Nach der Eingabe dieses Zeichens bleibt die Feststellfunktion weiterhin aktiv.

Buchstabenautomatik in Dialogfeldern des TI-89 Titanium

In einigen Fällen braucht auf dem TI-89 Titanium die Taste alpha oder 2nd [a-lock] nicht gedrückt zu werden, wenn Buchstaben eingegeben werden sollen. Die Buchstabenautomatikfunktion ist eingeschaltet, wenn zuerst ein Dialogfeld angezeigt wird. Dies gilt für folgende Dialogfelder:

Dialogfeld	Buchstabenautomatik
Katalog Dialogfeld	Befehle werden in alphabetischer Reihenfolge aufgelistet. Durch Drücken einer Buchstabentaste geht der Cursor zum ersten mit diesem Buchstaben beginnenden Befehl.
Einheiten Dialogfeld	Geben Sie in den einzelnen Einheitenkategorien den ersten Buchstaben einer Einheit bzw. einer Konstanten ein. Weitere Hinweise finden Sie in *Konstanten und Maßeinheiten*.
Dialogfelder mit Eingabefeldern	Hierzu gehören beispielsweise: Create New Folder, Rename und Save Copy As.

Hinweis: Zur Eingabe von Zahlen schalten Sie die Buchstabenautomatik mit alpha aus. Mit alpha oder 2nd [a-lock] wechseln Sie zur Eingabe von Buchstaben.

Buchstabenautomatik wird nicht in Dialogfeldern eingeschaltet, in denen nur Zahlen eingegeben werden müssen, wie beispielsweise in Resize Matrix, Zoom Factors und Table Setup.

Wenn Sie Sonderzeichen eingeben müssen

Im Menü [2nd] [CHAR] können Sie aus zahlreichen Sonderzeichen wählen. Weitere Hinweise finden Sie unter “Eingabe von Sonderzeichen” im Modul *Texteditor*.

Zahlen eingeben

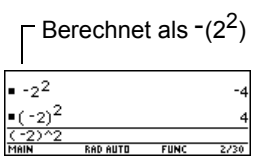
Mit dem Tastenfeld können Sie für Ihre Berechnungen positive und negative Zahlen eingeben. Sie können Zahlen auch in wissenschaftlicher Schreibweise eingeben.

Eine negative Zahl eingeben

1. Drücken Sie die Minustaste [(-)]. (Nicht die Subtraktionstaste [−] verwenden.)
2. Geben Sie die Zahl ein.

Wie der TI-89 Titanium / Voyage™ 200 Graphischer Rechner eine Negation in bezug auf andere Funktionen auflöst, können Sie unter Equation Operating System (EOS™)-Hierarchie im Modul *Technische Referenz* nachlesen. Es ist beispielsweise wichtig, zu beachten, dass Funktionen wie x^2 vor einer Negation ausgewertet werden.

Verwenden Sie Klammern ([(] und [)]), falls Sie im Zweifel sind, wie eine Negation ausgewertet wird.



Sollten Sie [−] anstelle von [(-)] verwenden (oder umgekehrt), erhalten Sie eine Fehlermeldung oder unerwartete Ergebnisse. Beispiel:

- 9 [×] [(-)] 7 = -63
  – aber –
  9 [×] [−] 7 führt zur Anzeige einer Fehlermeldung.
- 6 [−] 2 = 4
  – aber –
  6 [(-)] 2 = -12 weil die Eingabe als 6(-2) interpretiert wird und zur Multiplikation führt.
- [(-)] 2 [+] 4 = 2
  – aber –
  [−] 2 [+] 4 substrahiert 2 von der vorherigen Antwort und addiert dann 4.

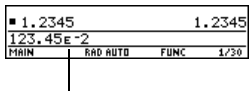
Wichtig: Verwenden Sie [−] zum Substrahieren und [(-)] für die Negation.

Eine Zahl in wissenschaftlicher Schreibweise eingeben

1. Geben Sie den Zahlenteil vor dem Exponenten ein. Dieser Wert kann ein Term sein.
2. Drücken Sie:
   [EE]
   [2nd] [EE]
   Auf dem Display erscheint E.
3. Geben Sie den Exponent als ganze Zahl mit bis zu 3 Stellen ein. Sie können auch negative Exponenten verwenden.

Auch bei Eingabe einer Zahl in wissenschaftlicher Schreibweise wird das Ergebnis nicht in wissenschaflicher oder technischer Schreibweise angezeigt.

Das Displayformat wird durch die Moduseinstellungen und die Größe der Zahl bestimmt.



Steht für 123.45×10^{-2}

Terme und Anweisungen eingeben

Sie führen eine Berechnung eines Terms durch. Sie leiten eine Aktion durch die Ausführung der geeigneten Anweisung ein. Die Berechnung von Termen und die Ergebnisanzeige erfolgt gemäß den Moduseinstellungen

Definitionen

Term	Besteht aus Zahlen, Variablen, Operatoren, Funktionen und deren Argumenten, die eine einzige Antwort ergeben. Beispiel: πr^2+3. • Geben Sie einen Term in derselben Reihenfolge ein, wie er normalerweise geschrieben wird. • Wo die Eingabe eines Wertes erforderlich ist, können Sie meistens einen Term eingeben.
Operator	Führt eine Operation, wie beispielsweise +, –, ∗, ^, durch. • Sowohl vor als auch hinter dem Operator muß ein Argument stehen. Beispiel: 4+5 und 5^2.

Funktion	Gibt einen Wert zurück. • Hinter der Funktion müssen ein oder mehrere Argumente (eingeklammert) stehen. Beispiel: $\sqrt{}(5)$ und **min(5,8)**.
Anweisung	Leitet eine Aktion ein. • Anweisungen können in Termen nicht verwendet werden. • Einige Anweisungen können ohne Argument stehen. Beispiel: ClrHome. • Für andere sind ein oder mehrere Argumente erforderlich. Beispiel: **Circle 0,0,5**. **Hinweis:** Setzen Sie das Argument bei Anweisungen nicht in Klammern.

Hinweis:

- Im Modul *Technische Referenz* werden sämtliche Standard-Funktionen und Anweisungen des TI-89 Titanium / Voyage™ 200 beschrieben.
- In diesem Referenzbuch wird die Bezeichnung Befehl zur allgemeinen Bezugnahme auf Funktionen und Anweisungen verwendet.

Automatische Multiplikation

Der TI-89 Titanium / Voyage™ 200 Graphischer Rechner erkennt eine implizite Multiplikation, sofern sie nicht im Gegensatz zu einer bereits belegten Schreibweise steht.

	Eingabe:	**Interpretation durchTI-89 Titanium / Voyage™ 200:**
Gültig	2π 4 sin(46) 5(1+2) or (1+2)5 [1,2]a 2(a)	2∗π 4∗sin(46) 5∗(1+2) or (1+2)∗5 [a 2a] 2∗a
Ungültig	xy a(2) a[1,2]	Einzelne Variable namens xy Funktionsaufruf Matrixelement a[1,2]

Klammern

Terme werden gemäß der im Modul *Technische Referenz* beschriebenen Equation Operating System (EOS™)-Hierarchie ausgewertet. Möchten Sie die Reihenfolge der Auswertungen ändern oder einfach sicherstellen, dass der Term in der erforderlichen Reihenfolge ausgewertet wird, können Sie Klammern setzen.

Zuerst wird der Inhalt einer Klammer berechnet. Beispielsweise berechnet EOS bei 4(1+2) zunächst (1+2) und multipliziert die Antwort dann mit 4.

Einen Term eingeben

Geben Sie einen Term ein, und drücken Sie dann [ENTER], um ihn auszuwerten. Einen Funktions- oder Anweisungsnamen können Sie folgendermaßen in der Eingabezeile eingeben:

- Drücken Sie, sofern vorhanden, dessen Taste. Drücken Sie:
  [2nd] [SIN]
  [SIN]

  – oder –
- Wählen Sie ihn, sofern vorhanden, aus einem Menü. Wählen Sie beispielsweise **2:abs** aus dem Number-Submenü des MATH-Menüs.

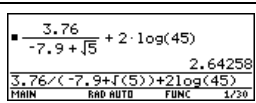
  – oder –
- Geben Sie den Namen buchstabenweise über die Tastatur ein. (Auf dem TI-89 Titanium drücken Sie hierzu [alpha] und [2nd] [a-lock].) Groß- und Kleinbuchstaben können beliebig kombiniert werden; Beispiel: **sin(** oder **Sin(**.

Beispiel

Berechnen Sie
$3.76 \div (-7.9 + \sqrt{5}) + 2 \log 45.$

Tippen Sie den Funktionsnamen für dieses Beispiel ein.

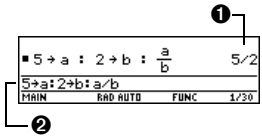
Sie drücken	Anzeige
3.76 [÷] [(] [(-)] 7.9 [+] [2nd] [√]	3.76/(⁻7.9+√([2nd] [√] fügt "√(" ein, weil das Argument in Klammern stehen muß.
5 [)] [)]	3.76/(⁻7.9+√(5)) Drücken Sie [)] einmal zum Schließen von √(5) und erneut, um um (-7.9+ √5) zu schließen.
[+] 2 [2nd] [a-lock] LOG [alpha] [(] 45 [)]	3.76/(⁻7.9+√(5))+2log(45) Die Argumente von log müssen in ( ) stehen.
[ENTER]	$\frac{3.76}{-7.9+\sqrt{5}}+2\cdot\log(45)$ 2.64258 3.76/(-7.9+√(5))+2log(45) MAIN RAD AUTO FUNC 1/30

Sie drücken	Anzeige
3.76 ÷ ((-) 7.9 + 2nd [√]	3.76/(⁻7.9+√(2nd [√] fügt "√(" ein, weil das Argument in Klammern stehen muß.
5))	3.76/(⁻7.9+√(5)) Drücken Sie) einmal zum Schließen von √(5) und erneut, um (⁻7.9 + √5) zu schließen.
+ 2 LOG (45)	3.76/(⁻7.9+√(5))+2log(45) Die Argumente von log müssen in ( ) stehen.
ENTER	

Hinweis: Sie können **log** auch aus dem
Tastaturmodell: CATALOG
Tastaturmodell: 2nd [CATALOG]

Mehrere Terme in einer Zeile eingeben

Möchten Sie mehr als einen Term oder eine Anweisung gleichzeitig eingeben, trennen Sie diese durch einen Doppelpunkt, indem Sie 2nd [:] drücken.

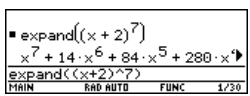


❶ Zeigt nur das letzte Ergebnis an.
❷ → wird angezeigt, wenn Sie einer Variablen anhand von STO▸ einen Wert einspeichern.

Eine Eingabe oder eine Antwort ist zu lang für eine Zeile

Kann im Protokoll-Bereich die Eingabe und deren Antwort nicht in einer Zeile angezeigt werden, wird die Antwort in der nächsten Zeile angezeigt.

Bei Eingaben oder Antworten, deren Länge eine Zeile übersteigt, wird am Ende der Zeile ▶ angezeigt.

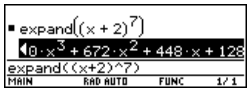


So können Sie sich die gesamte Eingabe oder die gesamte Antwort ansehen:

1. Drücken Sie ▲, um den Cursor von der Eingabezeile in den Protokoll-Bereich hochzusetzen. Dadurch wird die letzte Antwort markiert.

2. Bei Bedarf können Sie die Eingabe oder Antwort, die Sie sehen möchten, durch ▲ und ▼ markieren. Mit ▲ können Sie beispielsweise den Protokoll-Bereich von Antwort zu Eingabe aufwärts durchlaufen.

3. Verwenden Sie zum Scrollen nach rechts und links ▶ und ◀ oder 2nd ▶ und 2nd ◀.



Hinweis: Beim Scrollen nach rechts wird ◀ am Zeilenanfang angezeigt.

4. Möchten Sie zur Eingabezeile zurückkehren, drücken Sie ESC.

Eine Berechnung fortsetzen

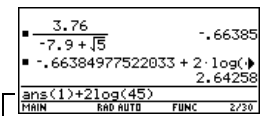
Wenn Sie ENTER eingeben, um einen Term auszuwerten, läßt der TI-89 Titanium / Voyage™ 200 den Term in der Eingabezeile stehen und markiert ihn. Sie können die letzte Antwort weiterverwenden oder einen neuen Term eingeben.

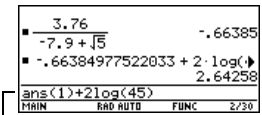
Sie drücken:	**Der** TI-89 Titanium / Voyage™ 200 **führt dies aus:**
+, −, ×, ÷, ^, oder STO▸	Ersetzt die Eingabezeile durch die Variable **ans(1)**. Dadurch können Sie die letzte Antwort als Anfang für einen neuen Term verwenden.
Eine beliebige andere Taste	Löscht die Eingabezeile und beginnt einen neuen Eintrag.

Beispiel

Berechnen Sie 3.76 ÷ (-7.9 + √5). Addieren Sie dann 2 log 45 zum Ergebnis.

TI-89 Titanium

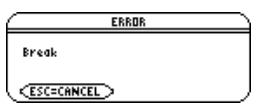
Sie drücken	Anzeige
3.76 [÷] [(] [(-)] 7.9 [+] [2nd] [√] 5 [)] [)] [ENTER] [+] 2 [2nd] [a-lock] LOG [alpha] [(] 45 [)] [ENTER]	 Wenn Sie [+] drücken, wird die Eingabezeile durch die Variable **ans(1)** ersetzt, welche die letzte Antwort enthält.

Sie drücken	Anzeige
3.76 ÷ ((-) 7.9 + 2nd [√] 5)) ENTER + 2 LOG (45) ENTER	 Wenn Sie + drücken, wird die Eingabezeile durch die Variable **ans(1)** ersetzt, welche die letzte Antwort enthält.

Eine Berechnung anhalten

Während eine Berechnung durchgeführt wird, wird rechts in der Statuszeile BUSY angezeigt. Um die Berechnung anzuhalten, drücken Sie ON

Bevor die Meldung “break” angezeigt wird, kann es zu einer Verzögerung kommen.



Drücken Sie ESC, um zur aktuellen Anwendung zurückzukehren.

Formate der angezeigten Ergebnisse

Ein Ergebnis kann in jedem der verfügbaren Formate angezeigt werden. In diesem Abschnitt werden die TI-89 Titanium / Voyage™ 200 Graphischer Rechner-Modi und

deren Einstellungen, die sich auf die Anzeigeformate auswirken, beschrieben. Sie können die aktuellen Moduseinstellungen überprüfen oder verändern.

Pretty Print-Modus

Die Standardeinstellung ist: **Pretty Print = ON**. Exponenten, Wurzeln, Brüche etc. werden in ihrer traditionellen Schreibweise angezeigt. Sie können Pretty Print durch MODE ein- und ausschalten.

Pretty Print	
EIN	**AUS**
$\pi^2, \frac{\pi}{2}, \sqrt{\frac{x-3}{2}}$	π^2, π/2, √((x–3)/2)

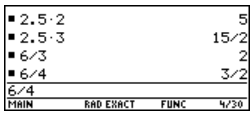
Die Eingabezeile zeigt Terme nicht in Pretty Print an. Bei eingeschaltetem Pretty Print-Modus wird, nachdem Sie ENTER gedrückt haben, im Protokoll-Bereich sowohl die Eingabe als auch die Antwort in Pretty Print angezeigt.

Exact/Approx-Modus

Die Standardeinstellung ist: **Exact/Approx = AUTO**. Mit MODE können Sie eine von drei Einstellungen auswählen.

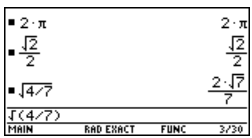
Da es sich bei AUTO um eine Kombination der beiden anderen Einstellungen handelt, sollten Sie mit allen drei Einstellungen vertraut sein.

EXACT — Jedes Ergebnis, bei dem es sich nicht um eine ganze Zahl handelt, wird als Bruch oder in symbolischer Form angezeigt (1/2, π, $\sqrt{2}$, etc.).



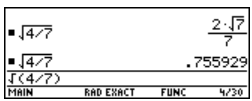
Zeigt ganzzahlige Ergebnisse.

Zeigt Ergebnisse in Form von vereinfachten Brüchen.



Zeigt symbolisches π.

Zeigt die symbolische Form von Wurzeln, die nicht als ganze Zahl bestimmt werden können.

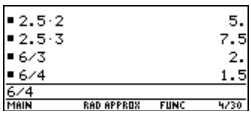


Drücken Sie ◆ ENTER, um die EXACT-Einstellung kurzzeitig zu übergehen und eine Gleitpunktdarstellung einzublenden.

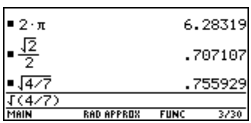
Hinweis: Durch das Beibehalten von Brüchen und symbolischen Formen setzt EXACT die Gefahr von Rundungsfehlern herab, die über Zwischenresultate in Kettenrechnungen entstehen könnten.

APPROXIMATE — Alle numerischen Ergebnisse werden, wo möglich, in Gleitkommaschreibweise (dezimal) angezeigt.

Hinweis: Die Ergebnisse werden mit der Präzision des TI-89 Titanium / Voyage™ 200 gerundet und gemäß den aktuellen Moduseinstellungen angezeigt.



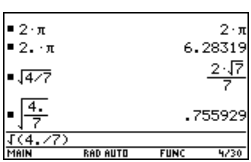
Brüche werden numerisch ausgewertet.



Symbolische Formen werden, wo möglich, numerisch ausgewertet.

Da nicht definierte Variablen nicht ausgewertet werden können, werden sie algebraisch behandelt. Ist beispielsweise die Variable r unbestimmt, dann gilt: $\pi r^2 = 3.14159 \cdot r^2$.

AUTO — Verwendet die EXACT-Form, wo dies möglich ist, und die APPROXIMATE-Form, wenn die Eingabe einen Dezimalpunkt enthält. Außerdem können einige Funktionen die Anzeige eines APPROXIMATE-Ergebnisses bewirken, auch wenn Ihre Eingabe keinen Dezimalpunkt enthält.



Eine Dezimalzahl in der Eingabe führt zu einem Gleitkommaergebnis.

Hinweis: Um die EXACT-Form beizubehalten, sollten Sie Brüche anstelle von Dezimalzahlen verwenden. Schreiben Sie beispielsweise 3/2 anstatt 1.5.

In folgender Tabelle werden die drei Einstellungen miteinander verglichen.

Eingabe	Exact-Ergebnis	Approximate-Ergebnis	Auto-Ergebnis
8/4	2	2.	2
8/6	4/3	1.33333	4/3
8.5∗3	51/2	25.5	25.5
√(2)/2	$\frac{\sqrt{2}}{2}$	.707107	$\frac{\sqrt{2}}{2}$
π∗2	2·π	6.28319	2·π
π∗2.	2·π	6.28319	6.28319

— Im AUTO-Modus erzwingt ein Dezimalbruch in der Eingabe ein Gleitkomma-ergebnis.

Hinweis: Möchten Sie, unabhängig von der aktuellen Einstellung, eine Eingabe in APPROXIMATE-Form auswerten, drücken Sie ♦ ENTER.

Stellenanzeige-Modus Display Digits

Standardeinstellung ist: **Display Digits = FLOAT 6**. Dies bedeutet, daß die Ergebnisse auf maximal sechs Stellen gerundet werden. Sie können MODE verwenden, um andere Einstellungen zu wählen. Die Einstellungen gelten für alle Exponentialformate.

Der TI-89 Titanium / Voyage™ 200 berechnet und erhält intern alle Dezimalergebnisse mit bis zu 14 gültigen Stellen (angezeigt werden aber maximal 12 Stellen).

Einstel-lung	Beispiel		Beschreibung
FIX (0–12)	123. 123.5 123.46 123.457	(FIX 0) (FIX 1) (FIX 2) (FIX 3)	Ergebnisse werden auf die gewählte Anzahl Dezimalstellen gerundet.
FLOAT	123.456789012		Die Anzahl der Dezimalstellen ist je nach Ergebnis unterschiedlich.
FLOAT (1–12)	1.E 2 1.2E 2 123. 123.5 123.46 123.457	(FLOAT 1) (FLOAT 2) (FLOAT 3) (FLOAT 4) (FLOAT 5) (FLOAT 6)	Ergebnisse werden auf die Gesamtanzahl gewählter Stellen gerundet.

Hinweis:

- Unabhängig von der **Display Digits**-Einstellung wird der volle Wert für interne Gleitkommaberechnungen verwendet, um ein Höchstmaß an Genauigkeit zu erzielen.
- Kann der Absolutwert eines Ergebnisses nicht durch die gewählte Stellenzahl angezeigt werden, wird es automatisch in wissenschaftlicher Schreibweise dargestellt.

Exponential Format-Modus

Standardeinstellung ist: **Exponential Format = NORMAL**. Mit MODE können Sie eine von drei Einstellungen wählen.

Einstellung	Beispiel	Beschreibung
NORMAL	12345.6	Kann ein Ergebnis nicht durch die im Display Digits-Modus bestimmte Stellenanzahl dargestellt werden, schaltet TI-89 Titanium / Voyage™ 200 für dieses Resultat von NORMAL auf SCIENTIFIC um.
SCIENTIFIC	1.23456E 4 ❶ ❷	1.23456×10^4
ENGINEERING	1.23456E 3 ❸ ❹	12.3456×10^3

❶ Stets 1 Stelle links vom Komma.
❷ Exponent (Potenz von 10).

❸ Kann 1, 2, oder 3 Stellen links vom Komma aufweisen.
❹ Der Exponent ist ein Vielfaches von 3.

Hinweis: Beträgt der Absolutwert einer Zahl in einer Eingabe weniger als 0,001, so wird diese im Protokoll-Bereich in wissenschaftlicher Schreibweise, also in SCIENTIFIC angezeigt.

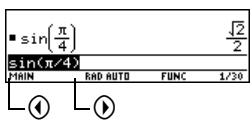
Eine Eingabe in der Eingabezeile bearbeiten

Wenn Sie wissen, wie eine Eingabe bearbeitet wird, können Sie viel Zeit sparen. Die Korrektur eines Eingabefehlers läßt sich meistens schneller erledigen als die Neueingabe des gesamten Terms.

Die Markierung in der Eingabezeile entfernen

Nachdem Sie zum Auswerten eines Terms ENTER gedrückt haben, läßt der TI-89 Titanium / Voyage™ 200 Graphischer Rechner diesen Term in der Eingabezeile stehen und markiert ihn. Möchten Sie diesen Term bearbeiten, müssen Sie zunächst die Markierung entfernen. Anderenfalls könnten Sie den Term durch Überschreiben unabsichtlich löschen.

Bewegen Sie den Cursor zu der Seite des Terms, die bearbeitet werden soll.



- ◀ setzt den Cursor an den Anfang.
- ▶ setzt den Cursor an das Ende des Ausdrucks.

Den Cursor bewegen

Nachdem Sie die Markierung entfernt haben, können Sie den Cursor an die entsprechende Stelle im Term bewegen.

Bewegen des Cursors:	Drücken Sie:	
Nach links oder rechts innerhalb eines Terms.	◀ oder ▶	Halten Sie die Taste gedrückt, um die Bewegung zu wiederholen.
Zum Anfang eines Terms.	2nd ◀	
Zum Ende eines Terms.	2nd ▶	

Hinweis: Sollten Sie versehentlich ▲ anstatt ◀ oder ▶ drücken, wird der Cursor nach oben in den Protokoll-Bereich gesetzt Drücken Sie ESC oder ▼, bis der Cursor wieder in der Eingabezeile erscheint.

Ein Zeichen löschen

Zum Entfernen:	Drücken Sie:	
Das Zeichen links vom Cursor.	←	Halten Sie ← gedrückt, um mehrere Zeichen zu löschen.
Das Zeichen rechts vom Cursor.	♦ ←	

Zum Entfernen:	Drücken Sie:	
Alle Zeichen rechts vom Cursor.	CLEAR (nur einmal)	Befinden sich rechts vom Cursor keine Zeichen, löscht CLEAR die gesamte Eingabezeile.

Die Eingabezeile löschen

Zum Löschen der Eingabezeile drücken Sie:

- CLEAR, wenn der Cursor am Anfang oder Ende der Eingabezeile steht.

  – oder –

- CLEAR CLEAR, wenn der Cursor nicht am Anfang oder Ende der Eingabezeile steht. Durch erste Drücken werden alle Zeichen rechts vom Cursor entfernt, und das zweite Drücken löscht die Eingabezeile.

Ein Zeichen einfügen oder überschreiben

Der TI-89 Titanium / Voyage™ 200 verfügt sowohl über einen Einfügemodus als auch über einen Überschreibmodus. Die Standardeinstellung ist der Einfügemodus. Möchten Sie zwischen Einfüge- und Überschreibmodus umschalten, drücken Sie 2nd [INS].

/ befindet sich im:	Das nächste Zeichen, das Sie eingeben:
	Wird dort eingefügt, wo der Cursor steht.
	Ersetzt das markierte Zeichen.

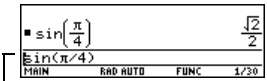
Hinweis: Am Cursor können Sie erkennen, ob Sie, sich im Einfüge- oder im Überschreibmodus befinden.

Mehrere Zeichen ersetzen oder löschen

Markieren Sie zunächst die entsprechenden Zeichen. Ersetzen oder löschen Sie dann alle markierten Zeichen.

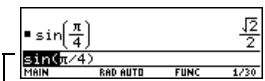
Mehrere Zeichen markieren:

1. Setzen Sie den Cursor auf eine Seite des zu markierenden Zeichens.



Um sin(durch cos(zu ersetzen, bringen Sie den Cursor neben sin.

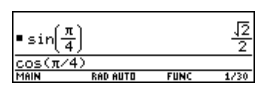
2. Halten Sie ⇧ gedrückt, und drücken Sie ◀ oder ▶, um Zeichen links oder rechts vom Cursor zu markieren.



Halten Sie ⇧ gedrückt, und drücken Sie ▶ ▶ ▶ ▶.

Die markierten Zeichen ersetzen oder entfernen:

1. Geben Sie die neuen Zeichen ein.
2. Drücken Sie ←.



Hinweis: Wenn Sie zu ersetzende Zeichen markieren, beachten Sie, daß einige Funktionstasten automatisch eine Klammer öffnen.

Menüs

Um die Tastatur überschaubar zu lassen, verwendet das Handheld Menüs für den

Zugriff auf viele Operationen. Dieser Abschnitt gibt einen Überblick, wie ein Unterpunkt aus einem Menü ausgewählt wird. Einzelne Menüs werden in den entsprechenden Moduln dieses Handbuchs beschrieben.

Ein Menü anzeigen

Drücken Sie:	Es wird angezeigt:
F1, F2, etc.	Ein Menüleisten-Menü — Erscheint unterhalb der Menüleiste am oberen Rand der meisten Anwendungsbildschirme. Über dieses Menü können Sie verschiedene für die Anwendung nützliche Operationen wählen.
APPS	Apps-Arbeitsfläche oder **APPLICATIONS**-Menü — Hier wählen Sie eine Anwendung aus einer Liste aus.
2nd [CHAR]	**CHAR**-Menü — Über dieses Menü können Sie Sonderzeichen (griechische Zeichen, mathematische Symbole etc.) auswählen.
2nd [MATH]	**MATH**-Menü — Über dieses Menü können Sie mathematische Operationen auswählen.
CATALOG 2nd [CATALOG]	**CATALOG**-Menü — Über dieses Menü können Sie aus einer vollständigen, alphabetisch geordneten Liste der TI-89 Titanium / Voyage™ 200 -Standard-Funktionen und -Anweisungen wählen. Ferner können hiermit benutzerdefinierte oder Flash-Anwendungsfunktionen gewählt werden (sofern solche definiert oder geladen wurden).

Drücken Sie:	Es wird angezeigt:
2nd [CUSTOM]	**CUSTOM**-Menü — Dieses Benutzermenü können Sie individuell gestalten, indem Sie alle verfügbaren Funktionen, Anweisungen oder Zeichen darin auflisten. Der TI-89 Titanium / Voyage™ 200 enthält ein vorgegebenes Benutzermenü, das Sie bearbeiten oder neu definieren können. Nähere Informationen zum Menü Custom finden Sie in den Modulen *Startbildschirm des Rechners und/oder* Programmieren.

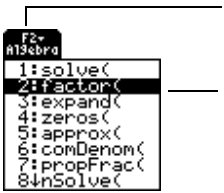
Einen Menüpunkt auswählen

So wählen Sie einen Punkt aus einem angezeigten Menü:

- Drücken Sie die Zahl oder den Buchstaben links neben dem Menüpunkt. Für einen Buchstaben beim TI-89 Titanium drücken Sie zunächst alpha und dann die jeweilige Buchstabentaste.

  – oder –

- Markieren Sie den Menüpunkt mit den Cursortasten (▼ und ▲), und drücken Sie dann ENTER. (Beachten Sie, daß die Markierung durch Betätigung von ▲ auf dem ersten Punkt auf den letzten Punkt übergeht und umgekehrt.)



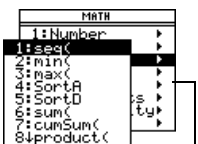
▾ bedeutet, daß sich unterhalb der Menüleiste ein Menü öffnet, wenn Sie F2 drücken.

Möchten Sie **factor** wählen, drücken Sie 2 oder ⊙ ENTER. Dadurch wird das Menü geschlossen und die Funktion dort eingefügt, wo sich der Cursor befindet.

factor(

Menüpunkte mit der Markierung ▶ (Untermenüs)

Wählen Sie einen Menüpunkt, der mit ▶ endet, wird ein Untermenü angezeigt. Sie können dann einen Punkt aus dem Untermenü wählen.



Aus Platzgründen überlagert der TI-89 Titanium diese Menüs.

↓ bedeutet, daß Sie durch Abwärtsscrollen mit dem Cursor zusätzliche Punkte finden.

Bei List wird zum Beispiel ein Untermenü geöffnet, über welches Sie eine spezifische Listenfunktion wählen können.

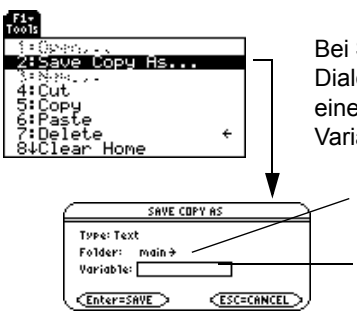
Für Menüpunkte mit Untermenü können Sie, wie nachfolgend beschrieben, das Cursorfeld verwenden.

- Möchten Sie das Untermenü des markierten Punktes anzeigen, drücken Sie ⊙. (Dies kommt der Wahl des Menüpunktes gleich.)

- Möchten Sie das Untermenü ausblenden, ohne einen Punkt zu wählen, drücken Sie ◀. (Dies kommt der Betätigung von ESC gleich.)
- Um den Cursor von der ersten direkt zur letzten Option eines Menüs zu steuern, drücken Sie ▲. Mit ▼ steuern Sie den Cursor von der letzten zur ersten Menüoption.

Menüpunkte mit ". . ." (Dialogfeld)

Wenn Sie einen Menüpunkt wählen, der ". . ." (Auslassungszeichen) enthält, wird ein Dialogfeld geöffnet, in welches Sie zusätzliche Informationen eingeben können.



Bei **Save Copy As ...** wird zum Beispiel ein Dialogfenster geöffnet, das Sie zur Eingabe eines Verzeichnis- und eines Variablennamens auffordert.

→ zeigt an, daß Sie ▶ drücken können, um aus einem Menü auszuwählen.

Ein Eingabefeld zeigt an, daß eine Eingabe verlangt wird. (Buchstabenautomatik ist auf dem TI-89 Titanium automatisch eingeschaltet.

Nach dem Ausfüllen eines Eingabefelds müssen Sie zweimal ENTER drücken, um die Eingabe zu speichern und die Dialog-Box zu schließen.

Ein Menü schließen

Möchten Sie das aktuelle Menü schließen, ohne einen Menüpunkt zu wählen, drücken Sie [ESC]. Wenn verschiedene Untermenüs geöffnet sind, müssen Sie [ESC] mehrmals betätigen, bis alle angezeigten Menüs geschlossen sind.

Zwischen Menüleisten-Menüs umschalten

So können Sie von einem Menüleisten-Menü auf ein anderes umschalten, ohne einen Menüpunkt zu wählen:

- Drücken Sie die Taste ([F1], [F2] etc.) des gewünschten Menüleisten-Menüs.
  – oder –
- Schalten Sie mit dem Cursorfeld zum nächsten (drücken Sie ▶) oder vorigen (drücken Sie ◀) Menüleisten-Menü um. Durch Betätigung von ▶ im letzten Menü gelangen Sie ins erste und umgekehrt.

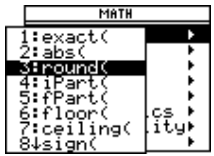
Wenn Sie ▶ verwenden, darf kein Menüpunkt, der über ein Untermenü verfügt, markiert sein. Ansonsten wird durch ▶ nicht das nächste Menüleisten-Menü sondern das Untermenü dieses Punktes angezeigt.

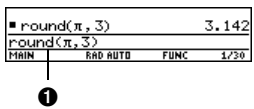
Beispiel: Einen Menüpunkt wählen

Runden Sie den Wert von π auf drei Dezimalstellen. Ausgangspunkt ist die leere Eingabezeile auf dem Hauptbildschirm:

1. Drücken Sie 2nd [MATH], um das **MATH**-Menü zu öffnen.

2. Drücken Sie **1**, um das **Number**-Untermenü zu öffnen. (Oder drücken Sie ENTER, da der erste Menüpunkt automatisch markiert wird.)
3. Drücken Sie **3**, um **round** zu wählen. (Oder ▼ ▼ und ENTER.)
4. Drücken Sie 2nd [π] , **3**) und dann ENTER, um den Term auszuwerten.





❶ Durch Wahl der Funktion in Schritt 3 wird **round(** automatisch in die Eingabezeile geschrieben.

Eine Anwendung auswählen

Der TI-89 Titanium / Voyage™ 200 Graphischer Rechner verfügt über verschiedene Anwendungen, dank derer Sie unterschiedliche Probleme lösen und untersuchen können. Sie können eine Anwendung aus einem Menü wählen, von der Apps-Arbeitsfläche aus, oder auf häufig gebrauchte Anwendungen direkt über die Tastatur zugreifen.

Aus dem APPLICATIONS-Menü

1. Wenn die Apps-Arbeitsfläche deaktiviert ist, drücken Sie APPS, um ein Menü mit der Liste der Anwendungen zu öffnen.

   Hinweis: Möchten Sie das Menü schließen, ohne einen Punkt zu wählen, drücken Sie ESC.

2. Wählen Sie eine Anwendung. Entweder:

   - Verwenden Sie zum Markieren der Anwendung die Cursortasten ⊙ oder ⊙, und drücken Sie dann ENTER.

     – oder –

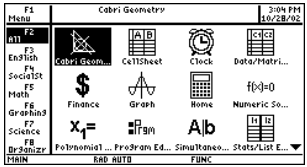
   - Drücken Sie die Nummer der gewünschten Anwendung.

Anwendung:	Dient zum:
FlashApps	Anzeige einer Liste von Flash-Anwendungen (falls vorhanden).
Y= Editor	Definieren, Bearbeiten und Auswählen von Funktionen oder Gleichungen zum Zeichnen von Graphen (siehe Module *Grundlagen der graphischen Darstellung von Funktionen, Parameterdarstellungen*, *Polardarstellung*, *Graphische Darstellung von Folgen*, *3D-Darstellungen* und *Darstellung von Differentialgleichungen*).
Window Editor	Einstellen der Fenstergröße für die Anzeige eines Graphen (*Grundlagen der graphischen Darstellung von Funktionen*).

Anwendung:	Dient zum:
Graph	Anzeigen von Graphen (*Grundlagen der graphischen Darstellung von Funktionen*).
Table	Anzeigen einer Tabelle von Funktionswerten (*Tabellen*).
Data/Matrix Editor	Eingeben und Bearbeiten von Listen, Daten und Matrizen. Sie können statistische Berechnungen durchführen und Statistik-Plots zeichnen (*Daten/Matrix Editor* und *Statistiken und Darstellung von Daten*).
Program Editor	Eingeben und Bearbeiten von Programmen und Funktionen (*Programmierung*).
Text Editor	Eingeben und Bearbeiten von Text (*Texteditor*).
Numeric Solver	Geben Sie einen Term oder Gleichung ein, definieren Sie Werte für alle, bis auf eine Variable und lösen Sie anschließend nach dieser Unbekannten auf (*Numerischer Gleichungslöser*).
Home	Eingeben von Termen und Anweisungen sowie zum Durchführen von Berechnungen.

Von der Apps-Arbeitsfläche aus

Wählen Sie mit den Cursortasten ein Anwendungssymbol auf der Apps-Arbeitsfläche und drücken Sie ENTER. Entweder wird daraufhin die Anwendung sofort geöffnet oder ein Dialogfeld angezeigt. (Ihre Apps–Arbeitsfläche kann anders aussehen, als die unten abgebildete.)



Das am häufigsten verwendete Dialogfeld enthält die folgenden Optionen für die Anwendung:

Option	Beschreibung
Current	Kehrt zu dem zuletzt angezeigten Bildschirm der App zurück. (Wenn keine Datei/Variable new verfügbar ist, ist die Vorgabe für diese Option beim Drücken von ENTER New.)
Open	Ermöglicht die Auswahl einer vorhandenen Datei.
New	Erstellt eine neue Datei mit dem im Eingabefeld angegebenen Namen.

Wählen Sie eine Option und drücken Sie ENTER. Die Anwendung wird angezeigt.

Hinweis: Der allgemeine Begriff *variable* wird für die Bezeichnung der von Ihnen erstellten Anwendungsdaten-Dateien verwendet.

Sie können mit jeder der folgenden Methoden aus einer Anwendung zur Apps-Arbeitsfläche zurückgelangen:

- Drücken Sie APPS.
- Drücken Sie im Vollbildmodus 2nd [QUIT].

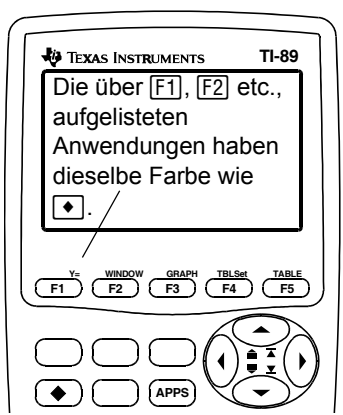
- Drücken Sie im Split-Screen-Modus 2nd [QUIT], um den Vollbildschirm zu öffnen und drücken Sie dann noch einmal 2nd [QUIT].

Drücken Sie 2nd [↔], um von der Apps-Arbeitsfläche zur letzten geöffneten Anwendung zurückzukehren.

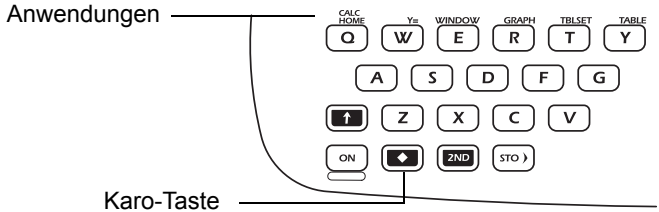
Über die Tastatur

Über die Tastatur können Sie auf die gebräuchlichsten Anwendungen zugreifen. Beispiel: ♦ [Y=] kommt der Betätigung von ♦ und dann F1 gleich. In diesem Handbuch wird die Notation ♦ [Y=] verwendet (ähnlich wie bei Zweit-Funktionen).

Anwendung:	Drücken Sie:
Home	HOME 2nd [CALC HOME]
Y= Editor	♦ [Y=]
Window Editor	♦ [WINDOW]
Graph	♦ [GRAPH]
Table Setup	♦ [TBLSET]
Table Screen	♦ [TABLE]



Beim Voyage™ 200, sind einige häufig genutzte Anwendungen über den QWERTY-Tasten angegeben.



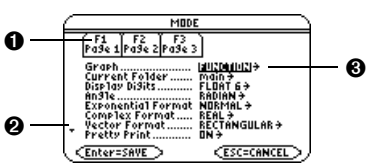
Hinweis: 2nd-Tasten-Funktionen werden in der obigen Grafik nicht angezeigt.

Betriebsarten einstellen

Die Betriebsarten oder Modi bestimmen, wie Zahlen und Graphen angezeigt und interpretiert werden. Modus-Einstellungen werden bei ausgeschaltetem Gerät durch die Constant Memory™-Funktion erhalten. Alle Zahlen, einschließlich der Matrizen- und Listenelemente, werden gemäß den aktuellen Modus-Einstellungen angezeigt.

Modus-Einstellungen kontrollieren

Drücken Sie MODE zum Öffnen des MODE-Dialogfelds mit der Liste der Betriebsarten und deren aktuellen Einstellungen.



❶ Es liegen drei Seiten mit Moduslisten vor. Drücken Sie F1, F2, oder F3 zur Sofortanzeige einer bestimmten Seite.
❷ Bedeutet, dass durch Abwärtsrollen weitere Modi angezeigt werden.
❸ Bedeutet, dass Sie mit ▶ oder ◀ ein Menü öffnen und daraus wählen können.

Hinweis: Momentan ungültige Betriebsarten werden unscharf angezeigt. Auf der zweiten Seite ist beispielsweise **Split 2 App** ungültig, wenn **Split Screen = FULL** ist. Durchlaufen Sie die Liste, überspringt der Cursor die unscharf angezeigten Einstellungen.

Modus-Einstellungen ändern

Im MODE-Dialogfeld:

1. Markieren Sie die zu ändernde Modus-Einstellung. Verwenden Sie ▼ oder ▲ (mit F1, F2, oder F3) zum Durchlaufen der Liste.
2. Drücken Sie ▶ oder ◀, um ein Menü mit den gültigen Einstellungen zu öffnen. Die aktuelle Einstellung ist markiert.
3. Wählen Sie die geeignete Einstellung. Entweder:
   - Markieren Sie die Einstellung mit ▼ oder ▲, und drücken Sie ENTER.

     – oder –

   - Drücken Sie die Zahl oder den Buchstaben dieser Einstellung.

   Hinweis: Möchten Sie ein Menü schließen und zum **MODE** Dialogfeld zurückkehren, ohne einen Menüpunkt zu wählen, drücken Sie ESC.
4. Ändern Sie bei Bedarf weitere Modus-Einstellungen.
5. Nach Abschluß aller Änderungen drücken Sie ENTER, um die Änderungen zu speichern und das Dialogfeld zu verlassen.

Wichtig: Wenn Sie zum Verlassen des **MODE**-Dialogfelds ESC anstatt ENTER drücken, werden sämtliche Modus-Änderungen rückgängig gemacht.

Überblick über die Betriebsarten

Hinweis: Detaillierte Informationen zu einem bestimmten Modus finden Sie im entsprechenden Abschnitt des vorliegenden Referenzbuchs.

Modus	Beschreibung
Graph	Art des darzustellenden Graphen: FUNCTION, PARAMETRIC, POLAR, SEQUENCE, 3D, oder DE.
Current Folder	Verzeichnis zum Speichern und Abrufen von Variablen. Haben Sie keine zusätzlichen Verzeichnisse angelegt, ist nur das MAIN-Verzeichnis verfügbar. Siehe “Unabhängige Variablen-Sätze in Verzeichnissen speichern” im Modul *Startbildschirm des Rechners*.
Display Digits	Maximale Stellenanzahl (FLOAT) oder feststehende Anzahl Dezimalstellen (FIX) für die Anzeige eines Gleitkommaergebnisses. Unabhängig von der Einstellung können bei einem Gleitkommaergebnis keinesfalls mehr als insgesamt 12 Stellen angezeigt werden.
Angle	Einheiten, in welchen Winkelwerte interpretiert und angezeigt werden: RADIAN oder DEGREE.
Exponential Format	Schreibweise für die Ergebnisanzeige: NORMAL, SCIENTIFIC oder ENGINEERING.
Complex Format	Format für die Anzeige komplexer Ergebnisse (wenn vorhanden): REAL (komplexe Ergebnisse werden nicht angezeigt, es sei denn, Sie verwenden eine komplexe Eingabe), RECTANGULAR oder POLAR.

Modus	Beschreibung
Vector Format	Format für die Anzeige von Vektoren mit 2 und 3 Elementen: RECTANGULAR, CYLINDRICAL, oder SPHERICAL.
Pretty Print	Dient zum Ein- oder Ausschalten der Pretty Print-Anzeige.
Split Screen	Unterteilt den Bildschirm in zwei Hälften und bestimmt die Anordnung dieser Teile: FULL (ungeteilter Bildschirm), TOP-BOTTOM oder LEFT-RIGHT. Siehe Modul *Geteilte Bildschirme*.
Split 1 App	Anwendung in der oberen oder linken Hälfte eines geteilten Bildschirms. Verwenden Sie keinen geteilten Bildschirm, ist dies die aktuelle Anwendung.
Split 2 App	Anwendung in der unteren oder rechten Hälfte eines geteilten Bildschirms. Diese Einstellung ist nur bei einem geteilten Bildschirm aktiv.
Number of Graphs	Richtet bei einem unterteilten Bildschirm beide Seiten ein, um unabhängige Graphen-Sätze anzuzeigen.
Graph 2	Wenn **Number of Graphs = 2**, wird die Art des Graphen in der zweiten Hälfte des geteilten Bildschirms gewählt. Siehe Modul *Startbildschirm des Rechners*.
Split Screen Ratio	Teilungsverhältnis der beiden Hälften eines geteilten Bildschirms: 1:1, 1:2 oder 2:1. (Nur)
Exact/Approx	Berechnet Terme und zeigt Ergebnisse in numerischer oder in rationaler/symbolischer Form an: AUTO, EXACT oder APPROXIMATE.

Modus	Beschreibung
Base	Hiermit können Sie Berechnungen durch Eingabe von Zahlen im Dezimal- (DEC), Hexadezimal- (HEX) oder Binärformat (BIN) durchführen.
Unit System	Erlaubt Ihnen die Auswahl aus drei Maßeinheitensystemen für die Vorgabeeinheiten der angezeigten Ergebnisse: **SI** (metrisch oder MKS); **Eng/US** (Fuß, Pfund, etc.); oder **Custom**.
Custom Units	Hier können Sie benutzerdefinierte Standardeinstellungen auswählen. Dieser Modus ist so lange unscharf, bis Sie Einheitensystem **3:CUSTOM** auswählen.
Language	Sprachauswahl für TI-89 Titanium / Voyage™ 200 (eine oder mehrere Sprachen in Abhängigkeit von den installierten Flash-Anwendungen für Sprachen).
Apps Desktop	Schaltet die Apps–Arbeitsfläche AN oder AUS.

Verwendung des Löschmenüs zum Beginn einer neuen Aufgabe

Im Hauptbildschirm können Sie mit dem **Clean Up**-Menü in einem definierten Zustand eine neue Berechnung starten, ohne den Speicher des TI-89 Titanium / Voyage™ 200 Graphischer Rechner zurücksetzen zu müssen.

Clean Up-Menü

Rufen Sie auf dem Hauptbildschirm mit folgenden Tasten das Menü **Clean Up** auf:

[2nd] [F6]

[F6]

Menüelement	Beschreibung
Clear a–z	Löscht alle aus einem Zeichen bestehenden Variablennamen im aktuellen Verzeichnis, sofern sie nicht gesperrt oder archiviert sind. Sie werden zur Bestätigung des Vorgangs mit [ENTER] aufgefordert.
	Aus einem Zeichen bestehende Variablennamen werden häufig in symbolischen Berechnungen wie der folgenden verwendet: **solve(aøx^2+bøx+c=0,x)** Wenn einer der Variablen bereits ein Wert zugewiesen wurde, kann eine Berechnung zu unerwarteten Ergebnissen führen. Um dies zu vermeiden, können Sie vor Beginn der Berechnung **1:Clear a-z** drücken.

Menüelement	Beschreibung
NewProb	Fügt **NewProb** in die Eingabezeile ein. Zur Ausführung des Befehls müssen Sie dann ENTER drücken. **NewProb** nimmt verschiedene Operationen vor, durch welche eine neue Aufgabe in einem definierten Zustand begonnen werden kann, ohne dass Sie den Speicher zurücksetzen müssen: • Löscht alle aus einem Zeichen bestehenden Variablennamen im aktuellen Verzeichnis (ebenso wie **1:Clear a-z**), sofern sie nicht gesperrt oder archiviert sind. • Schaltet alle Funktionen und Statistikplots im aktuellen Graphik-Modus aus (**FnOff** und **PlotsOff**). • Führt **ClrDraw**, **ClrErr**, **ClrGraph**, **ClrHome**, **ClrIO** und **ClrTable** durch.
Restore custom default	Ist ein anderes als das vorgegebene Benutzermenü aktiv, können Sie hiermit das vorgegebene wiederherstellen. Weitere Informationen zum Menü Custom finden Sie im Modul *Startbildschirm des Rechners*.

Hinweis:

- Benutzen Sie zum Definieren einer Variablen, die Sie beibehalten möchten, mehr als ein Zeichen im Namen. Dadurch verhindern Sie, daß sie versehentlich durch **1:Clear a–z** gelöscht wird.
- Informationen zum Kontrollieren und Rücksetzen des Speichers oder anderer Systemvorgaben finden Sie unter *Speicher-und Variablen-Verwaltung*.

Verwendung des Catalog Dialogs

Mit Hilfe des CATALOG können Sie auf alle vorhandenen TI-89 Titanium / Voyage™ 200-Befehle (Funktionen und Anweisungen) in einer Liste zugreifen. Darüber hinaus können im Dialogfeld CATALOG benutzerdefinierte Funktionen oder Funktionen von Flash-Anwendungen gewählt werden (sofern solche geladen bzw. definiert sind).

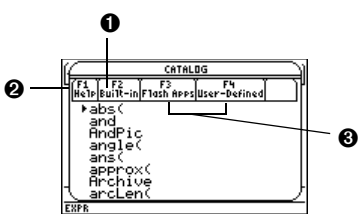
Anzeige des Dialogfelds CATALOG

Dieses Dialogfeld wird mit folgenden Tasten angezeigt:

CATALOG

2nd [CATALOG]

CATALOG ruft standardmäßig F2 **Built-in** auf; hiermit wird eine alphabetische Liste der vorinstallierten TI-89 Titanium / Voyage™ 200 -Befehle (Funktionen und Anweisungen) angezeigt.



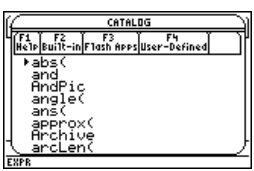
❶ F2 Built-in wird standardmäßig angezeigt.
❷ F1 Help zeigt die Parameter eines Befehls in einem Dialogfeld an.
❸ F3 und F4 ermöglichen Zugriff auf Flash-Anwendungs- und benutzerdefinierte Funktionen.

Hinweis: Derzeit nicht verfügbare Optionen sind unscharf. Beispiel: F3 **Flash Apps** ist unscharf, wenn keine Flash-Anwendung installiert ist. F4 **User-Defined** ist unscharf, wenn keine Funktion bzw. kein Programm definiert wurde.

Aus dem CATALOG wählen

Wählen Sie einen Befehl, wird dessen Name neben dem Cursor in die Eingabezeile eingefügt. Deshalb sollten Sie den Cursor vor der Wahl des Befehls an die erforderliche Position bringen.

1. Drücken Sie
   - CATALOG
   - 2nd [CATALOG]
2. Drücken Sie F2 **Built-in**.



- Die Befehle sind in der Liste alphabetisch geordnet. Befehle, die nicht mit einem Buchstaben beginnen (+, %, √, Σ etc.), befinden sich am Ende der Liste.
- Möchten Sie **CATALOG** verlassen, ohne einen Befehl zu wählen, drücken Sie ESC.

Hinweis: Wird die Liste mit vorhandenen Funktionen und Befehlen zum ersten Mal angezeigt, beginnt sie beim ersten Wort. Bei nächstmaliger Anzeige startet sie an der Stelle, an der sie verlassen wurde

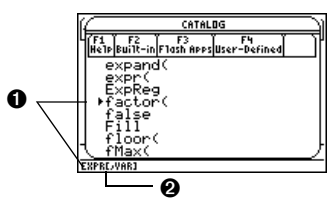
3. Setzen Sie die Marke ▶ neben den Befehl, und drücken Sie [ENTER].

Um den ▶ Anzeiger zu bewegen:	Drücken oder geben Sie ein:
Um je einen Befehl	⊙ oder ⊙
Um je eine Seite	[2nd] ⊙ oder [2nd] ⊙
Zum ersten Befehl, der mit einem bestimmten Buchstaben beginnt	Den Buchstaben. (Drücken Sie beim dem TI-89 Titanium nicht zuerst [alpha]. Sonst müssen Sie [alpha] oder [2nd] [a-lock] erneut drücken, um einen Buchstaben eingeben zu können.)

Hinweis: Drücken Sie ⊙ um vom Anfang direkt zum Ende zu kommen. Drücken Sie ⊙ um vom Ende direkt zum Anfang zu kommen.

Hilfe-Informationen zu Parametern

Für Befehle, die mit ▶ gekennzeichnet sind, werden in der Statuszeile gegebenenfalls die erforderlichen und optionalen Parameter und deren Art angegeben.



❶ Gekennzeichneter Befehl und dessen Parameter.
❷ Klammern [ ] stehen für optionale Parameter.

In obengenanntem Beispiel ist die Syntax für factor:

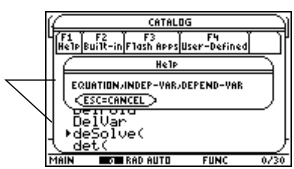
factor(*expression***)**	erforderlich
– oder –	
factor(*expression,variable***)**	optional

Hinweis: Genaue Angaben zu den Parametern finden Sie in der Beschreibung dieses Befehls im Modul *Technische Referenz*.

Anzeigen der KATALOG-Hilfe

Sie können die Parameter eines Befehls durch betätigen der Taste F1 **Help** in einem Dialogfeld anzeigen lassen. Diese Parameter stimmen mit denen in der Statuszeile überein.

Gezeigter Befehl und seine Parameter.



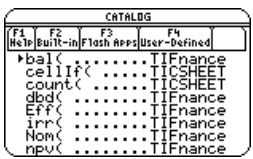
Einige Befehle, wie **ClrDraw**, benötigen keine Parameter. Wenn Sie einen dieser Befehle wählen, werden in der Statuszeile keine Parameter angezeigt und bei Betätigung von F1 **Help** erscheint die Anzeige Unavailable.

Drücken Sie ESC, um das Dialogfeld CATALOG Help zu verlassen.

Auswahl einer Flash-Anwendungsfunktion

Flash-Anwendungen können eine oder mehrere Funktionen enthalten. Bei Auswahl einer Funktion wird der Funktionsname an der Cursorposition in die Eingabezeile eingefügt. Daher sollte der Cursor vor Auswahl einer Funktion an die richtige Position gebracht werden.

1. Drücken Sie folgende Tasten:
   - CATALOG
   - 2nd [CATALOG]
2. Drücken Sie F3 **Flash Apps**. (Diese Option ist unscharf, wenn keine Flash-Anwendungen auf dem TI-89 Titanium / Voyage™ 200 installiert sind.)



- Die Funktionsnamen sind in der Liste alphabetisch sortiert. In der linken Spalte werden die Funktionsnamen aufgelistet, in der rechten die Flash-Anwendung, zu der die Funktion gehört.
- In der Statuszeile stehen Hinweise zu den einzelnen Funktionen.
- Mit ESC wird der Bildschirm ohne Auswahl einer Funktion ausgeblendet.

3. Bewegen Sie den Zeiger ▶ zur gewünschten Funktion, und drücken Sie ENTER.

Steuern des Zeigers ▶:	**Taste:**
Funktionsweise	⊙ oder ⊙
Seitenweise	2nd ⊙ oder 2nd ⊙

Steuern des Zeigers ▶:	Taste:
Zur ersten mit dem Buchstaben beginnenden Funktion	Den Buchstaben. (Drücken Sie beim dem TI-89 Titanium nicht zuerst alpha. Sonst müssen Sie alpha oder 2nd [a-lock] erneut drücken, um einen Buchstaben eingeben zu können.)

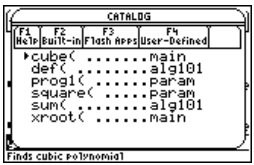
Auswahl einer benutzerdefinierten Funktion bzw. eines Programms

Sie können selbst Funktionen oder Programme erstellen und mit F4 **User-Defined** darauf zugreifen. Hinweise zum Erzeugen von Funktionen finden Sie im Modul *Startbildschirm des Rechners* unter “Erstellen und Berechnen benutzerdefinierter Funktionen” sowie im Modul *Programmierung* unter “Funktionseingabe”. *Programmierung* enthält auch Hinweise zum Erstellen und Ausführen eines Programms.

Nach Auswahl einer Funktion bzw. eines Programms wird der entsprechende Name an der Cursorposition in die Eingabezeile eingefügt. Daher sollte der Cursor vor Auswahl einer Funktion bzw. eines Programms an die richtige Position gebracht werden.

1. Drücken Sie folgende Tasten:
   CATALOG
   2nd [CATALOG]

2. Drücken Sie F4 **User-Defined**. (Diese Option ist unscharf, wenn keine Funktion bzw. kein Programm definiert wurde.)



- Die Funktions- bzw. Programmnamen sind alphabetisch sortiert. In der linken Spalte befinden sich die Funktions- bzw. Programmnamen, in der rechten das Verzeichnis mit der Funktion bzw. dem Programm.
- Enthält die erste Funktions- bzw. Programmzeile einen Kommentar, wird dieser in der Statuszeile angezeigt.
- Mit ESC wird der Bildschirm ohne Auswahl einer Funktion bzw. eines Programms geschlossen.

Hinweis: Im **VAR-LINK** Bildschirm können Variablen, Verzeichnis und Flash-Anwendungen verwaltet werden (siehe *Speicher-und Variablen-Verwaltung*).

3. Bewegen Sie den Zeiger ▶ zu einer Funktion bzw. einem Programm, und drücken Sie ENTER.

Um den ▶ Anzeiger zu bewegen:	**Drücken oder geben Sie ein:**
Eine einzelne Funktion oder Programm	⊙ oder ⊙
Seitenweise	2nd ⊙ oder 2nd ⊙

Um den ▶ Anzeiger zu bewegen:	Drücken oder geben Sie ein:
Zur ersten mit dem Buchstaben beginnenden Funktion bzw. dem Programm	Den Buchstaben. (Drücken Sie beim dem TI-89 Titanium nicht zuerst [alpha]. Sonst müssen Sie [alpha] oder [2nd] [a-lock] erneut drücken, um einen Buchstaben eingeben zu können.)

Variablenwerte speichern und abrufen

Möchten Sie einen Wert speichern, speichern Sie ihn als eine benannte Variable. Sie können in Termen dann den Namen anstelle des Wertes verwenden. Trifft der TI-89 Titanium / Voyage™ 200 Graphischer Rechner den Namen in einem Term an, wird der gespeicherte Wert der Variablen eingesetzt.

Regeln für Variablennamen

Ein Variablenname:

- Kann 1 bis 8 Stellen bestehend aus Buchstaben und Ziffern umfassen. Dies schließt griechische Buchstaben (außer π), akzentuierte und internationale Buchstaben mit ein.
  - Leerzeichen sind nicht erlaubt.
  - Das erste Zeichen darf keine Ziffer sein.
- Kann Groß- und Kleinbuchstaben beinhalten. Die Namen **AB22**, **Ab22**, **aB22** and **ab22** beziehen sich alle auf dieselbe Variable.

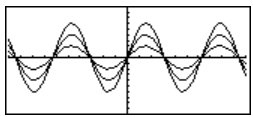
- Muß von den vom Gerät bereits zugeteilten Namen abweichen. Bereits zugeteilte Namen umfassen:
  - Integrierte Funktionen (wie **abs**) und Anweisungen (wie **LineVert**). Siehe Modul *Technische Referenz*.
  - System-Variablen (z.B. **xmin** und **xmax**, die zum Speichern von Graph-Werten verwendet werden). Eine Liste finden Sie im Modul *Technische Referenz*.

Beispiele

Variable	Beschreibung
myvar	OK.
a	OK.
Log	Nicht OK; Name ist bereits der **log**-Funktion zugeteilt.
Log1	OK.
3rdTotal	Nicht OK; beginnt mit einer Ziffer.
circumfer	Nicht OK; mehr als 8 Zeichen.

Datenarten

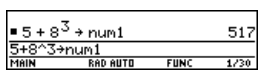
Datenarten	Beispiele
Terme	2.54, 1.25E6, 2π, xmin/10, $2+3i$, $(x-2)^2$, $\sqrt{2}/2$
Listen	{2 4 6 8}, {1 1 2}

Datenarten	Beispiele
Matrizen	$\begin{bmatrix}1 & 0 & 0\end{bmatrix}$, $\begin{bmatrix}1 & 0 & 0\\3 & 4 & 6\end{bmatrix}$
Zeichenketten	“Hello”, “The answer is:”, “xmin/10”
Bilder	
Funktionen	myfunc(arg), ellipse(x,y,r1,r2)

Einen Wert in einer Variablen speichern

1. Geben Sie den zu speichernden Wert ein, bei dem es sich auch um einen Term handeln kann.
2. Drücken Sie STO▸. Das Speichersymbol (→) erscheint.
3. Schreiben Sie den Variablennamen.

   Hinweis: TI-89 Titanium-Benutzer sollten bei der Eingabe von Variablen bei Bedarf alpha drücken.

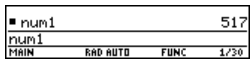


4. Drücken Sie ENTER.

Zum kurzzeitigen Speichern eines Werts in einer Variable können Sie den “mit”-Operator verwenden. Siehe “Werte und Einstellungs-Restriktionen ersetzen” in *Symbolisches Rechnen*.

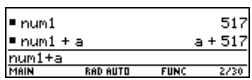
Eine Variable anzeigen

1. Schreiben Sie den Variablennamen.
2. Drücken Sie [ENTER].



Bei einer nicht definierten Variablen wird im Ergebnis der Variablenname angegeben.

In diesem Beispiel ist die Variable a nicht definiert. Deshalb wird sie als eine symbolische Variable verwendet.

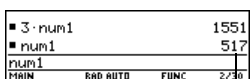


Hinweis: In *Symbolisches Rechnen* finden Sie Informationen zum symbolischen Rechnen.

Eine Variable in einem Term verwenden

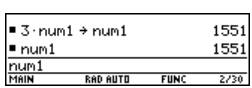
1. Schreiben Sie den Variablennamen in den Term.
2. Drücken Sie [ENTER] zum Auswerten des Terms.

   Hinweis: Eine Liste der vorhandenen Variablennamen können Sie, wie in *Speicherund Variablen-Verwaltung* erläutert, mit [2nd] [VAR-LINK] anzeigen.



Der Variablenwert wurde nicht geändert.

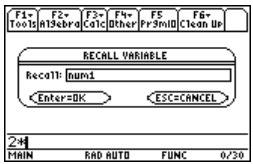
Soll das Ergebnis den vorigen Wert der Variablen ersetzen, müssen Sie das Ergebnis speichern.



Den Wert einer Variablen abrufen

Es kann vorkommen, dass Sie in einem Term anstelle des Variablennamens den tatsächlichen Wert der Variablen verwenden möchten.

1. Drücken Sie [2nd] [RCL], um ein Dialogfeld zu öffnen.
2. Schreiben Sie den Variablennamen.
3. Drücken Sie zweimal [ENTER].

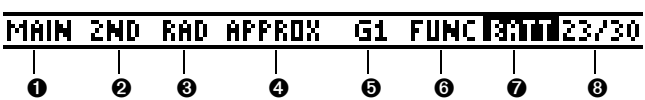


In diesem Beispiel wird der in **num1** gespeicherte Wert neben dem Cursor in die Eingabezeile eingesetzt.

Statuszeilen-Anzeigen im Display

Die Statuszeile wird am unteren Rand aller Anwendungsbildschirme eingeblendet. Sie enthält Informationen über den aktuellen Status des TI-89 Titanium / Voyage™ 200 Graphischer Rechner einschließlich einiger wichtiger Modus-Einstellungen.

Statuszeilen-Anzeigen



❶ Aktuelles Verzeichnis
❷ Modifikator-taste

❸ Winkel-Modus
❹ Exact/Approx-Modus
❺ Graph-Nummer
❻ Graph-Modus
❼ Batterien austauschen
❽ History-Paare, Busy/Pause, Gesperrte Variable

Anzeige	Bedeutung
Aktuelles Verzeichnis	Zeigt den Namen des aktuellen Verzeichnisses an. Siehe “Unabhängige Variablen-Sätze in Verzeichnissen speichern” im Modul *Startbildschirm des* Rechners. MAIN ist der Standardordner.
Modifikator-Taste	Zeigt wie unten beschrieben an, welche Modifikatortaste aktiv ist.
2nd	[2nd] — Verwendet die Zweitfunktion der nächsten Taste, die Sie drücken.
◆	[◆] — Greift auf die Karo-Funktion der nächsten Taste zu, die Sie drücken.
⬆	[⬆] — Schreibt den nächsten Buchstaben als Großbuchstaben. Auf dem TI-89 Titanium, können Sie mit [⬆] einen Buchstaben eingeben, ohne [alpha] benutzen zu müssen.
(▯)	[alpha] — Der TI-89 Titanium schreibt für die nächste Buchstabentaste, die Sie drücken, einen Kleinbuchstaben.

Anzeige	Bedeutung
	2nd [a-lock] — Die Kleinbuchstaben-Feststellfunktion ist aktiv. Bis Sie diese ausschalten, schreibt der TI-89 Titanium für alle Tasten, die Sie drücken, den entsprechenden Kleinbuchstaben. Zum Ausschalten der Feststellfunktion drücken Sie alpha.
	⇧ alpha — Die Großbuchstaben-Feststellfunktion ist aktiv. Bis Sie diese ausschalten, schreibt der TI-89 Titanium für alle Tasten, die Sie drücken, den entsprechenden Großbuchstaben. Zum Ausschalten der Feststellfunktion drücken Sie alpha.
	Im Zusammenhang mit einer Cursortaste nutzt der Voyage™ 200 alle für die Grafik- und Geometriedarstellung verfügbaren ”Ziehen”-Leistungsmerkmale.
Angle Mode	Gibt an, in welchen Einheiten Winkelwerte interpretiert und angezeigt werden. Den Angle-Modus können Sie mit der Taste MODE ändern.
RAD	Bogenmaß.
DEG	Grad
Exact/Approx-Modus	Gibt an, wie Antworten berechnet und angezeigt werden. Den Exact/Approx-Modus können Sie mit der Taste MODE ändern.
AUTO	Automatisch
EXACT	Genau
APPROX	Angenähert

Anzeige	Bedeutung
Graph-Nummer	Wurde der Bildschirm für die Anzeige zweier unabhängiger Graphen unterteilt, gibt dies an, welcher Graph aktiv ist — **G1** oder **G2**. (Anzeige von **GR#1** oder **GR#2** auf dem Voyage™ 200.)
Graph-Modus	Gibt an, welche Art von Graphen gezeichnet werden können. (Den Graph-Modus können Sie mit der Taste MODE ändern).
FUNC	**y(x)**-Funktionen
PAR	**x(t)**- und **y(t)**-Parameterdarstellungen
POL	**r(θ)**-Polargraphen
SEQ	**u(n)** Folgen
3D	**z(x,y)** 3D-Graphen
DE	**y'(t)** Differentialgleichungen
Batterie	Wird nur angezeigt, wenn die Batteriespannung zu niedrig wird. Wird BATT mit einem schwarzen Hintergrund angezeigt, ersetzen Sie die Batterien so schnell wie möglich.
Protokollpaare/Busy/Pause, Gesperrte Variable	Welche Informationen in diesem Bereich der Statuszeile angezeigt werden, hängt von der aktuellen Anwendung ab.
23/30	Wird nur auf dem Hauptbildschirm angezeigt, um über die Anzahl der Eingabe-/Antwort-Paare im Protokoll-Bereich Auskunft zu erteilen. Beachten Sie die History Information auf der Statuszeile des moduls *Startbildschirm des Rechners*.

Anzeige	Bedeutung
BUSY	Eine Berechnung wird durchgeführt oder ein Graph gezeichnet.
PAUSE	Sie haben das Zeichnen eines Graphen oder ein Programm angehalten.
🔒	Die im aktuellen Editor (Daten/Matrix-Editor, Programm-Editor oder Texteditor) geöffnete Variable ist gesperrt oder archiviert und kann nicht geändert werden.

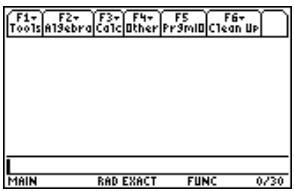
Hinweis:

- Um [2nd], [♦], [alpha] oder [⇧] zu enfernen, drücken Sie dieselbe Taste erneut, oder drücken Sie eine andere Modifikatortaste.
- Verfügt die nächste Taste, die Sie drücken, nicht über eine Karo-Funktion oder einen Buchstaben, dann hat die Taste ihre normale Wirkung.

Hauptbildschirm des Rechners

Hauptbildschirm des Rechners

Der Hauptbildschirm des Rechners ist der Startpunkt für mathematische Operationen, inklusive der Ausführung von Befehlen, Berechnung von Ausdrücken und Anzeige der Ergebnisse.



Ein leerer Hauptbildschirm des Rechners

Dieses Modul beschreibt die Bestandteile des Hauptbildschirms des Rechners; Blättern und Ändern des Verlaufsbereichs; die Verwendung von Ausschneiden, Kopieren und Einfügen und mehr.

Hinweis: In diesem Modul wird der Begriff “Hauptbildschirm des Rechners ” verwendet. Andere Module verwenden den Begriff “Hauptbildschirm”. Beide Begriffe sind austauschbar und beschreiben den selben Bildschirm.

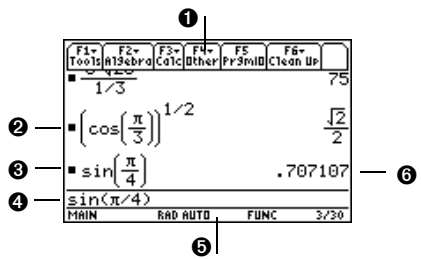
Anzeigen des Startbildschirms des Rechners

Wenn Sie Ihren TI-89 Titanium oder Voyage™ 200 Graphischer Rechner zum ersten Mal einschalten, wird die Apps-Arbeitsfläche angezeigt. Um den Startbildschirm

anzuzeigen, markieren Sie das Startbildschirm-Symbol, und drücken Sie [ENTER]. Sie können den Startbildschirm auch durch Drücken von [HOME] (TI-89 Titanium) oder [♦] [CALC HOME] (Voyage™ 200) anzeigen. Wenn Sie den Apps-Arbeitsflächen-Modus deaktivieren, wird der Startbildschirm automatisch angezeigt.

Bestandteile des Startbildschirms des Rechners

Das folgende Beispiel enthält bereits eingegebene Daten und dient zur Beschreibung der Hauptbestandteile des Startbildschirms. Eingabe-/Antwort-Paare werden im History-Bereich im “Pretty-Print-Modus” angezeigt. “Pretty-Print” zeigt Terme in der Form an, wie sie an die Tafel geschrieben oder in Lehrbüchern dargestellt werden.



❶ Menüleiste
Zeigt Menüs mit Operationen für den Startbildschirm des Rechners an. Zum Anzeigen eines Menüleisten-Menüs drücken Sie [F1], [F2] etc.

❷ Pretty Print-Anzeige
Zeigt Exponenten, Wurzeln, Brüche etc. in der traditionellen Form.

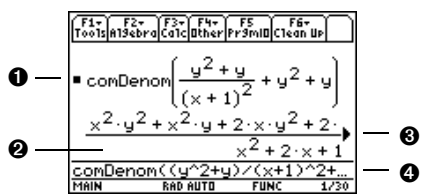
❸ Letzter Eintrag
Ihre letzte Eingabe.

❹ Eingabezeile
Hier geben Sie Ausdrücke oder Anweisungen ein.

❺ Satuszeile
Zeigt den aktuellen Status des Rechengeräts inklusive verschiedener wichtiger Moduseinstellungen.

❻ Letzte Antwort
Ergebnis Ihrer letzten Eingabe. Beachten Sie, dass Ergebnisse nicht in der Eingabezeile angezeigt werden. Hinweis: In diesem Beispiel wurde ♦ ENTER (Approx) verwendet.

Das folgende Beispiel zeigt eine Antwort, die sich nicht in derselben Zeile wie der Term befindet. Beachten Sie, dass die Antwortlänge die Bildschirmbreite übersteigt. Ein Pfeil (▶) zeigt an, dass die Antwort fortgesetzt wird. Die Eingabezeile enthält eine Auslassung (...). Eine Auslassung weist darauf hin, dass die Eingabelänge die Bildschirmbreite übersteigt.



❶ Letzter Eintrag
Der “Pretty-Print-Modus” ist eingeschaltet. Exponenten, Wurzeln, Brüche etc. werden in traditioneller Schreibweise angezeigt.

❷ Protokoll-Bereich
Auflistung eingegebener Eingabe/Antwort-Paare. Mit neuen Eingaben rollen die bestehenden Paare aufwärts im Bildschirm.

❸ Fortsetzung der Antwort
Markieren Sie die Antwort, und drücken Sie ⊙, um durch die Antwort nach rechts zu scrollen und deren Rest anzuzeigen. Beachten Sie, dass sich die Antwort nicht in derselben Zeile wie der Term befindet.

❹ Ausdruck wird fortgeführt (…)
Drücken Sie ⊙, um die Eingabe nach rechts zu scrollen und deren Rest anzuzeigen. Drücken Sie 2nd ◖ oder 2nd ⊙,um zum Anfang oder Ende der Eingabezeile zu gelangen.

Protokoll-Bereich

Der Protokoll-Bereich zeigt bis zu acht frühere Eingabe-/Antwort-Paare an (je nach Komplexität und Höhe der angezeigten Terme). Bei vollem Display scrollen die Daten über den oberen Seitenrand heraus. Mit dem Protokoll-Bereich können Sie:

- Vorherige Einträge und Antworten erneut anschauen. Mit dem Cursor können Sie die aus der Seite gescrollten Einträge und Antworten wieder anzeigen.
- Vorherige Einträge oder Antworten in der Eingabezeile wieder aufrufen oder in sie einfügen, so dass Sie diese wiederverwenden oder bearbeiten können.

Durch den Protokoll-Bereich scrollen

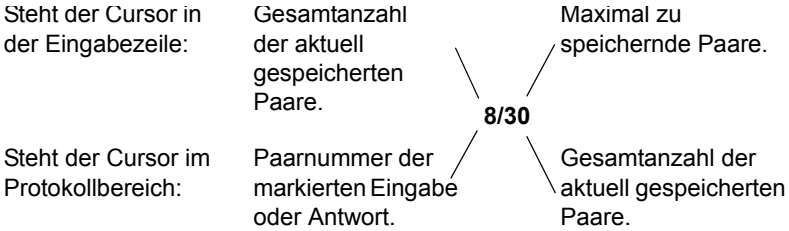
Der Cursor befindet sich im Normalfall in der Eingabezeile. Sie können ihn aber in den Protokoll-Bereich führen.

Wirkung:	Vorgehensweise:
Eingaben oder Antworten anzeigen, die aus dem Bild gescrollt sind	• Von der Eingabezeile aus drücken Sie ⊖, um die letzte Antwort zu markieren. • Verwenden Sie weiterhin ⊖, um den Cursor innerhalb des Protokoll-Bereichs von Antwort zu Eingabe aufwärts zu bewegen.
Zum ältesten oder neuesten Protokoll-Paar springen	Befindet sich der Cursor im Protokoll-Bereich, drücken Sie ♦ ⊖ bzw. ♦ ⊕.
Eingaben oder Antworten ansehen, die länger als eine Zeile sind (▶ am Ende der Zeile)	Setzen Sie den Cursor auf die Eingabe oder die Antwort. Scrollen Sie mit ▷ und ◁ nach rechts bzw. links (oder 2nd ▷ und 2nd ◁, um zum Ende oder zum Anfang zu gelangen).
Den Cursor in die Eingabezeile zurückbrigen	Drücken Sie ESC, oder ⊕, bis sich der Cursor wieder in der Eingabezeile befindet.

Hinweis: Ein Beispiel für die Ansicht einer langen Antwort wird detailliert beschrieben.

Protokollinformation in der Statuszeile

Informationen über die Eingabe-/Antwort-Paare können Sie der Protokollanzeige in der Statuszeile entnehmen. Beispiel:



Das Gerät ist so eingestellt, dass die letzten 30 Eingabe-/Antwort-Paare gespeichert werden. Ist der Protokoll-Bereich voll (angezeigt durch 30/30), wird das nächste Eingabe-/Antwort-Paar gespeichert und das älteste gelöscht. Die Protokollanzeige verändert sich nicht.

Den Protokoll-Bereich ändern

Wirkung:	Vorgehensweise:
Die Anzahl speicherbarer Paare ändern	Drücken Sie F1, und wählen Sie **9:Format**, oder drücken Sie: ◆ | ◆ **F** Drücken Sie dann ▶, und markieren Sie die neue Anzahl durch ▲ oder ▼. Drücken Sie zweimal ENTER.
Den Protokoll-Bereich löschen und alle gespeicherten Paare entfernen	Drücken Sie F1, und wählen Sie **8:Clear Home**, oder geben Sie **ClrHome** in die Eingabezeile ein.

Wirkung:	Vorgehensweise:
Ein bestimmtes Eingabe-/Antwort-Paar löschen	Setzen Sie den Cursor auf die Eingabe oder die Antwort. Drücken Sie ← oder CLEAR.

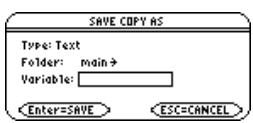
Eingaben im Startbildschirm des Rechners als Text-Editor-Skript speichern

Wenn Sie alle Eingabe des Protokoll-Bereichs speichern möchten, können Sie den Startbildschirm des Rechners in einer Textvariablen speichern. Sollen diese Eingaben dann wieder ausgeführt werden, öffnen Sie die Variable mit dem Text-Editor als Befehls-Skript.

Die Einträge des Protokoll-Bereichs speichern

Im Startbildschirm des Rechners:

1. Drücken Sie F1, und wählen Sie **2:Save Copy As**.



2. Geben Sie ein Verzeichnis und eine Textvariable zum Speichern der Einträge an.

   Hinweis: Es werden nur Eingaben gespeichert, keine Antworten.

Menüpunkt	Beschreibung
Type	Steht automatisch auf Text und kann nicht geändert werden.
Folder	Zeigt das Verzeichnis an, in welchem die Textvariable gespeichert wird. Möchten Sie ein anderes Verzeichnis verwenden, drücken Sie ▶, und ein Menü mit den vorhandenen Verzeichnissen wird geöffnet. Wählen Sie dann ein Verzeichnis.
Variable	Geben Sie einen gültigen, noch nicht verwendeten Variablennamen ein.

Hinweis: Näheres zu Verzeichnissen finden Sie im Modul *Speicher- und Variablen-Verwaltung*.

3. Drücken Sie ENTER (nach dem Ausfüllen eines Eingabefelds drücken Sie ENTER zweimal).

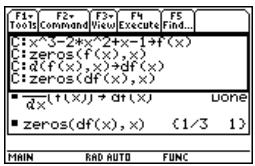
Gespeicherte Eingaben wiederherstellen

Da die Eingaben in einem Skriptformat gespeichert sind, können Sie sie im Startbildschirm des Rechners nicht direkt wiederherstellen. (Im Menü F1 des Startbildschirm des Rechners ist **1:Open** nicht verfügbar.) Statt dessen:

1. Öffnen Sie die Variable, in welcher die Hauptbildschirm-Eingaben gespeichert sind, mit dem Text-Editor.

   Die gespeicherten Eingaben werden als Befehlszeilen angezeigt, die Sie in beliebiger Reihenfolge einzeln ausführen können.

2. Setzen Sie den Cursor auf die erste Zeile des Skripts, und drücken Sie mehrmals F4, um die Befehle Zeile für Zeile auszuführen.
3. Zeigen Sie den wiederhergestellten Startbildschirm des Rechners an.



   Dieser geteilte Bildschirm zeigt den Text-Editor (mit Befehlszeilen-Skript) und den wiederhergestellten Startbildschirm des Rechners.

Hinweis: Umfassende Anweisungen zur Verwendung des Text-Editors und zum Ausführen eines Befehls-Skripts finden Sie im Modul *Texteditor*.

Informationen ausschneiden, kopieren und einfügen

Mit Hilfe der Funktionen Ausschneiden, Kopieren und Einfügen können Sie Informationen innerhalb derselben Anwendung oder auch von einer Anwendung zu einer anderen verschieben oder kopieren. Dafür wird die Zwischenablage des TI-89 Titanium / Voyage™ 200 Graphischer Rechner verwendet, ein Speicherbereich, der als temporärer Speicher dient.

Automatisches Einfügen oder “Ausschneiden/Kopieren/Einfügen”?

Auto-paste stellt eine schnelle Möglichkeit zum Kopieren eines Eintrags oder Ergebnisses aus dem Protokollbereich und zum Einfügen in die Eingabezeile bereit.

1. Verwenden Sie ▲ und ▼, um den Eintrag im Protokoll-Bereich zu markieren.
2. Drücken Sie ENTER, um diesen Eintrag automatisch in die Eingabezeile einzufügen.

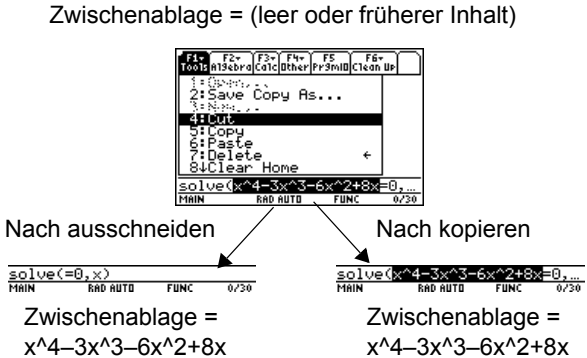
Zum Kopieren oder Verschieben von Informationen aus der Eingabezeile müssen Sie die Einzelfunktionen Ausschneiden, Kopieren und Einfügen verwenden. (Sie können im Protokoll-Bereich zwar kopieren, aber nicht ausschneiden oder einfügen.)

Informationen in die Zwischenablage kopieren oder ausschneiden

Ausgeschnittene oder kopierte Informationen werden in der Zwischenablage gespeichert. Beim Ausschneiden wird die Information von ihrem aktuellen Ort entfernt (dient zum Verschieben von Informationen), und beim Kopieren bleibt sie dort erhalten.

1. Markieren Sie die auszuschneidenden oder zu kopierenden Zeichen.

   Setzen Sie den Cursor in der Eingabezeile vor oder hinter die gewünschten Zeichen. Halten Sie ↑ gedrückt, und drücken Sie gleichzeitig ◀ oder ▶, um die Zeichen links bzw. rechts vom Cursor zu markieren.
2. Drücken Sie F1, und wählen Sie **4:Cut** oder **5:Copy**.



Hinweis: Ausschneiden, Kopieren und Einfügen sind auch ohne das über F1 aufzurufende Menü möglich, und zwar wie folgt:

- ♦ [CUT], ♦ [COPY], oder ♦ [PASTE]
- ♦ **X**, ♦ **C**, oder ♦ **V**

Ausschneiden ist nicht gleichbedeutend mit Löschen. Gelöschte Informationen werden nicht in der Zwischenablage gespeichert und können auch nicht wieder abgerufen werden.

Hinweis: Der vorhandene Inhalt der Zwischenablage wird durch die neu ausgeschnittenen oder kopierten Informationen ersetzt.

Informationen aus der Zwischenablage einfügen

Beim Einfügen wird der Inhalt der Zwischenablage an der aktuellen Cursorposition in die Eingabezeile eingesetzt. Der Inhalt der Zwischenablage bleibt unverändert.

1. Setzen Sie den Cursor an die Stelle, an welcher die Information eingefügt werden soll.
2. Drücken Sie F1, und wählen Sie **6:Paste**, oder verwenden Sie die Tastenkombination:
   - [PASTE]
   - **V**

Beispiel: Kopieren und Einfügen

Wenn Sie einen Term wiederverwenden möchten, ohne ihn jedesmal neu eingeben zu müssen:

1. Kopieren Sie die gewünschte Information.

   a) Verwenden Sie ⇧ ▶ oder ⇧ ◀, um den Term zu markieren.

   b) Drücken Sie:
   - [COPY]
   - **C**

   c) Drücken Sie hier ENTER, um die Eingabe auszuwerten.

2. Fügen Sie die kopierte Information in einen neuen Eintrag ein.

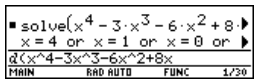
   a) Beginnen Sie mit einem neuen Eintrag und platzieren Sie den Cursor an die Stelle, an der Sie die kopierten Informationen einfügen wollen.

b) Drücken Sie F3 **1**, um die Funktion *d* differentiate zu wählen.

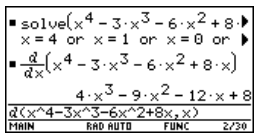
c) Fügen Sie den kopierten Term ein, indem Sie:

◆ [PASTE]

◆ **V**



d) Vervollständigen Sie die neue Eingabe, und drücken Sie ENTER.



Hinweis: Sie können einen Term auch wiederverwenden, indem Sie eine benutzerspezifische Funktion definieren.

3. Fügen Sie die kopierte Information in eine andere Anwendung ein.

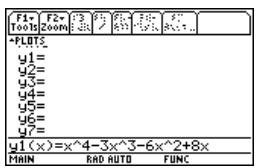
a) Drücken Sie ◆ [Y=], um den Y= Editor zu öffnen.

b) Drücken Sie ENTER, um **y1(x)** zu definieren.

c) Drücken Sie zum Einfügen:

◆ [PASTE]

◆ **V**



d) Drücken Sie ENTER, um die neue Definition zu speichern.

Hinweis: Durch Kopieren und Einfügen können Sie Informationen problemlos von einer Anwendung in eine andere übertragen.

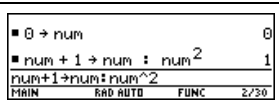
Eine frühere Eingabe oder die letzte Antwort wiederverwenden

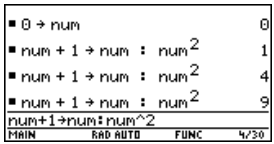
Sie können eine frühere Eingabe in unverändertem Zustand oder nach Bearbeitung wiederverwenden. Sie können auch die zuletzt errechnete Antwort in einen neuen Term einsetzen und wiederverwenden.

Den Term in der Eingabezeile wiederverwenden

Drücken Sie [ENTER], um einen Term auszuwerten, läßt der TI-89 Titanium / Voyage™ 200 Graphischer Rechner diesen in der Eingabezeile stehen und markiert ihn. Sie können die Eingabe nach Bedarf überschreiben oder wiederverwenden.

Ermitteln Sie zum Beispiel anhand einer Variablen das Quadrat von 1, 2, 3, etc. Setzen Sie, wie unten gezeigt, den Anfangswert fest und weisen ihn einer Variablen zu. Geben Sie dann den Zuwachs der Variablen ein und berechnen das Quadrat.

		Anzeige
0 [STO▸] [2nd] [a-lock] NUM [ENTER]	0 [STO▸] NUM [ENTER]	
NUM [alpha] [+] 1 [STO▸] [2nd] [a-lock] NUM [2nd] [:] NUM [^] 2 [ENTER]	NUM [+] 1 [STO▸] NUM [2nd] [:] NUM [^] 2 [ENTER]	

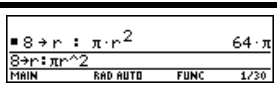
		Anzeige
ENTER ENTER	ENTER ENTER	

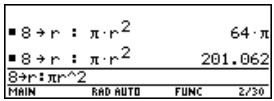
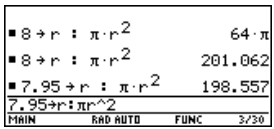
Hinweis: Das Wiederverwenden einer unveränderten Eingabe ist bei iterativen Berechnungen mit Variablen nützlich.

Arbeiten Sie mit der Gleichung $\mathbf{A=\pi r^2}$ und mit Versuch und Irrtum, um den Radius eines Kreises mit 200 Quadratzentimetern Fläche zu bestimmen.

Hinweis: Für kleine Änderungen können Sie die Eingabe bearbeiten, ohne sie ganz neuschreiben zu müssen.

Im folgenden Beispiel wird als geschätzter Wert 8 angegeben; danach wird das Ergebnis als Näherungswert im Fließpunktformat angegeben. Sie können den Wert bearbeiten, die Berechnung mit `7.95` erneut ausführen und den Vorgang fortsetzen, bis die Ergebnisgenauigkeit Ihren Vorstellungen entspricht

		Anzeige
8 STO▸ alpha R 2nd [:] 2nd [π] alpha R ^ 2 ENTER	8 STO▸ R 2nd [:] 2nd [π] R ^ 2 ENTER	

		Anzeige
♦ ENTER	♦ ENTER	
◀ ♦ DEL 7.95 ENTER	◀ ♦ DEL 7.95 ENTER	

Hinweis: Enthält die Eingabe einen Dezimalpunkt, wird das Ergebnis automatisch in Gleitkommaform angezeigt.

Eine frühere Eingabe abrufen

Sie können jede zuvor im Protokoll-Bereich gespeicherte Eingabe abrufen, selbst dann, wenn die Eingabe über den oberen Bildschirmrand hinausgescrollt ist. Die abgerufene Eingabe ersetzt den aktuell angezeigten Inhalt der Eingabezeile. Sie können die abgerufene Eingabe dann wieder ausführen oder auch bearbeiten.

Abzurufende Eingabe:	Drücken Sie:	Wirkung:
Letzte Eingabe (wenn Sie die Eingabezeile geändert haben)	2nd ENTRY einmal	Wurde die letzte Eingabe noch auf der Eingabezeile angezeigt, wird nun die davor letzte Eingabe abgerufen.

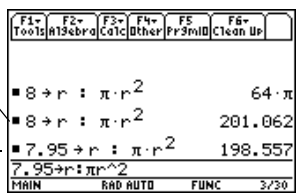
Abzurufende Eingabe:	Drücken Sie:	Wirkung:
Frühere Eingaben	2nd [ENTRY] wiederholt	Mit jeder Betätigung wird die jeweils letzte Eingabe vor der in der Eingabezeile angezeigten abgerufen.

Hinweis: Sie können zum Abrufen einer beliebigen früheren Eingabe auch die entry-Funktion verwenden. Siehe **entry( )** im Modul *Technische Referenz*.

Beispiel:

Steht die letzte Eingabe in der Eingabezeile, wird durch 2nd [ENTRY] diese Eingabe abgerufen.

Wurde die Eingabezeile bearbeitet oder gelöscht, wird durch 2nd [ENTRY] diese Eingabe abgerufen.



Die letzte Antwort abrufen

Jdesmal, wenn ein Term ausgewertet wird, speichert der TI-89 Titanium / Voyage™ 200 die Antwort in der Variablen **ans(1)**. Um diese Variable in die Eingabezeile einzusetzen, drücken Sie 2nd [ANS].

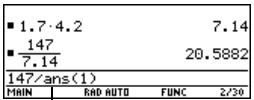
Berechnen Sie beispielsweise den Flächeninhalt eines Gartenstücks mit den Maßen 1,7 m auf 4,2 m. Berechnen Sie dann den Ertrag pro Quadratmeter, wenn das Grundstück eine Gesamternte von 147 Tomaten abwirft.

1. Ermitteln Sie den Flächeninhalt.

   1.7 × 4.2 ENTER

2. Ermitteln Sie den Ertrag.

   147 ÷ 2nd [ANS] ENTER



Die Variable **ans(1)** wird eingesetzt und ihr Wert in der Berechnung verwendet.

ans(1) enthält stets die letzte Antwort, und dementsprechend enthalten **ans(2)**, **ans(3)**, etc. frühere Antworten. So enthält **ans(2)** zum Beispiel die vorletzte Antwort.

Hinweis: Siehe **ans( )** in das modul *Technische Referenz*.

Eine Eingabe oder eine Antwort aus dem Protokoll-Bereich automatisch übernehmen

Sie können eine beliebige Eingabe oder Antwort aus dem Protokoll-Bereich wählen und eine Kopie davon in die Eingabezeile einfügen. Dadurch können Sie eine frühere Eingabe oder Antwort in einen neuen Term einsetzen, ohne dass Sie diese erneut schreiben müssen.

Wozu dient das automatische Einfügen

Die Wirkung des automatischen Einfügens ist ähnlich wie 2nd [ENTRY] und 2nd [ANS], die im vorigen Abschnitt beschrieben wurden. Es gibt aber Unterschiede.

Bei Eingaben:

Dient das Einfügen zum:	**Dient 2nd [ENTRY] zum:**
Einsetzen einer beliebigen früheren Eingabe in die Eingabezeile.	Ersetzen des Inhalts der Eingabezeile durch eine beliebige frühere Eingabe.

Bei Antworten:

Dient das Einfügen zum:	**Dient 2nd [ANS] zum:**
Einsetzen des angezeigten Werts einer beliebigen früheren Antwort in die Eingabezeile.	Einsetzen der Variablen **ans(1)**, die nur die letzte Antwort enthält. Jedesmal, wenn Sie eine Rechnung eingeben, wird **ans(1)** auf den Stand der letzten Antwort gebracht.

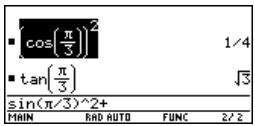
Hinweis: Sie können Daten auch mit Hilfe des Menüleisten-Menüs F1 einfügen.

Eine Eingabe oder Antwort automatisch einfügen

1. Setzen Sie den Cursor in der Eingabezeile an die Stelle, an welcher die Eingabe oder Antwort eingefügt werden soll.
2. Drücken Sie ⌃, um den Cursor in den Protokoll-Bereich zu bewegen. Die letzte Antwort wird markiert.

3. Markieren Sie die Eingabe oder Antwort, die Sie einfügen möchten, mit ▲ und ▼.

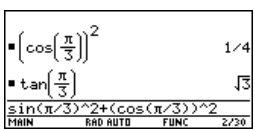
   - ▲ durchläuft den Protokoll-Bereich von Antwort zu Eingabe aufwärts.
   - Verwenden Sie ▲ zum Markieren von Eingaben, die über den Bildschirmrand herausgescrollt sind.

   Hinweis: Drücken Sie [ESC], um das automatische Einfügen abzubrechen und in die Eingabezeile zurückzukehren. Eine Eingabe oder Antwort, deren Länge eine Zeile übersteigt (▶ am Zeilenende), können Sie durch ▶ und ◀ oder [2nd] ▶und [2nd] ◀ansehen.

4. Drücken Sie [ENTER].

   Die markierte Eingabe wird in die Eingabezeile übernommen.

Dadurch wird die gesamte Eingabe oder Antwort eingefügt. Benötigen Sie nur einen Teil der Eingabe oder Antwort, bearbeiten Sie die Eingabezeile, um die überflüssigen Teile zu entfernen.

Benutzerdefinierte Funktionen erstellen und auswerten

Wenn Sie denselben Term (mit unterschiedlichen Werten) mehrmals benötigen, können Sie durch benutzerdefinierte Funktionen sehr viel Zeit sparen. Benutzerdefinierte

Funktionen können auch die Fähigkeiten Ihres TI-89 Titanium / Voyage™ 200 Graphischer Rechner über die integrierten Funktionen hinaus erweitern.

Format einer Funktion

Die nachfolgenden Beispiele zeigen benutzerdefinierte Funktionen mit einem und mit zwei Parametern. Sie können so viele Parameter wie nötig verwenden. In diesen Beispielen besteht die Definition aus einem einzelnen Term (oder einer Anweisung).

❶ Funktionsname
❷ Parameterliste
❸ Term

Verwenden Sie beim Definieren von Funktionen und Programmen eindeutige Parameternamen, die nicht die gleichen wie beim späteren Funktions- oder Programmaufruf sind.

Hinweis: Für Funktionsnamen gelten dieselben Regeln wie für Variablennamen. Siehe “Variablenwerte speichern und abrufen” in *Bedienung des Taschenrechners*.

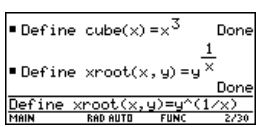
Verwenden Sie in der Parameterliste unbedingt dieselben Parameter wie in der Definition. Beispiel: **cube(n) = x^3** führt beim Auswerten der Funktion zu unerwarteten Ergebnissen.

Parameter (in diesen Beispielen x und y) sind Platzhalter, die alle beliebigen Werte darstellen, welche Sie an die Funktion übertragen. Sie stellen die Variablen x und y erst

dann dar, wenn Sie beim Auswerten der Funktion x und y ausdrücklich als Parameter übergeben.

Eine benutzerdefinierte Funktion erstellen

Gehen Sie nach einer der folgenden Methoden vor.

Methode	Beschreibung
STO▸	Speichern Sie einen Term in einem Funktionsnamen (einschließlich der Parameterliste).
Define-Befehl	Definieren Sie einen Funktionsnamen (einschließlich der Parameterliste) als Term. 
Programm-Editor	In *Programmierung* finden Sie Anleitungen zum Erstellen einer benutzerdefinierten Funktion.

Eine Funktion mit mehreren Anweisungen erstellen

Sie können auch benutzerdefinierte Funktionen erstellen, deren Definition aus mehreren Anweisungen besteht. In der Definition dürfen zahlreiche Steuer- und

Entscheidungsstrukturen enthalten sein (**If**, **ElseIf**, **Return**, etc.), die zum Programmieren verwendet werden.

Hinweis: Näheres zu Gemeinsamkeiten und Unterschieden von Funktionen und Programmen finden Sie in *Programmierung*.

Angenommen, Sie möchten eine Funktion erzeugen, die mehrere Kehrwerte ganzer Zahlen (**n**) addieren soll.

$$\frac{1}{n} + \frac{1}{n-1} + \ldots + \frac{1}{1}$$

Vor dem Erstellen einer Funktionsdefinition, die mehrere Anweisungen enthält, kann es hilfreich sein, sich die Definition zuerst in Blockform zu vergegenwärtigen.

```
❶ Func
❷ Local temp,i
  If fPart(nn)≠0 or nn≤0
❸   Return "bad argument"
  0→temp
❹ For i,nn,1,-1
    approx(temp+1/i)→temp
  EndFor
❺ Return temp
❶ EndFunc
```

❶ **Func** und **EndFunc** müssen am Anfang/ Ende der Funktion stehen.
❷ Variablen außerhalb der Parameterliste müssen als "lokal" deklariert werden.
❸ Gibt eine Meldung zurück, wenn nn keine ganze Zahl oder nn≤0 ist.
❹ Addiert die Kehrwerte.
❺ Gibt die Summe zurück.

Im Startbildschirm des Rechners müssen Sie eine aus mehreren Anweisungen bestehende Funktion in eine einzige Zeile eingeben. Sie verwenden dazu wie bei Funktionen, die nur eine Anweisung enthalten, den Befehl **Define**.

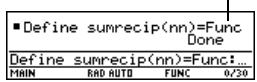
Trennen sie die einzelnen Anweisungen durch Doppelpunkte.

```
Define sumrecip(nn)=Func:Local temp,i: ... :EndFunc
```

Verwenden Sie Parameternamen, die beim Aufrufen der Funktion/des Programms nicht verwendet werden.

Im Startbildschirm des Rechners:

Funktionen mit mehreren Anweisungen: angezeigt als **Func**.

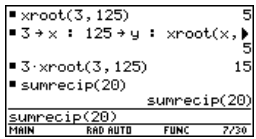


Eingabe einer Funktion mit mehreren Anweisungen in einer Zeile. Denken Sie an die Doppelpunkte.

Hinweis: Eine komplexe Funktion mit mehreren Anweisungen können Sie einfacher mit dem Programm-Editor erstellen als im Startbildschirm des Rechners. Siehe hierzu *Programmierung*.

Eine Funktion auswerten

Benutzerdefinierte Funktionen verwenden Sie wie jede andere Funktion. Sie können sie einzeln auswerten oder in einen anderen Term aufnehmen.



Eine Funktionsdefinition anzeigen und ändern

Sie möchten:	Vorgehensweise:
Eine Liste aller benutzerdefinierten Funktionen anzeigen	Drücken Sie 2nd [VAR-LINK], um den VAR-LINK-Bildschirm zu öffnen. Zum Angeben des Variablentyps **Function** müssen Sie unter Umständen die Menüleiste F2 **View** benutzen. (Siehe *Speicherund Variablen-Verwaltung*.) – oder – Drücken Sie: CATALOG F4 2nd [CATALOG] F4
Anzeige einer Liste von Flash-Anwendungsfunktionen	Drücken Sie: CATALOG F3 2nd [CATALOG] F3

Sie möchten:	Vorgehensweise:
Die Definition einer benutzerdefinierten Funktion anzeigen	Markieren Sie die Funktion im VAR-LINK Bildschirm, und rufen Sie das Menü Contents auf. [2nd] [F6] [F6] – oder – Drücken Sie im Startbildschirm des Rechners [2nd] [RCL]. Geben Sie den Funktionsnamen, nicht aber die Parameterliste (z.B. **xroot**) ein, und drücken Sie [ENTER] zweimal. – oder – Öffnen Sie die Funktion im Programm-Editor (siehe *Programmierung*).
Die Definition ändern	Drücken Sie im Startbildschirm des Rechners [2nd] [RCL], um die Definition anzuzeigen. Ändern Sie die Definition. Speichern Sie die neue Definition dann mit [STO▸] oder **Define**. – oder – Öffnen Sie die Funktion im Programm-Editor, bearbeiten Sie sie, und speichern Sie die Änderungen (siehe *Programmierung*).

Hinweis: Sie können eine benutzerdefinierte Funktion im Dialogfeld CATALOG anzeigen lassen, jedoch ihre Definition nicht im CATALOG einsehen oder bearbeiten.

Eine Eingabe oder Antwort ist “zu groß”

Manchmal kann eine Eingabe oder eine Antwort zu lang oder zu hoch sein, um vollständig im Protokoll-Bereich angezeigt werden zu können. Es kann auch sein, der

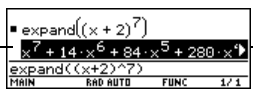
TI-89 Titanium / Voyage™ 200 Graphischer Rechner eine Antwort aus Speicherplatzmangel nicht anzeigen kann.

Eine Eingabe oder Antwort ist “zu lang”

Setzen Sie den Cursor auf den Protokoll-Bereich, und markieren Sie die Eingabe oder die Antwort. Scrollen Sie dann mit Hilfe des Cursorfelds. Beispiel:

- Hier wird eine überlange Antwort gezeigt.

Drücken Sie zum Scrollen nach links ◀ oder 2nd ◀.



Drücken Sie zum Scrollen nach rechts ▶ oder 2nd ▶.

- Hier wird eine Antwort angezeigt, die sowohl zu lang als auch zu hoch ist.

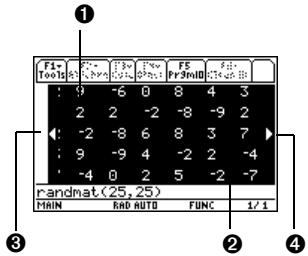
  Hinweis: In diesem Beispiel wird eine 25 x 25-Matrix mit der Funktion **randMat** erstellt.

❶ Drücken Sie zum Abwärtsscrollen ▼ oder [↑] ▼

Drücken Sie zum Abwärtsscrollen ▼ oder [2nd] ▼

❷ Drücken Sie zum Aufwärtsscrollen [↑] ▲

Drücken Sie zum Aufwärtsscrollen [2nd] ▲



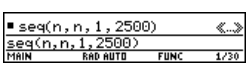
❸ Drücken Sie zum Scrollen nach links ◀ oder [2nd] ◀

❹ Drücken Sie zum Scrollen nach rechts ▶ oder [2nd] ▶

Bei Speicherplatzmangel

Verfügt der TI-89 Titanium / Voyage™ 200 Graphischer Rechner nicht mehr über genügend Speicherplatz zum Anzeigen einer Antwort, wird das Symbol <<...>> ausgegeben.

Beispiel:



Hinweis: In diesem Beispiel wird für die Erstellung einer Liste mit der Folge der ganzen Zahlen von 1 bis 2500 die Funktion seq verwendet.

Erscheint das Symbol <<...>> , kann die Antwort nicht angezeigt werden, auch nicht durch Markieren und Scrollen.

Zur Abhilfe können Sie:

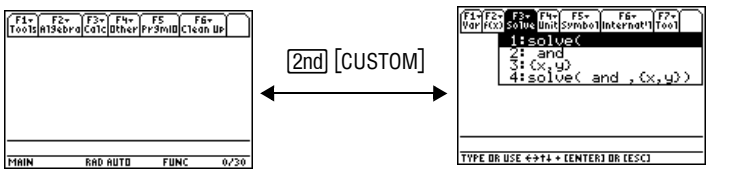
- Speicherkapazität freigeben: Löschen Sie hierzu nicht benötigte Variablen und/oder Flash-Anwendungen. Verwenden Sie dazu 2nd [VAR-LINK] (siehe *Speicherund Variablen-Verwaltung*).
- Die Aufgabe nach Möglichkeit in kleinere Teile splitten, die weniger Speicherplatz für Berechnung und Anzeige erfordern.

Das Benutzermenü verwenden

Der TI-89 Titanium / Voyage™ 200 Graphischer Rechner verfügt über ein benutzerspezifisches Menü, das Sie jederzeit aus- und einschalten können. Sie können entweder das vorgegebene Benutzermenü verwenden oder wie in *Programmierung* beschrieben Ihr eigenes erstellen.

Das Benutzermenü ein- und ausschalten

Wenn Sie das Benutzermenü einschalten, ersetzt es die normale Menüleiste, schalten Sie es aus, ist das normale Menü wieder aktiv. In folgendem Beispiel ist zunächst die normale Menüleiste des Startbildschirm des Rechners aktiv.



Normale Menüleiste, Startbildschirm des Rechners

Benutzerdefiniertes Menü

Hinweis: Sie können das Benutzermenü auch ein- oder ausschalten, indem Sie **CustmOn** bzw. **CustmOff** in die Eingabezeile tippen und [ENTER] drücken.

Sofern das Menü nicht geändert wurde, erscheint das vorgegebene Benutzermenü.

Menü	Funktion
[F1] **Var**	Gebräuchliche Variablennamen.
[F2] **f(x)**	Funktionsnamen wie z.B. f(x), g(x) und f(x,y).
[F3] **Solve**	Elemente zum Lösen von Gleichungen.
[F4] **Unit**	Gebräuchliche Einheiten wie z.B. _m und _l.
[F5] **Symbol**	Zeichen wie #, ? und ~.

Menü	Funktion
International 2nd F6 F6	Gebräuchliche akzentuierte Buchstaben wie è, é und ê.
Tool 2nd F7 F7	**ClrHome**, **NewProb**, und **CustmOff**.

Hinweis: Ein Benutzermenü kann Ihnen schnellen Zugriff auf häufig verwendete Elemente bieten. In *Programmierung* erfahren Sie, wie Sie Benutzermenüs für Ihre am häufigsten benötigten Elemente erstellen können.

Das vorgegebene Benutzermenü wiederherstellen

Gehen Sie wie folgt vor, wenn ein anderes als das vorgegebene Benutzermenü angezeigt wird und Sie das vorgegebene wiederherstellen möchten:

1. Drücken Sie im Startbildschirm des Rechners [2nd] [CATALOG] um das Benutzermenü auszuschalten und die normale Menüleiste des Statbildschirm des Rechners anzuzeigen.

2. Rufen Sie das Menü **Clean Up** auf, und wählen Sie **3:Restore custom default**.

   [2nd] [F6]

   [F6]

   Dadurch werden die zum Erstellen des Standardmenüs benötigten Befehle in die Eingabezeile kopiert.

   Hinweis: Das vorige Benutzermenü wird gelöscht. Wurde dieses Menü mit einem Programm erstellt (*Programmierung*), kann es später durch erneute Ausführung des Programms wieder erstellt werden.

3. Drücken Sie [ENTER], um die Befehle auszuführen und das vorgegebene Menü wiederherzustellen.

Ermitteln der Software-Version und der ID-Nummer

In manchen Situationen müssen Sie möglicherweise Angaben zu Ihrem TI-89 Titanium / Voyage™ 200 Graphischer Rechner und insbesondere die Software-Version und die ID-Nummer des Geräts herausfinden.

Anzeige des Info-Bildschirms “About”

1. Drücken Sie auf dem Startbildschirm des Rechners oder auf dem Apps-Desktop F1 und wählen Sie dann **A:About**.

   Der Inhalt Ihres Bildschirms weicht von nebenstehender Darstellung ab.

2. Drücken Sie ENTER oder ESC, um das Fenster zu schließen.

Wann brauchen Sie diese Angaben?

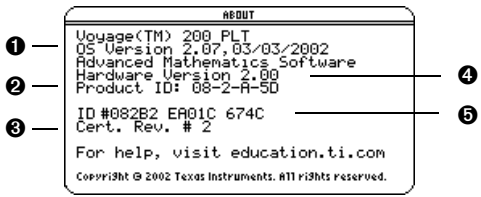
Die Angaben im About-Fenster sind für folgende Situationen vorgesehen:

- Wenn Sie neue oder Upgrade-Software für Ihren TI-89 Titanium / Voyage™ 200 erwerben, müssen Sie möglicherweise die aktuelle Software-Version und/oder die ID-Nummer Ihres Geräts angeben.
- Wenn Probleme mit Ihrem TI-89 Titanium / Voyage™ 200 auftreten und Sie sich mit dem technischen Dienst in Verbindung setzen müssen, kann die Kenntnis der Software-Version die Problemdiagnose erleichtern.

Der Bildschirm About zeigt die folgenden Informationen über Ihren Voyage™ 200 an:

- Hardwareversion
- BS (Advanced Mathematics Software)-Version
- Productidentifier (Produkt-ID)

- Geräte-ID
- Apps Zertifikatrevisionsnummer (engl. Cert. Rev.)



❶ BS version
❷ Produkt-ID
❸ Apps-Zertifikat Revisionsnummer
❹ Hardwareversion
❺ Geräte-ID (erfordert den Erwerb von Zertifikaten zur Installation gekaufter Apps)

Ihr Bildschirm wird anders aussehen, als der oben gezeigte Bildschirm.

Mit Symbolen arbeiten

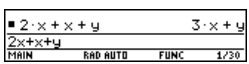
Definierte oder undefinierte Variablen verwenden

Beim Durchführen von Algebra- oder Analysisoperationen ist es wichtig, dass Sie sich mit dem Gebrauch von definierten und undefinierten Variablen und deren Auswirkungen vertraut machen. Sonst erhalten Sie unter Umständen anstelle des erwarteten algebraischen Terms eine Zahl als Ergebnis.

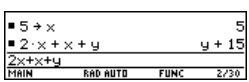
So werden undefinierte und definierte Variablen behandelt

Wenn Sie einen Term eingeben, der eine Variable enthält, handhabt der TI-89 Titanium / Voyage™ 200 Graphischer Rechner diese Variable auf eine von zwei Arten:

- Eine undefinierte Variable wird wie ein algebraisches Symbol behandelt.

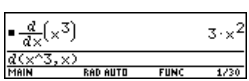


- Bei einer definierten Variablen (auch wenn sie mit 0 belegt ist) ersetzt ihr Wert die Variable.



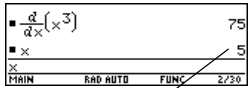
Um nachzuvollziehen, warum dies so wichtig ist, nehmen wir einmal an, Sie wollen die erste Ableitung von x^3 bezüglich x berechnen.

- Ist x undefiniert, dann hat das Ergebnis wahrscheinlich die Form, die Sie erwartet haben.



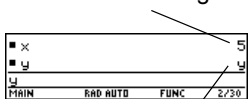
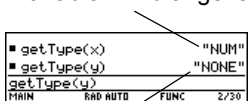
- Ist x dagegen definiert, dann hat das Ergebnis eventuell eine für Sie unerwartete Form.

  Hinweis: Beim Definieren einer Variablen sollten mehrere Zeichen im Namen verwendet werden. Lassen Sie Namen, die aus einem Zeichen bestehen, bei symbolischen Rechnungen undefiniert.



Wenn Sie nicht wissen, dass 5 zuvor in x gespeichert wurde, kann das Ergebnis 75 irreführend sein.

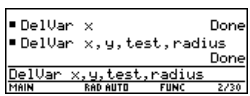
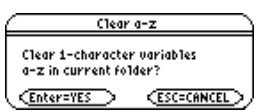
So ermitteln Sie, ob eine Variable undefiniert ist

Vorgehensweise:	**Beispiel:**
Geben Sie den Namen der Variablen ein.	Definiert: Der Wert der Variablen wird angezeigt.  Undefiniert: Der Name der Variablen wird angezeigt.
Verwenden Sie die **getType**-Funktion.	Definiert: Der Typ der Variablen wird angezeigt.  Undefiniert: “NONE” wird angezeigt.

Hinweis: Mit [2nd] [VAR-LINK] können Sie eine Liste der definierten Variablen gemäß der Beschreibung in *Speicher-und Variablen-Verwaltung* einsehen.

Eine definierte Variable löschen

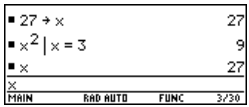
Sie können eine definierte Variable in eine undefinierte verwandeln, indem Sie diese löschen.

Löschen:	Vorgehensweise:
Eine oder mehrere definierte Variablen	Verwenden Sie die **DelVar**-Funktion.  Sie können Variablen auch gemäß der Beschreibung in *Speicher-und Variablen-Verwaltung* mit Hilfe des VAR-LINK-Bildschirms ([2nd] [VAR-LINK]) löschen.
Alle Variablen mit einem einzigen Buchstaben (a – z) im aktuellen Verzeichnis **Hinweis:** Näheres über Verzeichnisse finden Sie in *Weitere Funktionen des Hauptbildschirms*.	Wählen Sie im Menü Clean Up des Hauptbildschirms **1:Clear a-z**. Sie werden aufgefordert, den Löschvorgang durch [ENTER] zu bestätigen. 

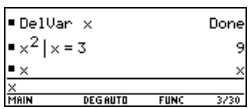
Eine Variable vorübergehend überschreiben

Geben Sie mit (|) den **"***with***"**-Operator ein, können Sie:

- Den definierten Wert einer Variablen vorübergehend überschreiben.



- Vorübergehend einen Wert für eine undefinierte Variable bestimmen.



Hinweis: Näheres über den | -Operator finden Sie unter Den "*with*"-Operator eingeben.

Geben Sie den Operator "*with*" (|) wie folgt ein:

[|]

2nd [|]

Exact-, Approximate- und Auto-Modus verwenden

Die Exact/Approx-Moduseinstellungen, die in *Bedienung des Taschenrechners* kurz dargestellt wurden, beeinflussen direkt die Präzision und Genauigkeit der Berechnung eines Ergebnisses durch den TI-89 Titanium / Voyage™ 200 Graphischer Rechner. Im vorliegenden Abschnitt wird der Zusammenhang zwischen diesen Moduseinstellungen und dem symbolischen Rechnen beschrieben.

EXACT-Einstellung

Bei der Einstellung Exact/Approx = EXACT arbeitet der Handheld mit exakter Arithmetik rationaler Zahlen und mit bis zu 614 Stellen im Zähler und 614 Stellen im Nenner. Die EXACT-Einstellung:

- Formt irrationale Zahlen so weit wie möglich ohne Näherung in Standardformen um. Zum Beispiel: $\sqrt{12}$ wird umgeformt in $2\sqrt{3}$ und **ln(1000)** in **3 ln(10)**.
- Konvertiert Gleitkommazahlen in rationale Zahlen. Zum Beispiel: 0.25 wird umgeformt in 1/4.

Bei den Funktionen **solve**, **cSolve**, **zeros**, **cZeros**, **factor**, ∫, **fMin** und **fMax** werden nur exakte symbolische Algorithmen verwendet. Diese Funktionen errechnen in der EXACT-Einstellung keine Näherungslösungen.

- Einige Gleichungen, wie beispielsweise $\mathbf{2^{-x} = x}$, haben Lösungen, die mit Funktionen und Operatoren des TI-89 Titanium / Voyage™ 200 nicht geschlossen dargestellt werden können.

- Bei derartigen Gleichungen werden in EXACT keine Näherungslösungen errechnet. Die Gleichung $\mathbf{2^{-x} = x}$ hat beispielsweise die Näherungslösung $\mathbf{x \approx 0.641186}$, die aber in der EXACT-Einstellung nicht angezeigt wird.

Vorteile	Nachteile
Die Ergebnisse sind exakt.	Bei Verwendung komplexerer rationaler Zahlen und irrationaler Konstanten können die Berechnungen: • So viel Speicherplatz beanspruchen, dass dieser vor Erhalt der Lösung schon erschöpft sein könnte. • Mehr Zeit in Anspruch nehmen. • Zu komplizierten Ergebnissen führen, die schwerer als ein Gleitkommaergebnis zu verstehen sind.

APPROXIMATE-Einstellung

Bei der Einstellung Exact/Approx = APPROXIMATE konvertiert der TI-89 Titanium / Voyage™ 200 rationale Zahlen und irrationale Konstanten in die Gleitkommaform. Beachten Sie jedoch die Ausnahmen:

- Bestimmte Standardfunktionen, bei welchen eines ihrer Argumente als ganze Zahl aufgefaßt wird, formen diese Zahl, wenn möglich, in eine ganze Zahl um. Zum Beispiel: d**(y(x), x, 2.0)** wird umgeformt in d**(y(x), x, 2)**.
- Ganzzahlige Gleitkommaexponenten werden in ganze Zahlen umgeformt. Zum Beispiel: $x^{2.0}$ wird selbst in der APPROXIMATE-Einstellung zu x^2.

Funktionen wie **solve** und ∫ (integrate) können sowohl mit exakten symbolischen als auch mit numerischen Techniken arbeiten. Diese Funktionen übergehen in der APPROXIMATE-Einstellung einige oder alle exakten symbolischen Methoden.

Vorteile	Nachteile
Werden keine exakten Ergebnisse benötigt, kann hier im Gegensatz zur EXACT-Einstellung Zeit und/oder Speicherplatz gespart werden. Näherungslösungen sind oft kompakter und verständlicher als exakte Ergebnisse.	Bei Ergebnissen mit undefinierten Variablen oder Funktionen zeigt sich häufig eine unvollständige Vereinfachung. Ein Koeffizient, der **0** sein sollte, könnte beispielsweise als kleiner Absolutwert, wie **1.23457E⁻11**, dargestellt werden.
Wenn Sie keine symbolischen Berechnungen durchführen möchten, hat die Verwendung von Näherungsergebnissen Ähnlichkeit mit dem Arbeiten auf vertrauten, herkömmlichen numerischen Rechnern.	Symbolische Operationen, wie z.B. Grenzwerte und Integration, führen in der APPROXIMATE-Einstellung zu wahrscheinlich weniger zufriedenstellenden Ergebnissen. Näherungsergebnisse sind oft weniger kompakt und verständlich als exakte Ergebnisse. Das Ergebnis **1/7**, zum Beispiel, ist Ihnen wahrscheinlich angenehmer als **.142857**.

AUTO-Einstellung

Bei der Einstellung Exact/Approx = AUTO arbeitet der TI-89 Titanium / Voyage™ 200 mit exakter rationaler Arithmetik, wenn alle Operanden rationale Zahlen sind. Anderenfalls

werden rationale Operanden in Gleitkommaform umgesetzt, und die Gleitkomma-Arithmetik wird verwendet. Das heißt, die Gleitkommaform “greift über”. Beispiel:

1/2 − **1/3** wird umgeformt in **1/6**
aber
0.5 − **1/3** wird zu **.16666666666667**

Dieses “Übergreifen” der Gleitkommaform überschreitet jedoch gewisse Grenzen, wie undefinierte Variablen, oder Elemente einer Liste oder einer Matrix nicht. Zum Beispiel:

(1/2 − **1/3) x + (0.5** − **1/3) y** wird umgeformt in **x/6 + .16666666666667 y**
und
{1/2 − **1/3, 0.5** − **1/3}** wird zu **{1/6, .16666666666667}**

Bei der AUTO-Einstellung bestimmen Funktionen, wie beispielsweise **solve**, so viele Lösungen wie möglich als exaktes Ergebnis und verwenden dann nötigenfalls zum Bestimmen zusätzlicher Lösungen numerische Näherungsverfahren. Dem entsprechend verwendet ∫ (integrate) numerische Nährungsverfahren, wenn exakte symbolische Methoden scheitern.

Vorteile	Nachteile
Sie sehen exakte Ergebnisse, wo dies machbar ist, und numerische Näherungsergebnisse, wenn sich exakte Ergebnisse als unpraktisch erweisen. Sie können das Format eines Ergebnisses häufig vorab steuern, indem Sie Koeffizienten entweder als rationale oder als Gleitkommazahlen eingeben.	Wenn Sie nur an exakten Ergebnissen interessiert sind, könnte die Suche nach Näherungsergebnissen eine Zeitverschwendung bedeuten. Wenn Sie nur an Näherungsergebnissen interessiert sind, könnte die Suche nach exakten Ergebnissen eine Zeitverschwendung bedeuten. Die Suche nach diesen exakten Ergebnissen könnte vor allem den Speicherplatz erschöpfen.

Automatische Vereinfachung

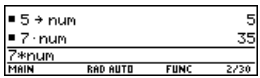
Wenn Sie in der Eingabezeile einen Term eingeben und ENTER drücken, wird dieser Term vom TI-89 Titanium / Voyage™ 200 Graphischer Rechner automatisch entsprechend vorgegebener Vereinfachungsregeln vereinfacht.

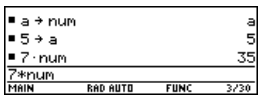
Vorgegebene Vereinfachungsregeln

Alle nachfolgenden Regeln werden automatisch angewendet. Zwischenergebnisse werden Ihnen nicht angezeigt.

- Hat eine Variable einen definierten Wert, so ersetzt dieser Wert die Variable.

  Ist die Variable in Form einer anderen Variablen definiert, dann wird sie durch ihren Wert auf der "niedrigsten Ebene" (genannt: infinite lookup) ersetzt.





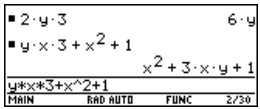
  Die voreingestellte Vereinfachung verändert Variablen, die Pfadnamen zur Angabe eines Verzeichnisse verwenden, nicht. So wird beispielsweise *x+class\x* nicht zu *2x* vereinfacht.

Hinweis: Näheres über Verzeichnis finden Sie in *Weitere Funktionen des Hauptbildschirms*.

- Bei Funktionen gilt:
  - Die Parameter werden vereinfacht. (Bei einigen integrierten Funktionen erfolgt die Vereinfachung einiger Parameter verzögert.)
  - Handelt es sich um eine bereits integrierte oder benutzerdefinierte Funktion, wird die Funktionsdefinition auf die vereinfachten Parameter angewendet. Die Funktionsform wird dann durch dieses Ergebnis ersetzt.

.

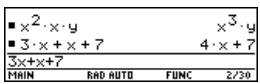
- Numerische Teilterme werden zusammengefaßt.
- Produkte und Summen werden sortiert.



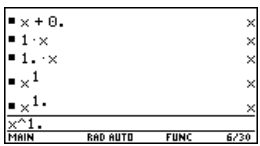
Produkte und Summen mit undefinierten Variablen werden nach dem Anfangsbuchstaben des Variablennamens sortiert.

- Undefinierte Variablen von *r* bis *z* werden als echte Variablen angenommen und in alphabetischer Reihenfolge am Anfang einer Summe aufgeführt.
- Undefinierte Variablen von *a* bis *q* werden als Repräsentanten von Konstanten angenommen und in alphabetischer Reihenfolge am Ende einer Summe (aber vor Zahlen) aufgeführt.

- Gleichnamige Faktoren und gleichnamige Terme werden zusammengefaßt.

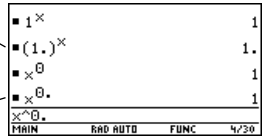


- Identitäten mit Nullen und Einsen werden ausgewertet.

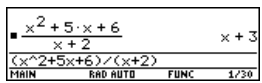


Durch diese Gleitkommazahl werden numerische Ergebnisse als Gleitkomma dargestellt.

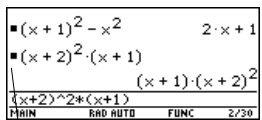
Wird eine ganze Gleitkommazahl als Exponent eingegeben, wird diese wie eine ganze Zahl behandelt (und führt nicht zu einem Gleitkommaergebnis).



- Gemeinesame Teiler in Bruchtermen werden gekürzt.

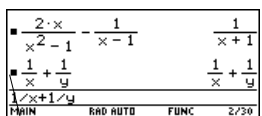


- Polynome werden entwickelt, bis keine weitere Vereinfachung stattfinden kann.



Keine Vereinfachung

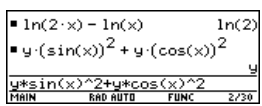
- Hauptnenner werden gebildet, bis keine weitere Vereinfachung stattfinden kann.



Keine Vereinfachung

- Identitäten werden ausgewertet. Zum Beispiel:

  ln(2x) = ln(2) + ln(x)
  und
  $\sin(x)^2 + \cos(x)^2 = 1$



Wie lange dauert der Vereinfachungsprozeß?

Je nach Komplexität einer Eingabe, eines Ergebnisses oder Zwischenterms kann es eine Weile dauern, bis ein Term entwickelt ist und die zum Vereinfachungsprozeß notwendigen gemeinsamen Teiler gekürzt wurden.

Möchten Sie einen Vereinfachungsvorgang, der zu lange dauert, unterbrechen, drücken Sie ON. Sie können dann versuchen, nur einen Teil des Terms zu vereinfachen. (Fügen Sie den gesamten Term mit Automatischem Einfügen in die Eingabezeile ein, und löschen Sie dann die unerwünschten Bestandteile.)

Verzögerte Vereinfachung bei gewissen eingebauten Funktionen

Für gewöhnlich werden Variablen automatisch auf ihre niedrigstmögliche Stufe vereinfacht, bevor sie an eine Funktion weitergereicht werden. Bei einigen Funktionen wird eine Gesamtvereinfachung jedoch solange verzögert, bis die Funktion durchgeführt wurde.

Funktionen, bei welchen die Vereinfachung verzögert wird

Funktionen, welche die verzögerte Vereinfachung verwenden, haben einen notwendigen var-Parameter, der die Funktion bezüglich dieser Variablen ausführt. Diese Funktionen verfügen über mindestens zwei Parameter mit der allgemeinen Form:

Function(*Term*, *var* [, …]**)**

Hinweis: Nicht alle Funktionen, die einen var-Parameter verwenden, benutzen die verzögerte Vereinfachung.

Zum Beispiel: **solve(x^2−x−2=0,x)**
d(x^2−x−2,x)
∫(x^2−x−2,x)
limit(x^2−x−2,x,5)

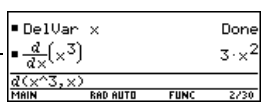
Bei Funktionen mit verzögerter Vereinfachung gilt:

1. Die *var*-Variable wird auf die niedrigste Stufe vereinfacht, auf der sie noch eine Variable bleibt (auch wenn sie noch weiter bis zu einer Zahl vereinfacht werden könnte).
2. Die Funktion wird mit der Variablen durchgeführt.
3. Kann *var* noch weiter vereinfacht werden, wird dieser Wert in das Ergebnis eingesetzt.

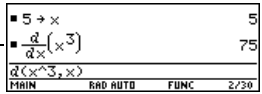
Hinweis: Je nach Situation kann für *var* ein numerischer Wert bestimmt werden oder nicht.

Beispiel:

x kann nicht vereinfacht werden.

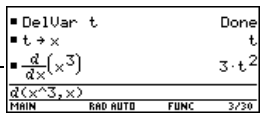


x wird nicht vereinfacht. Die Funktion verwendet x^3 und setzt dann 5 für x ein.

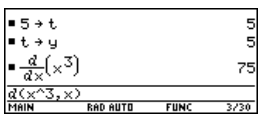


Hinweis: Im Beispiel rechts wird die Ableitung von x^3 in x=5 ermittelt. Bei einer etwaigen anfänglichen Vereinfachung von x^3 zu 75 würde nun die Ableitung von 75 vorliegen, was nicht das gewünschte Ergebnis darstellt.

x wird zu t vereinfacht. Die Funktion verwendet t^3.



x wird zu t vereinfacht. Die Funktion verwendet t^3 und setzt dann 5 für t ein.



Werte ersetzen und Beschränkungen festlegen

Mit dem "*with*"-Operator (|) können Werte temporär in einem Term ersetzt oder Bereichsbeschränkungen bestimmt werden.

Den “with”-Operator eingeben

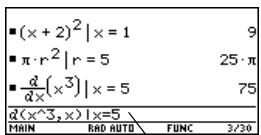
Geben Sie den Operator “*with*” (|) wie folgt ein:

[|]

2nd [|]

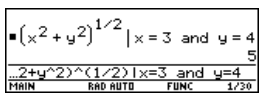
Ersetzen einer Variablen

Jedesmal, wenn eine spezifizierte Variable vorkommt, können Sie dafür einen Zahlenwert oder einen Term einsetzen.



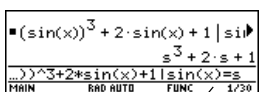
Erste Ableitung von x^3 für x = 5

Um mehrere Variablen gleichzeitig zu ersetzen, verwenden Sie den Boolesche *and*-Operator.



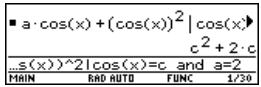
Ersetzen eines einfachen Terms

Jedesmal, wenn ein einfacher Term vorkommt, können Sie damit eine Variable, einen Zahlenwert oder einen anderen Term ersetzen.



Durch Einsetzen von s für sin(x) zeigt sich, daß der Term ein Polynom für sin(x) ist.

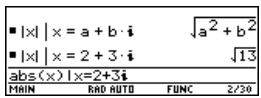
Durch Ersetzen eines häufig verwendeten (oder langen) Terms lassen sich Ergebnisse in kompakterer Form darstellen.



Hinweis: acos(x) ist nicht dasselbe wie **a∗cos(x).**

Ersetzen komplexer Werte

Komplexe Werte können genauso ersetzt werden wie andere Werte auch.



Alle undefinierten Variablen werden beim symbolischen Rechnen als reelle Zahlen gehandhabt. Zur Durchführung komplexer symbolischer Analysis müssen Sie eine komplexe Variable definieren. Zum Beispiel:

x+y*i*→z

Dann können Sie *z* als komplexe Variable verwenden. Sie können auch *z_* verwenden. Näheres hierzu erfahren Sie im Abschnitt über das Unterstreichungszeichen _ im Modul *Technische Referenz*.

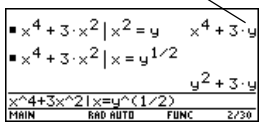
Hinweis:

- Einen Überblick über komplexe Zahlen finden Sie im Modul *Technische Referenz*.
- Das komplexe i erhalten Sie durch [2nd] [i]. Geben Sie nicht einfach den Buchstaben **i** auf der Tastatur ein.

Beachten Sie die Grenzen der Ersetzbarkeit

- Substitution tritt nur dann auf, wenn eine genaue Übereinstimmung für die Substitution besteht.

Nur x^2 wurde ersetzt, nicht x^4.



Definieren Sie die Substitution durch “niedrigere” Terme für eine vollständige Substitution.

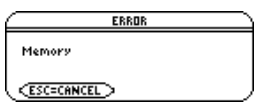
- Unendliche Rekursion kann auftreten, wenn Sie eine Substitutionsvariable durch sich selbst ersetzen.

Substituiert sin(x+1), sin(x+1+1), sin(x+1+1+1), etc.

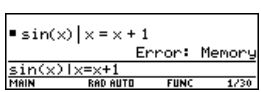
```
sin(x)|x=x+1
```

Wenn Sie eine Substitution eingegeben haben, die eine unendliche Rekursion hervorruft:

- Wird eine Fehlermeldung ausgegeben.

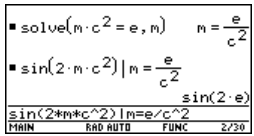


- Erscheint beim Drücken von [ESC] ein Fehler im Protokoll-Bereich.



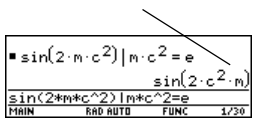
- Ein Term wird gemäß der automatischen Vereinfachungsregeln intern sortiert. Daher können Produkte und Summen eine andere Reihenfolge haben, als die, in der sie eingegeben wurden.

  - Als allgemein gültige Regel sollten Sie eine einzelne Variable substituieren.



  - Das Substituieren von allgemeineren Termen (**m • c^2=e** oder **c^2 • m=e**) läuft unter Umständen anders als erwartet ab.

Keine Übereinstimmung für die Substitution.



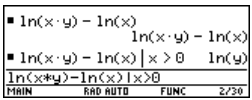
Hinweis: Verwenden Sie die **solve**-Funktion zur Unterstützung der Substitution einer einzelnen Variablen.

Bereichsbeschränkungen festlegen

Viele Identitäten und Transformationen gelten nur für einen bestimmten Bereich. Beispiel:

ln(x∗y) = ln(x) + ln(y)	nur wenn **x** und/oder **y** nicht negativ ist
sin^{-1}(sin(θ)) = θ	nur wenn θ ≥ **$^-$π/2** et θ ≤ **π/2**

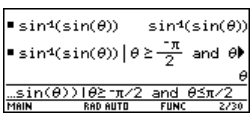
Verwenden Sie zur Angabe einer Bereichsbeschränkung den “*with*”-Operator.



Da ln(x∗y) = ln(x) + ln(y) nicht immer gültig ist, werden die Logarithmen nicht zusammengefaßt.

Mit einer Beschränkung ist die Identität gültig, und der Term wird vereinfacht.

Hinweis: Geben Sie **ln(x∗y)** statt **ln(xy)** ein, sonst wird xy als eine einzelne Variable namens xy interpretiert.



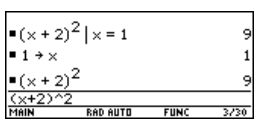
Da $\sin^{-1}(\sin(\theta)) = \theta$ nicht immer gültig ist, wird der Term nicht vereinfacht.

Mit einer Beschränkung kann der Term vereinfacht werden.

Hinweis: Für ≥ oder ≤ drücken Sie [♦] [>] oder [♦] [<].Sie können auch [2nd] [MATH] **8** oder [2nd] [UNITS] **2** verwenden, um sie aus einem Menü zu wählen.

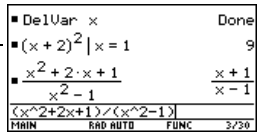
Substitutionen verwenden oder eine Variable bestimmen?

In vielen Fällen wird durch Bestimmen der Variablen der gleiche Effekt wie bei einer Substitution erzielt.

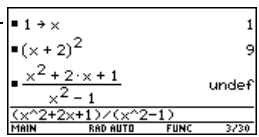


Dennoch ist die Substitution in den meisten Fällen vorzuziehen, weil die Variable hierbei nur für die aktuelle Berechnung bestimmt wird und bei späteren Berechnungen nicht wieder auftaucht.

Das Substituieren von x=1 hat keinerlei Auswirkung auf die nächste Berechnung.



Das Speichern von 1→x wirkt sich auch auf die folgenden Berechnungen aus.



Achtung: Wenn x definiert wurde, kann sich das auf alle Berechnungen mit x auswirken (solange bis x wieder gelöscht wird).

Überblick über das Algebra-Menü

Mit dem F2 **Algebra**-Symbolleistenmenü können Sie die am häufigsten verwendeten Algebrafunktionen wählen.

Das Algebra-Menü

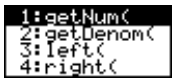
Drücken Sie vom Hauptbildschirm aus F2, um folgendes Menü zu öffnen:

Auf dieses Menü können Sie auch über das MATH-Menü zugreifen. Drücken Sie 2nd [MATH], und wählen Sie dann **9:Algebra**.

Hinweis: Eine vollständige Beschreibung jeder einzelnen Funktion und ihrer Syntax finden Sie in das modul *Technische Referenz*.

Menüpunkt	Beschreibung
solve	Löst eine Gleichung nach einer bestimmten Variablen auf. Hierdurch erhalten Sie, unabhängig von der **Complex Format**-Moduseinstellung, nur reelle Ergebnisse. Verbindet Lösungen in der Antwort mit "*and*" und "*or*". (Wählen Sie für komplexe Lösungen **A:Complex** aus dem Algebra-Menü.)
factor	Faktorisiert einen Term unter Berücksichtigung aller seiner Variablen oder nur einer bestimmten Variablen.
expand	Entwickelt einen Term bezüglich all seiner Variablen oder nur einer bestimmten Variablen.
zeros	Bestimmt die Nullstellen eines Terms bezüglich einer bestimmten Variablen. Anzeige in Listenform.

Menüpunkt	Beschreibung
approx	Wertet einen Term, wo dies möglich ist, unter Verwendung von Gleitkomma-Arithmetik aus. Dies kommt der Verwendung von MODE zum Einstellen von Exact/Approx = APPROXIMATE gleich (oder der Verwendung von ♦ ENTER, um einen Term auszuwerten).
comDenom	Berechnet einen gemeinsamen Nenner für alle Glieder eines Ausdrucks und stellt den Term als gekürzten Bruch dar.
propFrac	Liefert die Partialbruchzerlegung eines Terms.
nSolve	Berechnet für eine Gleichung eine einzige Gleitkommazahl-Lösung (während **solve** mehrere Lösungen in rationaler oder symbolischer Form anzeigen kann).
Trig	Öffnet das Untermenü:
	tExpand — Entwickelt trigonometrische Terme mit Winkelsummen und Winkelvielfachen.
	tCollect — Faßt die Produkte ganzzahliger Potenzen von trigonometrischen Funktionen in Winkelsummen und Winkelvielfachen zusammen. **tCollect** ist die Gegenfunktion zu **tExpand**.
Complex	Öffnet das Untermenü: Dies sind dieselben Funktionen wie **solve**, **factor** und **zeros**; sie errechnen aber auch komplexe Ergebnisse.

Menüpunkt	Beschreibung
Extract	Öffnet das Untermenü: 
	getNum — Wendet **comDenom** an und gibt den daraus folgenden Zähler zurück.
	getDenom — Wendet **comDenom** an und gibt den daraus folgenden Nenner zurück.
	left — Gibt die linke Seite einer Gleichung oder Ungleichung zurück.
	right — Gibt die rechte Seite einer Gleichung oder Ungleichung zurück.

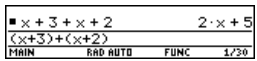
Hinweis: Die Funktionen **left** und **right** werden auch verwendet, um eine bestimmte Anzahl von Elementen oder Zeichen von der linken oder rechten Seite einer Liste oder Zeichenfolge zurückzugeben.

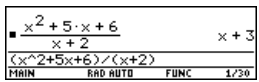
Übliche Algebra-Operationen

In diesem Abschnitt werden Beispiele für einige der Funktionen dargestellt, die im Menü F2 **Algebra** verfügbar sind. Vollständige Informationen zu allen Funktionen finden Sie im Modul *Technische Referenz*. Für manche Algebra-Operationen ist keine spezielle Funktion erforderlich.

Polynome addieren oder dividieren

Sie können Polynome direkt, ohne eine spezielle Funktion verwenden zu müssen, addieren oder dividieren.





Polynome faktorisieren und multiplizieren

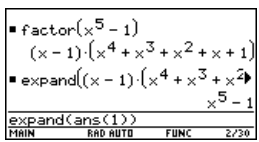
Verwenden Sie die Funktionen **factor** (F2 **2**) und **expand** (F2 **3**).

factor(*Term* [,*var*]**)**
└zum Zerlegen bezüglich einer Variablen

expand(*Term* [,*var*]**)**
└zum partiellen Entwickeln bezüglich einer Variablen

Faktorisieren Sie $\mathbf{x}^5 - \mathbf{1}$. Multiplizieren Sie dann das Ergebnis aus.

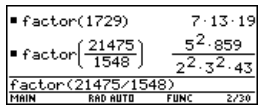
Beachten Sie, daß **factor** und **expand** gegensätzliche Operationen durchführen.



Primfaktoren einer Zahl ermitteln

Mit der Funktion **factor** (F2 **2**) können Sie mehr als nur eine einfache Zerlegung eines algebraischen Polynoms durchführen.

Sie können Primfaktoren einer rationalen Zahl ermitteln (ganze Zahl oder Quotient ganzer Zahlen).

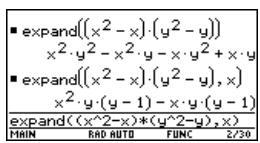


Teilweise Entwicklung

Mit dem optionalen var-Wert der Funktion **expand** (F2 **3**) können Sie eine teilweise Entwicklung vornehmen, die gleichnamige Potenzen einer Variablen zusammenfaßt.

Führen Sie eine vollständige Entwicklung von **$(x^2-x)\,(y^2-y)$** bezüglich aller Variablen durch.

Nehmen Sie dann eine teilweise Entwicklung bezüglich x vor.

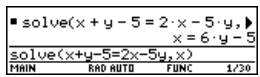


Eine Gleichung lösen

Verwenden Sie die Funktion **solve** (F2 **1**), um eine Gleichung nach einer bestimmten Variablen aufzulösen.

solve(*Gleichung*, *var***)**

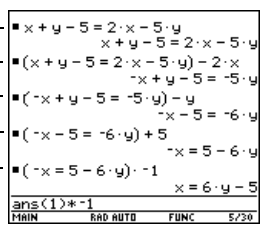
Lösen Sie **x + y − 5 = 2x − 5y** nach x auf.



Beachten Sie, dass **solve** nur das Endergebnis anzeigt.

Zwischenergebnisse können Sie sehen, indem Sie die Gleichung manuell Schritt für Schritt auflösen.

x + **y** − **5** = **2x** − **5y**
− **2 x**
− **y**
+ **5**
× (-) **1**



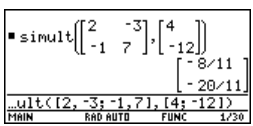
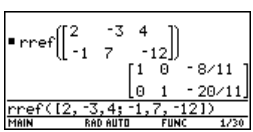
Hinweis: Bei der Operation − **2** × wird 2x von beiden Seiten subtrahiert.

Ein System linearer Gleichungen lösen

Es sei ein System mit zwei Gleichungen und zwei Unbekannten gegeben:

2x − 3y = 4
−x + 7y = −12

Dieses Gleichungssystem können Sie mit Hilfe einer der folgenden Methoden lösen.

Vorgehensweise	**Beispiel**
Verwenden Sie die **solve**-Funktion für die Lösung in einem Schritt.	Ermitteln Sie **solve(**2x−3y=4 and −x+7y=−12,{x,y}**)**
Verwenden Sie die **solve**-Funktion mit Substitution (|) für eine schrittweise Berechnung.	Siehe „*Symbolisches Rechnen"* im Modul *Mathematik-Schnellstart,* mit der Lösung für **x = −8/11** und **y = −20/11.**
Verwenden Sie die **simult**-Funktion mit einer Matrix.	Geben Sie die Koeffizienten als Matrix und die rechte Seite als konstanten Spaltenvektor ein. 
Verwenden Sie die **rref**-Funktion mit einer Matrix.	Geben Sie die Koeffizienten als erweiterte Matrix ein. 

Hinweis: Die Matrix-Funktionen **simult** und **rref** befinden sich nicht im F2 **Algebra**-Menü. Verwenden Sie 2nd [MATH] **4** oder den **Catalog**.

Die Nullstellen eines Terms ermitteln

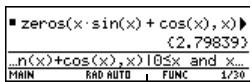
Verwenden Sie die Funktion **zeros** (F2 **4**).

zeros(*Term*, *var***)**

Verwenden Sie den Term **x**∗ **sin(x) + cos(x)**.

Ermitteln Sie die Nullstellen bezüglich x im Intervall **0** ≤ **x** und **x** ≤ **3**.

Hinweis: Für ≥ oder ≤ drücken Sie ♦ [>] oder ♦ [<]. Sie können auch 2nd [MATH] **8** oder 2nd [UNITS] **2** verwenden, um sie aus einem Menü zu wählen.



Geben Sie das Intervall mit Hilfe des “with” Operators ein.

Partialbruchzerlegungen und gemeinsame Nenner finden

Verwenden Sie die Funktionen **propFrac** (F2 **7**) und **comDenom** (F2 **6**).

propFrac(*rationaler Term* [,*var*]**)**

└ für Partialbrüche bezüglich einer Variablen

comDenom(*Term* [,*var*]**)**

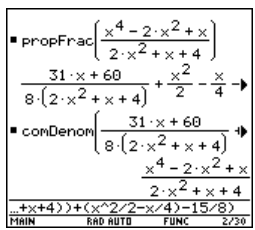
└ für gemeinsame Nenner, die gleichartige Potenzen dieser Variablen zusammenfassen

Finden Sie eine Partialbruchzerlegung für den Term: **(x^4−2x^2+x) / (2x^2+x+4)**.

Formen Sie die Lösung dann in einen einzigen Bruch um.

Beachten Sie, daß **propFrac** und **comDenom** entgegengesetzte Operationen durchführen.

Hinweis: Sie können **comDenom** mit Termen, Listen oder Matrizen verwenden.



Führen Sie dieses Beispiel an Ihrem TI-89 Titanium / Voyage™ 200 durch, scrollt die **propFrac**-Funktion über den oberen Bildschirmrand hinweg.

In diesem Beispiel gilt:

- $\frac{31x+60}{8}$ ist der Rest von **x**^{4}**−2x**^{2}**+x** dividiert durch **2x**^{2}**+x+4**.

- $\frac{x^2}{2}-\frac{x}{4}-15/8$ ist der Quotient.

Überblick über das Calc-Menü

Aus dem F3 **Calc**-Menü können Sie häufig verwendete Rechenfunktionen wählen.

Das Calc-Menü

Drücken Sie vom Ausgangsbildschirm aus F3, um folgendes Menü zu öffnen:

Auf dieses Menü können sie auch über das MATH-Menü zugreifen. Drücken Sie 2nd [MATH], und wählen Sie dann **A:Calculus**.

Hinweis: Eine vollständige Beschreibung jeder einzelnen Funktion und ihrer Syntax finden Sie im Modul *Technische Referenz.*

Menüpunkt	Beschreibung
d **differentiate**	Bildet die Ableitung eines Terms nach einer definierten Variablen.

Menüpunkt	Beschreibung
∫ **integrate**	Integriert einen Term bezüglich einer definierten Variablen.
limit	Berechnet den Grenzwert eines Terms bezüglich einer definierten Variablen.
Σ **sum**	Wertet einen Term für einzelne Variablenwerte innerhalb eines Bereichs aus und bildet dann deren Summe.
Π **product**	Wertet einen Term für einzelne Variablenwerte innerhalb eines Bereichs aus und berechnet dann deren Produkt.
fMin	Findet Werte einer Variablen, bei denen der Term ein Minimum ergibt.
fMax	Findet Werte einer Variablen, bei denen der Term ein Maximum ergibt.
arcLen	Liefert die Bogenlänge eines Terms in bezug auf eine definierte Variable.
taylor	Berechnet ein Taylorpolynom eines Terms bezüglich einer bestimmten Variablen.
nDeriv	Berechnet einen Differenzen-Quotienten eines Terms in Bezug auf eine definierte Variable.
nInt	Berechnet durch Quadratur (ein Näherungsverfahren unter Verwendung gewichteter Summen von Integrandenwerten) ein Integral als Gleitkommazahl.
deSolve	Ermittelt symbolisch die Lösungen vieler Differentialgleichungen erster und zweiter Ordnung mit oder ohne Anfangsbedingungen.

Hinweis: Das Ableitungssymbol d ist ein Sonderzeichen. Es kommt nicht der Eingabe von **D** über die Tastatur gleich. Verwenden Sie hierzu F3 **1** oder 2nd [d].

Übliche Operationen der Analysis

In diesem Abschnitt werden Beispiele für einige der Funktionen dargestellt, die im Menü F3 **Calc** verfügbar sind. Vollständige Informationen über alle Analysisfunktionen finden Sie im Modul *Technische Referenz*.

Integrieren und Differenzieren

Verwenden Sie die Funktionen ∫ **integrate** (F3 **2**) und d **differentiate** (F3 **1**).

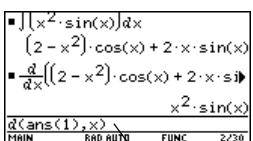
∫ (*Term*, *var* [,*unten*] [,*oben*])

zur Angabe von Integrationsgrenzen oder einer Integrationskonstanten

d (*Term*, *var* [,*Ordung*])

Integrieren Sie **x²∗sin(x)** bezüglich x.

Differenzieren Sie das Ergebnis bezüglich x.



Verwenden Sie F3 **1** oder 2nd [d], um d zu erhalten. Geben Sie nicht einfach **D** über die Tastatur ein.

Hinweis: Es können nur Terme integriert werden; differenzieren können Sie Terme, Listen oder Matrizen.

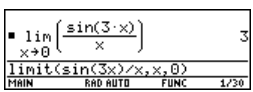
Einen Grenzwert ermitteln

Verwenden Sie die Funktion **limit** (F3 **3**).

limit(*Term*, *var*, *Komma* [,*Richtung*]**)**

- negativ = von links
- positiv = von rechts
- ausgelassen oder 0 = beides

Ermitteln Sie den Grenzwert von **sin(3x) / x**, wenn x gegen 0 geht.



Hinweis: Sie können für Terme, Listen oder Matrizen Grenzwerte ermitteln.

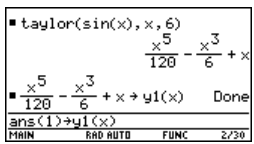
Ein Taylor Polynom ermitteln

Verwenden Sie die Funktion **taylor** (F3 **9**).

taylor(*Term*, *var*, *Ordnung* [,*Entwicklungspunkt*]**)**

└ bei Auslassung ist der Entwicklungspunkt 0

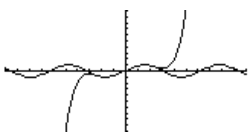
Ermitteln Sie ein Taylorpolynom der sechsten Ordnung für **sin(x)** bezüglich x.



Speichern Sie die Antwort als benutzerdefinierte Funktion namens **y1(x).**

Stellen Sie dann **sin(x)** und das Taylorpolynom graphisch dar.

Graph sin(x):Graph y1(x)



Wichtig: Die Grad-Modus-Skalierung mit π/180 kann bewirken, dass Analysis-Ergebnisse in einer anderen Form angezeigt werden.

Benutzerdefinierte Funktionen und symbolisches Rechnen

Sie können eine benutzerdefinierte Funktion als Parameter für die integrierten Algebra- und Analysisfunktionen des TI-89 Titanium / Voyage™ 200 Graphischer Rechner verwenden.

Angaben zum Erzeugen einer benutzerdefinierten Funktion

Siehe:

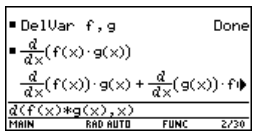
- “Benutzerdefinierte Funktionen erzeugen und auswerten” in *Weitere Funktionen des Hauptbildschirms*.
- “Eine auf dem Hauptbildschirm definierte Funktion graphisch darstellen” und “Eine teilweise definierte Funktion graphisch darstellen” in *Weitere Darstellungsarten*.
- “Überblick über die Eingabe von Funktionen” in *Programmierung*.

Undefinierte Funktionen

Sie können Funktionen wie **f(x)**, **g(t)**, **r(θ)**, etc. verwenden, welchen keine Definition zugeteilt wurde. Diese “undefinierten” Funktionen führen zu symbolischen Ergebnissen. Beispiel:

Verwenden Sie **DelVar**, um sicherzustellen, daß **f(x)** und **g(x)** nicht definiert sind.

Ermitteln Sie dann die Ableitung von **f(x)*g(x)** bezüglich x.



Hinweis: Drücken Sie F3 **1** oder 2nd [d] auf der Tastatur, um **d** aus dem Calc-Menü zu wählen.

Funktionen mit nur einer Anweisung

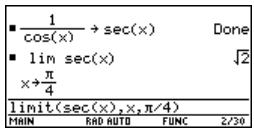
Sie können benutzerdefinierte Funktionen verwenden, die aus einem einzigen Ausdruck bestehen. Beispiel:

- Verwenden Sie [STO▸], um eine benutzerdefinierte Sekansfunktion zu erzeugen, für die gilt:

$$\sec(x) = \frac{1}{\cos(x)}$$

Ermitteln Sie dann den Grenzwert von **sec(x)**, wenn x gegen $\pi/4$ geht.

Hinweis: Drücken Sie [F3] **3**, um **limit** aus dem Calc-Menü zu wählen.

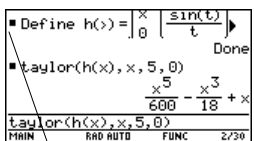


- Verwenden Sie **Define**, um eine benutzerdefinierte Funktion **h(x)** zu erzeugen, für die gilt:

$$h(x) = \int_0^x \frac{\sin(t)}{t}$$

Ermitteln Sie dann ein Taylorpolynom fünfter Ordnung für **h(x)** bezüglich x.

Hinweis: Drücken Sie [F3] **2** oder [2nd] [∫] auf der Tastatur, um ∫ aus dem Calc-Menü zu wählen. Drücken Sie [F3] **9**, um taylor zu wählen.



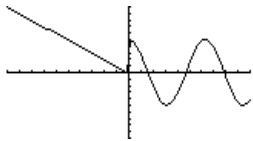
Definieren Sie h(x)= ∫(sin(t)/t,t,0,x).

Funktionen mit mehreren oder mit nur einer Anweisung

Benutzerdefinierte Funktionen mit mehreren Anweisungen sollten nur als Parameter für numerische Funktionen (wie z.B. **nDeriv** und **nInt**) verwendet werden.

In einigen Fällen ist es möglich, eine äquivalente Funktion mit nur einer Anweisung zu erzeugen. Es sei beispielsweise eine stückweise definierte Funktion mit zwei Teilen gegeben.

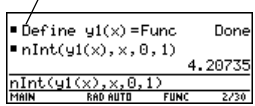
Wenn:	Verwenden Sie den Term:
$x < 0$	$-x$
$x \geq 0$	$5\cos(x)$



- Erzeugen Sie eine benutzerdefinierte Funktion mit mehreren Anweisungen in der Form:

```
Func
   If x<0 Then
   Return  x
Else
   Return 5cos(x)
EndIf
EndFunc
```

Definieren Sie
y1(x)=Func:If x<0
Then: ... :EndFunc



Integrieren Sie dann **y1(x)** numerisch bezüglich x.

Hinweis: Drücken Sie F3 **B:nInt** um **nInt** aus dem Calc-Menü zu wählen.

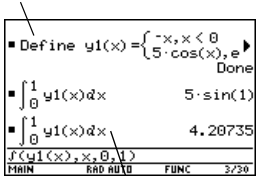
- Erzeugen Sie eine äquivalente benutzerdefinierte Funktion mit nur einer Anweisung.

  Verwenden Sie die integrierte **when**-Funktion des TI-89 Titanium / Voyage™ 200.

  Integrieren Sie dann **y1(x)** bezüglich x.

  Hinweis: Drücken Sie F3 **2** oder 2nd [∫] auf der Tastatur, um ∫ aus dem Calc-Menü zu wählen.

Definieren Sie y1(x)=when(x<0, ⁻x, 5cos(x))



Drücken Sie ♦ ENTER, um ein Gleitkommaergebnis zu erhalten.

Falls ein Speicherplatzfehler auftritt

Der TI-89 Titanium / Voyage™ 200 Graphischer Rechner speichert Zwischenergebnisse in seinem Speicher und löscht sie wieder, wenn der Rechenvorgang abgeschlossen ist. Je nach Komplexität eines Rechenvorgangs kann der Speicherplatz des TI-89 Titanium / Voyage™ 200 Graphischer Rechner jedoch erschöpft sein, bevor das Endergebnis ausgerechnet werden kann.

Speicherplatz freimachen

- Löschen Sie nicht benötigte Variablen, vor allem die besonders großen.

- Mit Hilfe von 2nd [VAR-LINK] (Beschreibung in *Speicher-und Variablen-Verwaltung*) werden Variablen und/oder Flash-Anwendungen angezeigt und gelöscht.

- Im Hauptbildschirm:
  - Löschen Sie den Protokoll-Bereich (F1 **8**) oder nur nicht benötigte Protokoll-Paare.
  - Sie können die Anzahl der Protokoll-Paare, die gespeichert werden, auch mit F1 **9** verringern.
- Setzen Sie mit MODE den Modus Exact/Approx = APPROXIMATE. (Bei Ergebnissen mit einer großen Anzahl von Stellen wird dadurch weniger Speicherplatz als bei AUTO oder EXACT benötigt. Bei Ergebnissen mit nur wenigen Stellen wird dadurch mehr Speicherplatz benötigt.)

Probleme vereinfachen

- Teilen Sie das Problem auf.
  - Teilen Sie **solve(**a∗b=0,*var***)** auf in **solve(**a=0,*var***)** und **solve(**b=0,*var***)**. Lösen Sie jeden Teilbereich, und kombinieren Sie die Ergebnisse.
- Tauchen einige undefinierte Variablen nur in einer bestimmten Kombination auf, dann ersetzen Sie diese Kombination durch eine einzige Variable.
  - Wenn m und c nur in Form von **m**∗**c**^{2} vorkommen, dann ersetzen Sie **m**∗**c**^{2} durch e.
  - In der Formel $\frac{(a+b)^2+\sqrt{(a+b)^2}}{1-(a+b)^2}$ ersetzen Sie (a+b) durch c und verwenden $\frac{c^2+\sqrt{c^2}}{1-c^2}$. Ersetzen Sie in der Lösung c durch (a+b).

- Bei zusammengesetzten Termen mit gemeinsamen Nennern ersetzen Sie die Summen in den Nennern durch neue, eindeutige undefinierte Variablen.

  - In der Formel $\frac{x}{\sqrt{a^2+b^2}+c}+\frac{y}{\sqrt{a^2+b^2}+c}$ ersetzen Sie $\sqrt{a^2+b^2}+c$ durch d und verwenden $\frac{x}{d}+\frac{y}{d}$. Ersetzen Sie in der Lösung d durch $\sqrt{a^2+b^2}+c$.

- Substituieren Sie bekannte Zahlenwerte für undefinierte Variablen möglichst bald, besonders, wenn es sich dabei um einfache ganze Zahlen oder Brüche handelt.
- Formulieren Sie eine Aufgabe um, um Bruchpotenzen zu vermeiden.
- Lassen Sie relativ kleine Terme weg, um Näherungslösungen zu finden.

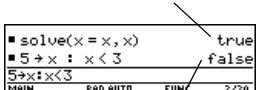
Besondere Konstanten beim symbolischen Rechnen

Ein Rechenergebnis kann eine der in diesem Abschnitt beschriebene besondere Konstante enthalten. In einigen Fällen kann es auch vorkommen, dass Sie eine Konstante als Teil Ihrer Eingabe eingeben müssen.

true, false

Geben den Wahrheitswert einer Identität oder eines Booleschen Ausdrucks an.

x=x ist wahr bei jedem beliebigen Wert von x.

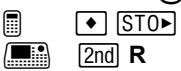


5<3 ist falsch.

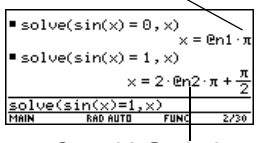
Diese Schreibweise gibt eine “beliebige ganze Zahl” an.

Taucht eine willkürliche ganze Zahl in derselben Sitzung mehrmals auf, wird jedes Erscheinen fortlaufend durchnumeriert. Nach 255 beginnt die fortlaufende Numerierung der willkürlichen ganzen Zahlen wieder bei @n0. Mit Clean Up **2:NewProb** kann auf @n1 zurückgesetzt werden.

Hinweis: Für @ drücken Sie:



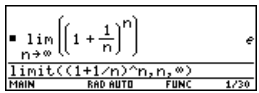
Eine Lösung ist bei jedem ganzzahligen Vielfachen von π gegeben.



Sowohl @n1 als auch @n2 stellen jede beliebige ganze Zahl dar, aber diese Schreibweise kennzeichnet unterschiedliche Ergebnisse.

∞, e

∞ bezeichnet unendlich und e die Konstante 2.71828... (Euler'sche zahl, Basis der natürlichen Logarithmen).



Diese Konstanten werden sowohl in Eingaben als auch in Ergebnissen häufig verwendet.

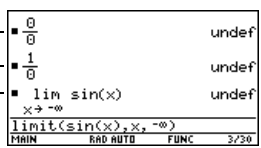
Hinweis:
Für ∞ drücken Sie:
[◆] [∞]
[2nd] [∞]

Für *e* drücken Sie:
[◆] [e^x]
[2nd] [e^x]

undef

Kennzeichnet ein undefiniertes Ergebnis.

Mathematisch unbestimmt
±∞ (unbestimmtes Zeichen)
Ein nicht-eindeutiger Grenzwert



Konstanten und Maßeinheiten

Konstanten oder Einheiten eingeben

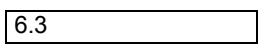
Sie können entweder über ein Menü aus einer Liste verfügbarer Konstanten und Einheiten wählen oder diese direkt über die Tastatur eingeben.

Über ein Menü

Hier wird dargestellt, wie Sie eine Einheit wählen. Verfahren Sie ebenso bei der Wahl einer Konstanten.

Im Hauptbildschirm:

1. Tippen Sie den Wert oder den Term ein.



2. Rufen Sie das Dialogfeld **UNITS** wie folgt auf:

   [2nd] [UNITS]

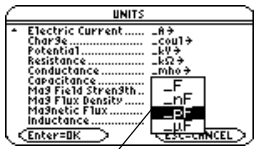
   [♦] [UNITS]

3. Setzen Sie den Cursor mit ▼ und ▲ auf die betreffende Kategorie.

   Hinweis: Durchblättern Sie die Kategorien Seite für Seite mit [2nd] ▼ und [2nd] ▲.

4. Drücken Sie ENTER, um die markierte (Standard) Einheit zu wählen,
   – oder –
   Wenn Sie eine andere Einheit aus der Kategorie auswählen möchten, drücken Sie ▶, markieren Sie die gewünschte Einheit und drücken Sie dann ENTER.

   Hinweis: Haben Sie eine eigene Einheit für eine bereits vorhandene Kategorie erstellt (Siehe "Benutzerdefinierte Einheiten erstellen" seite XX), wird diese im Menü aufgeführt.



Cursorsteuerung ist auch durch Eingabe des Anfangsbuchstabens eines Namens möglich.

Die gewählte Einheit wird in die Eingabezeile eingefügt. Konstanten- und Einheitennamen beginnen stets mit einem Unterstrich (_).

6.3_pF

Über die Tastatur

Ist Ihnen bekannt, welche Abkürzung der TI-89 Titanium / Voyage™ 200 für eine bestimmte Konstante oder Einheit verwendet , können Sie diese direkt über die Tastatur eingeben. Beispiel:

256_m

- Das erste Zeichen muß ein Unterstrich (_) sein; dieses Zeichen wird wie folgt erzeugt:
  ◆ [_]
  2nd [_]

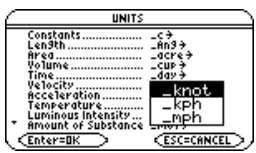
- Nicht obligatorisch ist ein Leerzeichen oder ein Multiplikationssymbol ( ) vor dem Unterstrich. **256_m, 256 _m**, und **256∗_m** sind zum Beispiel äquivalente Eingabeformen.
    - Beim Anfügen von Einheiten an eine Variable muß allerdings entweder ein Leerzeichen oder das Zeichen ∗ vor dem Unterstrich eingegeben werden. Beispielsweise wird *x_m* wird als Variable behandelt, nicht als *x* mit einer Einheit.

Hinweis: Einheiten können mit Klein- oder Großbuchstaben eingegeben werden.

Kombinieren mehrerer Einheiten

Es kann sich die Notwendigkeit ergeben, zwei oder mehr Einheiten aus unterschiedlichen Kategorien miteinander zu kombinieren.

Stellen Sie sich zum Beispiel vor, Sie möchten eine Geschwindigkeit in Metern pro Sekunde eingeben, die Kategorie **Velocity** im Dialogfeld **UNITS** enthält diese Einheit aber noch nicht.



Meter pro Sekunde läßt sich dann durch Kombination von **_m** und **_s** aus den Kategorien **Length** und **Time** eingeben.

3∗9.8_m/_s

Kombinieren Sie die Einheiten _m und _s. Es besteht keine vordefinierte Einheit _m/_s.

Hinweis: Erstellen Sie für häufig verwendete Kombinationen benutzerdefinierte Einheiten.

Verwendung von Klammern in Rechnungen mit Einheiten

In manchen Berechnungen ist es erforderlich, einen Wert und dessen Einheiten in einer Klammer **( )** zusammenzufassen, um eine korrekte Auswertung zu gewährleisten. Dies gilt insbesondere für Divisionsaufgaben. Beispiel:

Zur Berechnung von:	Geben Sie folgendes ein:
$\frac{100_m}{2_s}$	100_m/(2_s) 50 • $\frac{m}{s}$ Die Klammer für (2_s) muß gesetzt werden. Dies ist für die Division von Bedeutung. Das Weglassen der Klammer führt zu unerwarteten Einheiten. Beispiel: 100_m/2_s 50. ∗_m ∗_s

Hinweis: Sind Sie sich nicht ganz sicher, wie ein Wert und seine Einheiten ausgewertet werden, fassen Sie diese in einer Klammer zusammen **( )**.

Eine Einheit wird in einer Berechnung ähnlich wie eine Variable behandelt, deshalb erhalten Sie unerwartete Einheiten, wenn Sie die Klammer nicht setzen. Beispiel:

100_m wird als **100∗_m** behandelt
und
2_s als **2∗_s**

Ohne Klammer wird die Eingabe folgendermaßen berechnet:

100∗_m / 2∗_s = $\frac{100 * _m}{2}$ **∗_s = 50. ∗_m ∗_s**

Von einer Einheit in eine andere konvertieren

Es besteht die Möglichkeit, eine Konvertierung von einer Einheit in eine andere Einheit derselben Kategorie, einschließlich einer benutzerdefinierten, vorzunehmen (Benutzerdefinierte Einheiten erstellen).

Alle Einheiten außer Temperatureinheiten

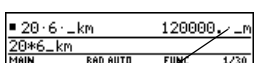
Verwenden Sie eine Einheit in einer Berechnung, so wird sie automatisch in die aktuelle Standardeinheit für diese Kategorie konvertiert und in dieser Form angezeigt, es sei denn, Sie verwenden, wie weiter unten beschrieben, den Konvertierungsoperator ▸. In folgendem Beispiel wird davon ausgegangen, daß Ihre Standardeinheiten auf das SI-System eingestellt sind (Standardeinheiten für angezeigte Ergebnisse einstellen).

Hinweis:

- Beachten Sie die Liste vordefinierter Einheiten.
- Das Dialogfeld **UNITS** enthält ein Menü mit den verfügbaren Maßeinheiten.

So multiplizieren Sie 6 Kilometer mit 20.

20∗6_km



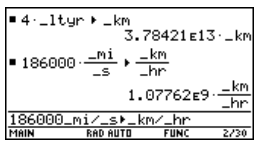
Angezeigt in der Standardlängeneinheit (_m im SI-System).

Soll in eine andere als die Standardeinheit konvertiert werden, verwenden Sie den Konvertierungsoperator ▸.

Term_Einheit1 ▸ *_Einheit2*

└ Zur Eingabe von ▸ drücken Sie 2nd ▸.

So konvertieren Sie 4 Lichtjahre in Kilometer:



4_ltyr ▸ _km

So konvertieren Sie 186000 Meilen/Sekunde in Kilometer/Stunde:

186000_mi/_s ▸ _km/_hr

Verwendet ein Term eine Kombination von Einheiten, so können Sie nur für einige der Einheiten eine Konvertierung angeben. Die übrigen Einheiten werden gemäß Ihren Standardeinstellungen angezeigt.

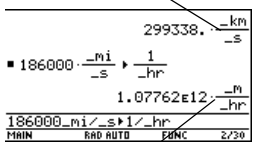
So konvertieren Sie 186000 Meilen/Sekunde von Meilen in Kilometer:

186000_mi/_s ▸ _km

So konvertieren Sie 186000 Meilen/Sekunde von Sekunden in Stunden:

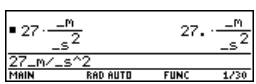
186000_mi/_s ▸ 1/_hr

Da keine Konvertierung der Zeit angegeben ist, wird diese in ihrer Standardeinheit angezeigt (in diesem Beispiel _s).



Da keine Konvertierung der Länge angegeben ist, wird diese in ihrer Standardeinheit angezeigt (in diesem Beispiel _m).

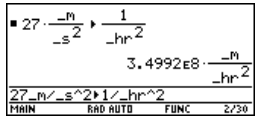
So geben Sie Meter pro Sekunde im Quadrat ein:



27_m/_s^2

So konvertieren Sie Meter pro Sekunde im Quadrat von Sekunden in Stunden:

27_m/_s^2 ▸ 1/_hr^2



Temperaturwerte

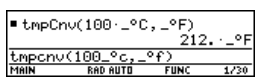
Zum Konvertieren eines Temperaturwertes ist **tmpCnv()** anstelle des Operators ▸ zu verwenden.

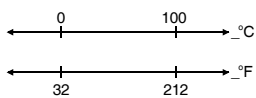
tmpCnv(*Term*_°*TempEinheit1*, _°*TempEinheit2***)**

└ Zur Eingabe von ° drücken Sie [2nd] [°]

So konvertieren Sie z.B. 100_°C in _°F:

tmpCnv(*100*_°*c*, _°*f***)**



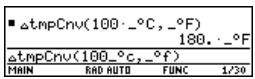


Temperaturbereiche

Verwenden Sie zum Konvertiern eines Temperaturbereichs (Differenz zwischen zwei Temperaturwerten) **ΔtmpCnv().**

ΔtmpCnv(*Term_°TempEinheit1, _°TempEinheit2*)

So konvertieren Sie z.B. den Bereich 100_°C in den entsprechenden Bereich in _°F:

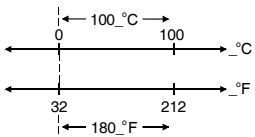


ΔtmpCnv(*100_°c, _°f*)

Hinweis: Für Δ, drücken Sie:

[D]

G D



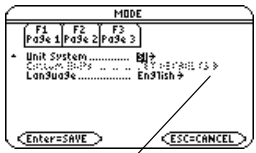
Standardeinheiten für angezeigte Ergebnisse einstellen

Alle Ergebnisse, welche Einheiten enthalten, werden in der Standardeinheit für die betroffene Kategorie angezeigt. Ist die Standardeinheit für Länge beispielsweise *_m*, dann wird jedes Längenergebnis in Metern angezeigt (auch wenn Sie in die Rechnung *_km* oder *_ft* eingegeben haben).

Bei Verwendung des SI- oder des ENG/US-Systems

Die Einheitensysteme **SI** und **ENG/US** (Einstellung über **Seite 3** des **MODE**-Bildschirms) verwenden integrierte Standardeinheiten, die nicht geändert werden können.

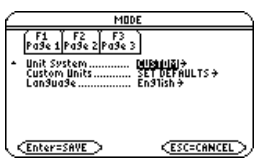
Die Standardeinheiten für diese Systeme sind verfügbar.



Wenn Unit System=SI oder ENG/US, ist der Menüpunkt Custom Units nicht verfügbar. Es kann keine Vorgabe für einzelne Kategorien eingestellt werden.

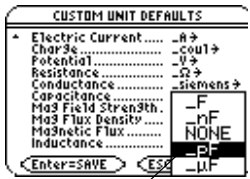
Einstellung benutzerspezifischer Standardeinheiten

So stellen Sie benutzerspezifische Standardeinheiten ein:



1. Drücken Sie MODE **F3** ▶ **3**, um **Unit System auf CUSTOM** einzustellen.
2. Markieren Sie **SET DEFAULTS** mit ▼.
3. Drücken Sie ▶, um das Dialogfeld **CUSTOM UNIT DEFAULTS** anzuzeigen.

4. Nun kann für jede Kategorie eine Vorgabe markiert werden: Drücken Sie ▶, und wählen Sie eine Einheit aus der Liste.
5. Drücken Sie zweimal ENTER, um Ihre Änderungen zu speichern und den **MODE**-Bildschirm zu verlassen.



Der Cursor bewegt sich auch, indem Sie alpha drücken und den Anfangsbuchstaben einer Einheit eingeben.

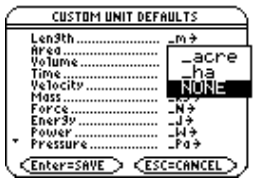
Hinweis:

- Mit **setUnits()** können Sie Standardeinheiten einstellen und mit **getUnits()** Informationen über diese abrufen. Siehe Modul *Technische Referenz*.
- Wenn das Dialogfeld **CUSTOM UNIT DEFAULTS** geöffnet wird, zeigt es zunächst die aktuellen Standardeinheiten an.

Was bedeutet die Vorgabe NONE?

Für viele Kategorien kann **NONE** als Standardeinheit gewählt werden.

Das bedeutet, dass Ergebnisse dieser Kategorie in den Standardeinheiten ihrer Komponenten angezeigt werden.



Zum Beispiel: **Area = Length**^{2}, dann ist **Length** die Komponente von **Area**.

- Sind die Standardeinstellungen **Area=_acre** und **Length=_m** (Meter), dann werden die Flächen-Ergebnisse in **_acre**-Einheiten angezeigt.
- Stellen Sie **Area auf NONE** ein, so werden die Flächen-Ergebnisse in **_m**^{2}-Einheiten angezeigt.

Hinweis: NONE ist für Grundkategorien wie **Länge** und **Masse**, die keine Komponenten besitzen, nicht verfügbar.

Benutzerdefinierte Einheiten erstellen

Die Liste der verfügbaren Einheiten jeder Kategorie kann erweitert werden, indem Sie einer einer oder mehreren vordefinierten Einheiten eine neue Einheit definieren. Auch die Verwendung eigenständiger Einheiten ist möglich.

Wozu dienen benutzerdefinierte Einheiten?

Einige Gründe für die Erstellung einer eigenen Einheit sind:

- Sie möchten Längenwerte in Dekametern eingeben. Definieren Sie deshalb *10_m* als neue Einheit mit dem Namen *_dekm*.
- Anstatt *_m/_s*^{2} als Beschleunigungseinheit einzugeben, definieren Sie diese Einheitenkombination als einfache Einheit namens *_ms2*.
- Sie möchten berechnen, wie häufig eine Person blinzelt. Sie können hierfür *_blinzeln* als gültige Einheit verwenden, ohne sie zu definieren. Diese eigenständige Einheit wird wie eine undefinierte Variable behandelt. *3_blinzeln* wird beispielsweise wie *3a* behandelt.

Hinweis: In einer Kategorie erzeugte Maßeinheiten können im Menü des Dialogfelds **UNITS** ausgewählt werden. Die Einheit kann aber nicht mit MODE als Standardeinheit für die angezeigten Ergebnisse eingestellt werden.

Regeln für benutzerdefinierte Einheitennamen

Für Einheitennamen gelten ähnliche Regeln wie für Variablennamen.

- Der Name darf aus maximal 8 Zeichen bestehen.
- Das erste Zeichen muß ein Unterstrich-Zeichen sein. Dieses Zeichen wird wie folgt erzeugt:
  ◆ [_]
  2nd [_]
- Als zweites Zeichen kann jedes für Variablennamen gültige Zeichen, außer _ oder einer Ziffer verwendet werden. *_9f* ist zum Beispiel unzulässig.
- Für die übrigen Zeichen (bis zu 6) kann jedes für Variablennamen gültige Zeichen, außer einem Unterstrich-Zeichen verwendet werden.

Definieren einer Einheit

Gehen Sie zum Definieren einer Einheit ebenso wie zum Speichern einer Variablen vor.

Definition → *_neueEinheit*
└ Erzeugung von → mit STO▸

So definieren Sie beispielsweise eine Dekameter-Einheit:

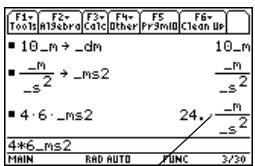
10_m →_dekm

So definieren Sie eine Beschleunigungseinheit:

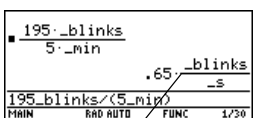
_m/_s^2 →_ms2

Zur Berechnung von 195 mal Blinzeln in 5 Minuten als **_blinzeln/_min:**

195_blinzeln/(5_min)



Wenn die Vorgaben für Länge und Zeit _m bzw. _s sind.



Wenn die Vorgabe für Zeit _ s ist.

Hinweis:

- Unabhängig davon, ob Sie die Buchstaben in Groß- oder Kleinschreibung eingeben, werden benutzerdefinierte Einheiten stets in Kleinbuchstaben angezeigt.
- **Hinweis:** Benutzerdefinierte Einheiten wie *_dm* werden als Variablen gespeichert. Sie können wie jede Variable gelöscht werden.

Liste vordefinierter Konstanten und Einheiten

In diesem Abschnitt werden die vordefinierten Konstanten und Einheiten nach Kategorien aufgeführt. All diese Konstanten und Einheiten können aus dem Dialogfeld **UNITS** gewählt werden. Beachten Sie bei der Verwendung von MODE zur Einstellung von Standardeinheiten, dass Kategorien mit nur einer definierten Einheit nicht aufgeführt sind.

Vorgaben für SI und ENG/US

Für die Einheitensysteme **SI** und **ENG/US** werden integrierte Standardeinheiten verwendet. Diese eingebauten Vorgaben sind in diesem Abschnitt durch (**SI**) und (**ENG/US**) gekennzeichnet. Für einige Kategorien verwenden beide Systeme dieselben Vorgaben.

Beachten Sie bitte, dass einige Kategorien keinerlei Standardeinheit besitzen.

Konstanten

	Beschreibung	**Wert**
_c	Lichtgeschwindigkeit	2.99792458E8_m/_s
_Cc	Coulombsche Konstante	8.9875517873682E9_N•m^2/_$coul^2$
_g	Schwerebeschleunigung	9.80665_m/_s^2
_Gc	Gravitationskonstante	6.67259E-11_m^3/_kg/_s^2
_h	Planck-Konstante	6.6260755E-34_J•_s

	Beschreibung	Wert
_k	Boltzmann-Konstante	1.380658E-23_J/_°K
_Me	Ruhemasse des Elektrons	9.1093897E-31_kg
_Mn	Ruhemasse des Neutrons	1.6749286E-27_kg
_Mp	Ruhemasse des Protons	1.6726231E-27_kg
_Na	Avogadrosche Zahl	6.0221367E23 /_mol
_q	Elektronenladung	1.60217733E-19_coul
_Rb	Bohr-Radius	5.29177249E-11_m
_Rc	allgemeine Gaskonstante	8.31451_J/_mol/_°K
_Rdb	Rydberg-Konstante	10973731.53413 /_m
_Vm	Molvolumen	2.241409E-2_m^3/_mol
_ε0	elektrische Feldkonstante	8.8541878176204E-12_F/_m
_σ	Stefan-Boltzmannsche Konstante	5.6705119E-8_W/_m^2/_°K^4
_ϕ0	Flußquant	2.0678346161E-15_Wb
_μ0	magnetische Feldkonstante	1.2566370614359E-6_N/_A^2
_μb	Bohrsches Magneton	9.2740154E-24_J•_m^2/_Wb

Hinweis:

- Der TI-89 Titanium / Voyage™ 200vereinfacht Einheitenterme und zeigt die Ergebnisse gemäß Ihrer Einstellung für Standardeinheiten an. Deshalb können die

auf Ihrem Bildschirm angezeigten konstanten Werte von den Werten in dieser Tabelle abweichen.

- Griechische Buchstaben finden Sie in der *Schnellübersicht der Tastaturbelegungen*.

Länge

_Ang	Ångström	_mi	Meile
_au	astronomische Einheit	_mil	1/1000 Zoll
_cm	Zentimeter	_mm	Millimeter
_fath	Fathom	_Nmi	Seemeile
_fm	Femtometer	_pc	Parsec
_ft	Fuß (ENG/US)	_rod	Rod
_in	Zoll	_yd	Yard
_km	Kilometer	_µ	Mikrometer
_ltyr	Lichtjahr	_Å	Ångström
_m	Meter (SI)		

Fläche

_acre	Acre	NONE	(SI) (ENG/US)
_ha	Hektar		

Volumen

_cup	Cup	_ml	Milliliter
_floz	fluid ounce	_pt	Pint
_flozUK	fluid ounce (UK)	_qt	Quart
_gal	Gallone (US)	_tbsp	Eßlöffel
_galUK	Gallone (UK)	_tsp	Teelöffel
_l	Liter	NONE	(SI) (ENG/US)

Zeit

_day	Tag	_s	Sekunde (SI) (ENG/US)
_hr	Stunde	_week	Woche
_min	Minute	_yr	Jahr
_ms	Millisekunde	_µs	Mikrosekunde
_ns	Nanosekunde		

Geschwindigkeit

_knot	Knoten	_mph	Meilen pro Stunde
_kph	Stundenkilometer	NONE	(SI) (ENG/US)

Beschleunigung

keine vordefinierten Einheiten

Temperatur

_°C	°Celsius (Zur Eingabe von °drücken Sie 2nd [°].)	_°K	°Kelvin
_°F	°Fahrenheit	_°R	°Rankine (keine Vorgabe)

Lichtstärke

_cd	Candela (keine Vorgabe)

Stoffmenge

_mol	Mol (keine Vorgabe)

Masse

_amu	atomare Masseneinheit	_oz	Unze
_gm	Gramm	_slug	Slug
_kg	Kilogramm (SI)	_ton	Tonne
_lb	Pfund (ENG/US)	_tonne	Tonne

_mg	Milligramm	_tonUK	Longtonne
_mton	Tonne		

Kraft

_dyne	Dyn	_N	Newton (SI)
_kgf	Kraftkilogramm	_tonf	Krafttonne
_lbf	Kraftpfund (ENG/US)		

Energie

_Btu	Btu (obsolete britische Einheit für Arbeit und Energie) (ENG/US)	_J	Joule (SI)
_cal	Kalorie	_kcal	Kilokalorie
_erg	Erg	_kWh	Kilowatt-Stunde
_eV	Elektronenvolt	_latm	Liter-Atmosphäre
_ftlb	Fuß-Pfund		

Leistung

_hp	Pferdestärke (ENG/US)	_W	Watt (SI)
_kW	Kilowatt		

Druck

_atm	Atmosphäre	_mmHg	Millimeter Quecksilbersäule
_bar	Bar	_Pa	Pascal (SI)
_inH2O	Zoll Wasser	_psi	Pfund pro Quadratzoll (ENG/US)
_inHg	Zoll Quecksilbersäule	_torr	Millimeter Quecksilbersäule
_mmH2O	Millimeter Wasser		

Viskosität, kinematische

_St	Stokes

Viskosität, dynamische

_P	Poise

Frequenz

_GHz	Gigahertz	_kHz	Kilohertz
_Hz	Hertz (SI) (ENG/US)	_MHz	Megahertz

Elektrischer Strom

_A	Ampere (SI) (ENG/US)	_mA	Milliampere
_kA	Kiloampere	_μA	Mikroampere

Ladung

_coul	Coulomb (SI) (ENG/US)

Potential

_kV	Kilovolt	_V	Volt (SI) (ENG/US)
_mV	Millivolt	_volt	Volt

Widerstand

_kΩ	Kiloohm	_ohm	Ohm
_MΩ	Megaohm	_Ω	Ohm (SI) (ENG/US)

Leitfähigkeit

_mho	mho (ENG/US)	_siemens	siemens (SI)
_mmho	millimho	_μmho	micromho

Kapazität

_F	Farad (SI) (ENG/US)	_pF	Pikofarad
_nF	Nanofarad	_µF	Mikrofarad

Magn. Feldstärke

_Oe	Oersted	NONE	(SI) (ENG/US)

Magn. Flußdichte

_Gs	Gauß	_T	Tesla (SI) (ENG/US)

Magnetischer Fluß

_Wb	Weber (SI) (ENG/US)

Induktivität

_henry	Henry (SI) (ENG/US)	_nH	Nanohenry
_mH	Millihenry	_µH	Mikrohenry

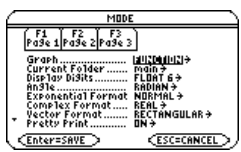
Graphische Darstellung von Funktionen

Schritte zur graphischen Darstellung von Funktionen

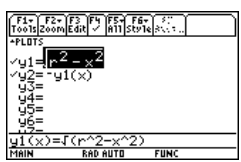
Eine oder mehrere **y(x)**-Funktionen können Sie gemäß den im Folgenden beschriebenen Schritten graphisch darstellen. Eine detaillierte Beschreibung zu jedem Schritt finden Sie auf den nachfolgenden Seiten. Wahrscheinlich müssen Sie nicht bei jeder graphischen Darstellung einer Funktion alle Schritte durchführen.

Funktionen graphisch darstellen

1. Stellen Sie den **Graph**-Modus (MODE) auf **FUNCTION** und, wenn nötig, stellen Sie den **Angle**-Modus ein.



2. Definieren Sie die Funktionen im Y= Editor (◆ [Y=]).
3. Wählen Sie (F4), welche definierten Funktionen graphisch dargestellt werden sollen.



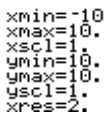
   Hinweis: Möchten Sie eventuelle stat-Daten-Plots ausschalten, drücken Sie F5 **5**, oder verwenden Sie F4.

4. Stellen Sie den Anzeigestil für eine Funktion ein.

[2nd] [F6]
[F6]

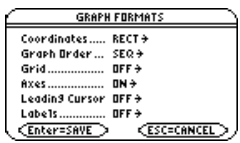
Dies ist eine Option. Mehrere Funktionen können so optisch voneinander unterschieden werden.

5. Definieren Sie das Ansichtfenster ([♦] [WINDOW]).



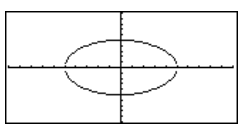
[F2] **Zoom** ändert auch das Ansichtfenster.

6. Ändern Sie bei Bedarf das Graph-Format.



[F1] **9**
– oder –
[♦] [|]
[♦] **F**

7. Stellen Sie die gewählten Funktionen graphisch dar ([♦] [GRAPH]).



Den Graphen untersuchen

Im Graphikbildschirm können Sie:

- Die Koordinaten jedes Pixels mittels des frei beweglichen Cursors oder die Koordinaten eines geplotteten Punktes durch das Tracen einer Funktion anzeigen.

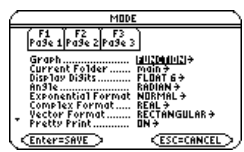
- Einen Abschnitt des Graphen können Sie mit dem F2 **Zoom**-Menü verkleinern oder vergrößern.
- Einen Nullpunkt, Tief- und Hochpunkt etc. können Sie mit dem F5 **Math**-Menü ermitteln.

Den Graph-Modus einstellen

Bevor Sie **y(x)**-Funktionen graphisch darstellen, müssen Sie FUNCTION wählen. Es könnte auch erforderlich sein, den Angle-Modus einzustellen, der bestimmt, wie der TI-89 Titanium / Voyage™ 200 Graphischer Rechner trigonometrische Funktionen darstellt.

Graph-Modus

1. Drücken Sie MODE, um das **MODE**-Dialogfeld mit den aktuellen Modus-Einstellungen zu öffnen.



2. Stellen Sie den Graph-Modus auf **FUNCTION**. Siehe “Betriebsarten einstellen” in *Bedienung des Taschenrechners*.

   Stellen Sie bei Graphen ohne komplexe Zahlen **Complex Format** auf **REAL** ein. Ansonsten könnten Graphen mit Potenzen, wie $\mathbf{x}^{1/3}$ betroffen werden.

In diesem Modul werden zwar die **y(x)**-Funktionen genauer beschrieben, der TI-89 Titanium / Voyage™ 200 bietet Ihnen jedoch sechs verschiedene Graph-Modus-Einstellungen zur Auswahl an.

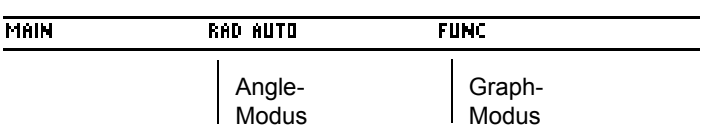
Graph-Modus-Einstellung	Beschreibung
FUNCTION	**y(x)**-Funktionen
PARAMETRIC	**x(t)**-und **y(t)**-Parameterdarstellungen
POLAR	**r(θ)**-Polardarstellungen
SEQUENCE	**u(n)** -Folgen
3D	**z(x,y)** -3D-Gleichungen
DIFFERENTIAL EQUATION	**y'(t)**-Differentialgleichungen

Angle-Modus (Winkelmodus)

Stellen Sie für trigonometrische Funktionen den Angle-Modus auf die Einheit (RADIAN oder DEGREE) ein, in welcher Winkelwerte eingegeben und angezeigt werden sollen.

Die Statuszeile kontrollieren

Den aktuellen Graph- und Angle-Modus können Sie der Statuszeile am unteren Bildschirmrand entnehmen.



Funktionen für die graphische Darstellung definieren

Im Darstellungsmodus FUNCTION können Sie Funktionen mit den Namen **y1(x)** bis **y99(x)** graphisch darstellen. Verwenden Sie zum Definieren und Bearbeiten dieser Funktionen den Y= Editor. (Der Y= Editor listet die Funktionsnamen für den aktuellen Darstellungsmodus auf. Im Darstellungsmodus POLAR heißen Funktionen zum Beispiel **r1(θ)**, **r2(θ)** etc.)

Eine neue Funktion definieren

1. Drücken Sie ♦ [Y=], um den Y= Editor zu öffnen.

Plots — Scrollen Sie über y1= hinaus, finden Sie eine Liste der Stat-Plots.

Funktionsliste — Sie können die Liste der Funktionen und Definitionen durchlaufen.

Eingabezeile — Hier definieren oder bearbeiten Sie die in der Liste markierte Funktion.

Hinweis: Die Funktionsnamen werden in der Liste abgekürzt, z.B. **y1**, in der Eingabezeile wird aber der ganze Name angezeigt, z.B. **y1(x)**.

2. Drücken Sie ⊙ und ⊙, um den Cursor auf undefinierte Funktionen zu setzen. (Zum seitenweisen Scrollen verwenden Sie 2nd ⊙ und 2nd ⊙.)
3. Drücken Sie ENTER oder F3, um den Cursor in die Eingabezeile zu setzen.
4. Geben Sie den Term für die Definition der Funktion ein.

- Beim graphischen Darstellen von Funktionen ist *x* die unabhängige Variable.
- Der Term kann sich auch auf andere Variablen, einschließlich Matrizen, Aufzählungen und andere Funktionen beziehen. Nur Fließkommazahlen und Listen von Fließkommazahlen erzeugen einen Plot.

Hinweis: Bei einer undefinierten Funktion brauchen Sie weder ENTER noch F3 zu drücken. Sobald Sie schreiben, springt der Cursor in die Eingabezeile.

5. Drücken Sie ENTER, nachdem Sie den Term eingegeben haben.

In der Funktionsliste wird nun die neue Funktion angezeigt, die automatisch für die graphische Darstellung gewählt wird.

Hinweis: Wenn Sie den Cursor versehentlich in die Eingabezeile setzen, drücken Sie ESC, um ihn in die Funktionsliste zurückzusetzen.

Eine Funktion bearbeiten

Ausgangspunkt ist der Y= Editor:

1. Drücken Sie ▼ und ▲, um die Funktion zu markieren.
2. Drücken Sie ENTER oder F3, um den Cursor in die Eingabezeile zu setzen.
3. Wählen Sie eine der folgenden Methoden.
   - Bewegen Sie den Cursor innerhalb des Terms durch ▶ und ◀, und bearbeiten Sie den Term. Siehe “Eine Eingabe in der Eingabezeile bearbeiten” in *Bedienung des Taschenrechners*.

     – oder –
   - Drücken Sie CLEAR ein- oder zweimal, um den alten Term zu entfernen, und geben Sie den neuen ein.

4. Drücken Sie [ENTER].

   In der Funktionsliste wird nun die bearbeitete Funktion angezeigt, die automatisch für die graphische Darstellung gewählt wird.

   Hinweis: Möchten Sie die Änderungen nicht ausführen, drücken Sie [ESC] anstatt [ENTER].

Eine Funktion löschen

Ausgangspunkt ist der Y= Editor:

Löschen:	Vorgehensweise:
Eine Funktion aus der Funktionsliste	Markieren Sie die Funktion, und drücken Sie [←] oder [CLEAR].
Eine Funktion aus der Eingabezeile	Drücken Sie [CLEAR] ein- oder zweimal (je nach Position des Cursors), und drücken Sie dann [ENTER].
Sämtliche Funktionen	Drücken Sie [F1], und wählen Sie dann **8:Clear Functions**. Wenn Sie zum Bestätigen aufgefordert werden, drücken Sie [ENTER].

Hinweis: [F1] **8** löscht keine Stat-Plots.

Sie müssen eine Funktion nicht löschen, um zu verhindern, dass sie graphisch dargestellt wird. Wie in Graphisch darzustellende Funktionen auswählen beschrieben, können Sie die darzustellenden Funktionen auswählen.

Schnellzugriffstasten zum Bewegen des Cursors

Im Y=Editor:

Drücken Sie:	Um:
♦ ⊖ oder ♦ ⊕	Den Cursor auf Funktion 1 bzw. die letzte definierte Funktion zu verschieben. Steht der Cursor auf oder hinter der letzten definierten Funktion, bewegen Sie ihn mit ♦ ⊕ zu Funktion 99.

Im Hauptbildschirm oder in einem Programm

Sie müssen eine Funktion nicht entfernen, um zu verhindern, dass sie geplottet wird. Sie können vielmehr die Funktionen, die geplottet werden sollen, auswählen.

- Verwenden Sie die Befehle **Define** und **Graph**. Siehe:
  - “Eine im Hauptbildschirm definierte Funktion graphisch darstellen” und “Eine stückweise definierte Funktion graphisch darstellen” in *Weitere Darstellungsarten*.
  - “Überblick über die Eingabe einer Funktion” in *Programmierung*.
- Speichern Sie einen Term direkt in einer Funktionsvariablen. Siehe:
  - “Variablenwerte speichern und abrufen” in *Bedienung des Taschenrechners*.
  - “Benutzerdefinierte Funktionen erzeugen und auswerten” im Modul *Startbildschirm des Rechners*.

Hinweis: Benutzerdefinierte Funktionen können beinahe beliebig benannt werden. Möchten Sie diese in den Y= Editor aufnehmen, verwenden Sie Namen, wie **y1(x)**, **y2(x)** etc.

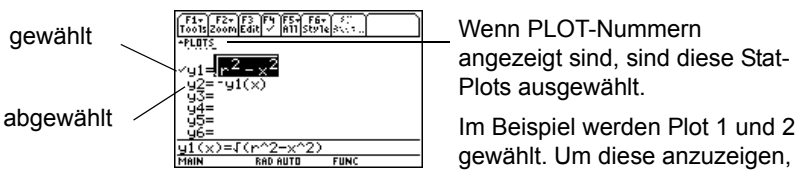
Graphisch darzustellende Funktionen auswählen

Unabhängig von der Anzahl der definierten Funktionen im Y= Editor können Sie auswählen, welche graphisch dargestellt werden sollen.

Funktionen wählen oder die Auswahl aufheben

Drücken Sie ♦ [Y=], um den Y= Editor zu öffnen.

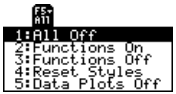
Ein "✓ " zeigt die Funktionen an, die beim nächsten Öffnen des Graphikbildschirms graphisch dargestellt werden.



Wenn PLOT-Nummern angezeigt sind, sind diese Stat-Plots ausgewählt.

Im Beispiel werden Plot 1 und 2 gewählt. Um diese anzuzeigen, scrollen Sie über y1= nach oben.

Wählen oder Auswahl aufheben:	**Vorgehensweise:**
Eine bestimmte Funktion	• Bewegen Sie den Cursor, um die Funktion zu markieren. • Drücken Sie F4. So wird eine nicht ausgewählte Funktion gewählt bzw. eine Auswahl aufgehoben.

Wählen oder Auswahl aufheben:	Vorgehensweise:
Alle Funktionen	• Drücken Sie F5, um das **All**-Menü zu öffnen. • Wählen Sie den entsprechenden Menüpunkt. 

Wenn Sie eine Funktion eingeben oder bearbeiten, müssen Sie diese nicht auswählen. Sie wird automatisch gewählt. Möchten Sie die Auswahl etwaiger Stat-Plots aufheben, drücken Sie F5 **5**, oder schließen Sie sie durch F4 aus.

Im Hauptbildschirm oder in einem Programm

Sie können Funktionen auch vom Hauptbildschirm oder von einem Programm aus wählen oder deren Auswahl aufheben.

- Verwenden Sie für Funktionen die Befehle **FnOn** und **FnOff** (verfügbar im Menü F4 **Other** des Hauptbildschirms). Siehe Modul *Technische Referenz*.
- Verwenden Sie für Stat-Plots die Befehle **PlotsOn** und **PlotsOff**. Siehe Modul *Technische Referenz*.

Den Zeichenstil einer Funktion einstellen

Sie können für jede definierte Funktion einen Stil einstellen, der bestimmt, wie die Funktion graphisch dargestellt wird. Dies ist bei der graphischen Darstellung mehrerer Funktionen hilfreich. Sie können beispielsweise eine Funktion durch eine durchgezogene Linie, eine durch eine punktierte Linie darstellen lassen etc.

Den Stil einer Funktion anzeigen oder ändern

Ausgangspunkt ist der Y= Editor:

1. Bewegen Sie den Cursor, um die gewünschte Funktion zu markieren.
2. Wählen Sie das Menü **Style** wie folgt. Drücken Sie:
   [2nd] [F6]
   [F6]

- Obwohl der Menüpunkt Line zu Beginn markiert ist, wird der aktuelle Stil der Funktion durch die Marke ✓ gekennzeichnet.
- Möchten Sie das Menü verlassen, ohne eine Änderung vorzunehmen, drücken Sie [ESC].

3. Wenn Sie eine Änderung vornehmen möchten, wählen Sie den gewünschten Stil.

Stil	Beschreibung
Line	Verbindet gezeichnete Punkte durch eine Linie. Dies ist die Standardeinstellung.
Dot	Stellt jeden gezeichneten Punkt als Pixel dar.
Square	Zeigt an jedem gezeichneten Punkt eine gefülltes Quadrat an.
Thick	Verbindet gezeichnete Punkte durch eine dicke Linie.
Animate	Ein runder Cursor bewegt sich längs des Anfangs des Graphen, hinterläßt aber keine Spur.
Path	Ein runder Cursor bewegt sich längs des Anfangs des Graphen, hinterläßt aber eine Spur.
Above	Schraffiert den Bereich oberhalb des Graphen.
Below	Schraffiert den Bereich unterhalb des dem Graphen.

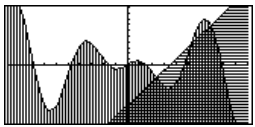
Soll Line als Stil für alle Funktionen eingestellt werden, drücken Sie F5, und wählen Sie **4:Reset Styles**.

Schraffieren oberhalb oder unterhalb des Graphen

TI-89 Titanium / Voyage™ 200 Graphischer Rechner verfügt über vier Schraffierungsarten, die abwechselnd verwendet werden. Soll eine Funktion schraffiert werden, wird die erste Art angewendet. Für die nächste zu schraffierende Funktion wird

die zweite Art angewendet usw. Bei der fünften zu schraffierende Funktion wird wieder auf die erste Art zurückgegriffen.

An den Schnittstellen schraffierter Flächen überschneiden sich die jeweiligen Schraffierungsarten.



Im Hauptbildschirm oder in einem Programm

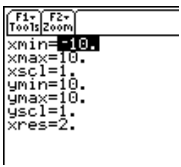
Sie können den Stil einer Funktion auch vom Hauptbildschirm oder von einem Programm aus einstellen. Siehe den Befehl **Style** im Modul *Technische Referenz*.

Das Ansichtfenster definieren

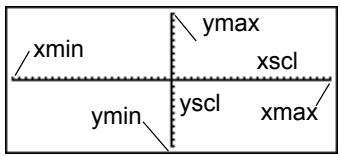
Das Ansichtfenster stellt den im Graphikbildschirm angezeigten Teil der Koordinatenebene dar. Durch die Einstellungen der Window-Variablen können Sie die Grenzen und andere Merkmale des Ansichtfensters definieren. Funktionsgraphen, Parametergraphen etc. verfügen über eigene, unabhängige Window-Variablensätze.

Window-Variable im Window-Editor anzeigen

Drücken Sie ♦ [WINDOW], um den Window Editor zu öffnen.



Window-Variablen
(angezeigt im Window-Editor)



Entsprechendes Ansichtfenster
(angezeigt im Graphikbildschirm)

Variable	Beschreibung
xmin, xmax, ymin, ymax	Grenzen des Ansichtfensters.
xscl, yscl	Einheiten auf der x- und der y-Achse.
xres	Bestimmt die Pixel-Auflösung (1 bis 10) für Funktionsgraphen. Standardeinstellung ist 2. • Bei 1 werden die Funktionen bei jedem Pixel entlang der x-Achse berechnet und dargestellt. • Bei 10 werden die Funktionen bei jedem zehnten Pixel entlang der x-Achse berechnet und dargestellt.

Zum Ausschalten der Einheiten nehmen Sie die Einstellung **xscl=0** und/oder **yscl=0** vor. Niedrige Werte für **xres** verbessern zwar die Auflösung, können aber die Zeichengeschwindigkeit herabsetzen.

Die Werte ändern

Ausgangspunkt ist der Window-Editor:

1. Bewegen Sie den Cursor, um den zu ändernden Wert zu markieren.
2. Gehen Sie nach einer der folgenden Methoden vor:
   - Geben Sie einen Wert oder einen Term ein. Sobald Sie schreiben, wird der alte Wert gelöscht.

     – oder –
   - Drücken Sie CLEAR, um den alten Wert zu löschen, und geben Sie dann den neuen ein.

     – oder –
   - Drücken Sie ◄ oder ►, um die Markierung zu entfernen, und bearbeiten Sie dann den Wert.

Werte werden mit der Eingabe gespeichert; Sie müssen nicht ENTER drücken. ENTER verschiebt den Cursor zur nächsten Window-Variablen. Geben Sie einen Term ein, wird dieser berechnet, wenn Sie den Cursor auf eine andere Window-Variable setzen oder den Window-Editor verlassen.

Im Hauptbildschirm oder in einem Programm

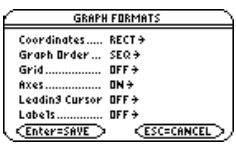
Sie können Werte auch direkt vom Hauptbildschirm oder von einem Programm aus in den Windows-Variablen speichern. Siehe “Variablenwerte speichern und abrufen” in *Bedienung des Taschenrechners*.

Das Graphik-Format ändern

Sie können das Graphik-Format so einstellen, dass die Achsen, Gitterpunkte und die Cursor-Koordinaten angezeigt oder ausgeblendet werden. Funktionsgraphen, Parametergraphen etc. verfügen über eigene, unabhängige Graph-Format-Sätze.

Graphik-Format-Einstellungen anzeigen

Wenn Sie sich im Y= Editor, im Window-Editor oder im Graphikbildschirm befinden, drücken Sie F1, und wählen **9:Format**.



- Das GRAPH FORMATS-Dialogfeld zeigt die aktuellen Einstellungen an.
- Zum Verlassen, ohne eine Änderung vorzunehmen, drücken Sie ESC.

Das Dialogfeld GRAPH FORMATS kann über die Bildschirme Y= Editor, Window Editor oder Graph auch mit folgenden Tasten aufgerufen werden:

◆ |

◆ **F**

Format	Beschreibung
Coordinates	Zeigt die Cursor-Koordinaten in kartesischer (RECT) oder Polarform (POLAR) an oder schaltet die Koordinaten aus (OFF).
Graph Order	Stellt Funktionen nacheinander (SEQ) oder gleichzeitig (SIMUL) graphisch dar.

Format	Beschreibung
Grid	Zeigt Gitterpunkte, die mit den Einheiten auf den Achsen übereinstimmen an (ON) oder blendet sie aus (OFF).
Axes	Zeigt die x- und y-Achse an (ON) oder blendet sie aus (OFF).
Leading Cursor	Zeigt einen Bezugscursor, der die Funktionen beim Zeichnen des Graphen nachzieht, an (ON) oder blendet ihn aus (OFF).
Labels	Zeigt die Bezeichnungen der x- und y-Achsen an (ON) oder blendet sie aus (OFF).

Zum Ausschalten der Teilstriche definieren Sie für das Ansichtfenster **xscl** und/oder **yscl = 0**.

Einstellungen ändern

Ausgangspunkt ist das GRAPH FORMATS-Dialogfeld:

1. Bewegen Sie den Cursor, um die Format-Einstellung zu markieren.
2. Drücken Sie ▶, um ein Menü mit den für dieses Format gültigen Einstellungen zu öffnen.
3. Wählen Sie eine Einstellung. Entweder:
   - Bewegen Sie den Cursor, um die Einstellung zu markieren, und drücken Sie dann ENTER.

     – oder –
   - Drücken Sie die Nummer dieser Einstellung.

4. Wenn Sie alle erforderlichen Änderungen der Format-Einstellungen abgeschlossen haben, drücken Sie ENTER, um Ihre Änderungen zu speichern, und schließen Sie das **GRAPH FORMATS**-Dialogfeld.

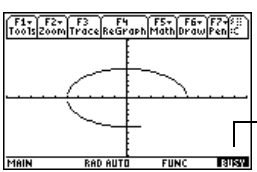
   Hinweis: Zum Entfernen eines Menüs oder Verlassen des Dialogfelds, ohne Änderungen zu speichern, verwenden Sie ESC anstatt ENTER.

Die gewählten Funktionen graphisch darstellen

Wenn Sie fertig sind, um die gewählten (aktivierten) Funktionen zu zeichnen, öffnen Sie den Graphikbildschirm. Für diesen Bildschirm werden der Anzeigestil und das Anzeigefenster verwendet, die Sie zuvor bestimmt haben.

Den Graphikbildschirm öffnen

Drücken Sie ♦ GRAPH. Der TI-89 Titanium / Voyage™ 200 Graphischer Rechner zeichnet automatisch den Graph für die ausgewählten Funktionen.



Anzeige BUSY erscheint während der Graph gezeichnet wird.

Wählen Sie im Y= Editor oder im Window Editor eine F2 **Zoom**-Operation, öffnet TI-89 Titanium / Voyage™ 200 automatisch den Graphikbildschirm.

Das Zeichnen unterbrechen

Während gezeichnet wird:

- Drücken Sie ENTER, um das Zeichnen kurzzeitig zu unterbrechen. (Die PAUSE-Anzeige ersetzt dann die Meldung BUSY.) Zur Fortsetzung drücken Sie erneut ENTER.
- Möchten Sie das Zeichnen abbrechen, drücken Sie ON. Drücken Sie F4 (**ReGraph**), um das Zeichnen von Anfang an wieder zu beginnen.

Das Ansichtfenster muß geändert werden

Je nach Einstellung kann die Zeichnung einer Funktion zu klein, zu groß oder zu sehr auf eine Bildschirmseite verschoben sein. So können Sie dies korrigieren:

- Definieren Sie das Ansichtfenster erneut, und bestimmen Sie andere Grenzen.
- Verwenden Sie eine Zoom-Operation.

Smart-Graph

Sobald Sie den Graphikbildschirm öffnen, wird durch die Smart-Graph-Funktion der vorherige Fensterinhalt angezeigt, sofern keine Änderungen vorgenommen wurden, die ein erneutes Zeichnen erfordern.

Smart-Graph führt die Aktualisierung des Fensters und das erneute Zeichnen des Graphen nur dann durch, wenn Sie:

- Eine Moduseinstellung geändert haben, die sich auf die Darstellung des Graphen, das Zeichenattribut einer Funktion, eine Window-Variable oder ein Graphik-Format auswirkt.

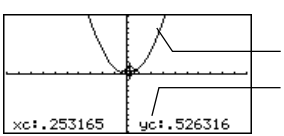
- Eine Funktion oder einen Stat-Plot ein- oder ausgeschaltet haben; (wenn Sie nur eine neue Funktion einschalten, wird diese durch Smart Graph im Graphikbildschirm hinzugefügt.)
- Die Definition einer gewählten Funktion oder den Wert einer Variablen in einer gewählten Funktion geändert haben.
- Ein gezeichnetes Objekt gelöscht haben.
- Eine Stat-Plot-Definition geändert haben.

Koordinaten mit dem frei beweglichen Cursor anzeigen

Zum Anzeigen der Koordinaten eines beliebigen Punktes auf dem Graphikbildschirm verwenden Sie den frei beweglichen Cursor. Sie können den Cursor auf jedes Pixel des Bildschirms setzen; er ist nicht an eine dargestellte Funktion gebunden.

Frei beweglicher Cursor

Beim ersten Öffnen des Graphikbildschirms ist kein Cursor sichtbar. Der Cursor wird durch Drücken einer Pfeiltaste des Cursortastenfelds aktiviert. Der Cursor bewegt sich von der Bildschirmmitte aus, und seine Koordinaten werden angezeigt.



y1(x)=x^2

“c” bedeutet, dass es sich hier um Cursor-Koordinaten handelt. Die Werte werden in den xc- und yc- System-Variablen gespeichert. Für kartesische Koordinaten steht xc und yc, für Polarkoordinaten r**c** und θc.

Werden auf dem Bildschirm keine Koordinaten angezeigt, stellen Sie das Graphik-Format auf **Coordinates = RECT** oder **POLAR** ein. Drücken Sie:

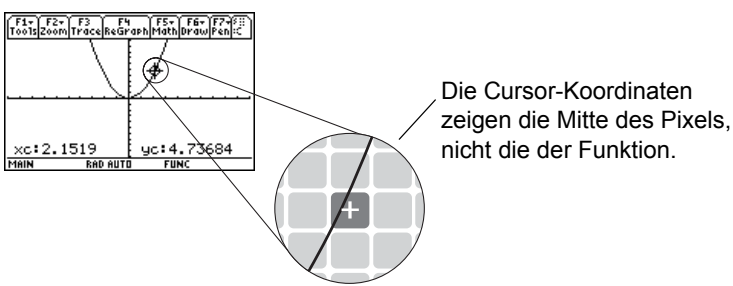
◆ |

◆ **F**

Verschieben des frei beweglichen Cursors:	**Drücken Sie:**
Auf ein angrenzendes Pixel	Cursortaste des Cursortastenfelds für beliebige Richtung.
In 10-Pixel-Schritten	2nd und dann einen Cursorpfeil.

Hinweis: Möchten Sie den Cursor und seine Koordinaten kurzzeitig ausblenden, drücken Sie CLEAR, ESC oder ENTER. Die nächste Bewegung des Cursors beginnt an seinem letzten Standort.

Setzen Sie den Cursor auf ein scheinbar “auf” der Funktion liegendes Pixel, liegt er wahrscheinlich nahe bei, nicht aber auf dem Graph der Funktion.



So erhöhen Sie die Genauigkeit:

- Verwenden Sie das auf der nachfolgenden Seite beschriebene **Trace**-Werkzeug, um die auf der Funktion liegende Koordinaten anzuzeigen.
- Vergrößern Sie einen Auschnitt des Graphen durch eine **Zoom**-Operation.

Eine Funktion tracen

Zum Anzeigen der exakten Koordinaten eines beliebigen geplotteten Punktes auf einer graphisch dargestellten Funktion verwenden Sie den F3 **Trace**-Befehl. Im Gegensatz zum frei beweglichen Cursor bewegt sich der Zeichencursor ausschließlich entlang der geplotteten Punkte einer Funktion.

Das Tracen beginnen

Drücken Sie im Graphikbildschirm F3.

Der Zeichencursor erscheint auf der Funktion, am mittleren **x**-Wert auf dem Bildschirm. Die Cursor-Koordinaten werden am unteren Bildschirmrand angezeigt.

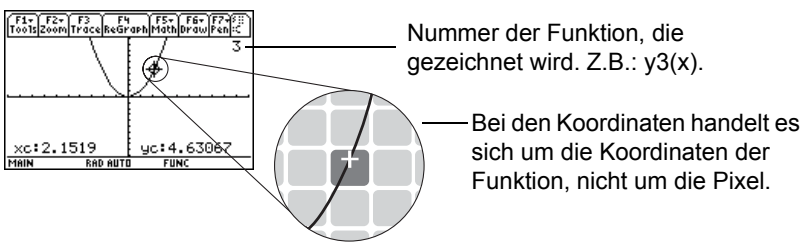
Werden mehrere Funktionen graphisch dargestellt, erscheint der Zeichencursor auf der im Y= Editor gewählten Funktion mit der niedrigsten Nummer. Die Funktionsnummer wird auf dem Bildschirm oben rechts angezeigt.

Werden Stat-Plots graphisch dargestellt, erscheint der Zeichencursor auf dem Stat-Plot mit der niedrigsten Nummer.

Bewegung entlang einer Funktion

Bewegung des Zeichencursors:	Vorgehensweise:
Zum vorigen oder nächsten geplotteten Punkt	Drücken Sie ◀ oder ▶.
Um ungefähr 5 geplottete Punkte (je nach **xres**-Window-Variable können es mehr oder weniger als 5 Punkte sein)	Drücken Sie 2nd ◀ oder 2nd ▶.
Zu einem bestimmten *x*-Wert auf der Funktion	Geben Sie den x-Wert ein, und drücken Sie ENTER.

Hinweis: Wenn Sie einen x-Wert eingeben, muß dieser zwischen **xmin** und **xmax** liegen.



Werden auf dem Bildschirm keine Koordinaten angezeigt, stellen Sie das Graph-Format auf **Coordinates = RECT** oder **POLAR** ein. Drücken Sie:

◆ |

◆ **F**

Jeder angezeigte y-Wert wird aus dem x-Wert berechnet; d.h. **y=yn(x)**. Ist die Funktion in einem x-Wert undefiniert, wird kein y-Wert angezeigt.

Sie können eine über den oberen oder unteren Rand des Ansichtfensters hinausragende Funktion auch über diese Punkte hinaus weitertracen. Sie sehen die Bewegung des Cursors in diesem Bereich außerhalb des Bildschirms zwar nicht, aber die eingeblendeten Koordinatenwerte stellen seine korrekten Koordinaten dar.

Hinweis: Zum Tracen einer über den oberen oder unteren Fensterrand hinausreichenden Funktion verwenden Sie QuickCenter.

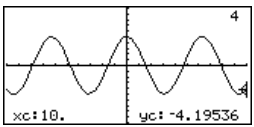
Von einer Funktion auf eine andere übergehen

Drücken Sie ▲ oder ▼, um mit demselben x-Wert auf die vorige oder nächste gewählte Funktion zu wechseln. Die Nummer der neuen Funktion wird auf dem Bildschirm angezeigt.

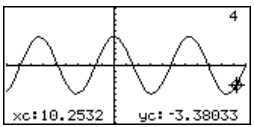
Welches die “vorige oder nächste” Funktion ist, hängt von der Reihenfolge der im Y= Editor gewählten Funktionen ab, nicht vom Erscheinungsbild der Graphen auf dem Bildschirm.

Automatischer Schwenk

Tracen Sie eine Funktion über den rechten oder linken Bildschirmrand hinaus, macht das Ansichtfenster einen automatischen Schwenk nach rechts oder links. Während der neue Graphik-Abschnitt gezeichnet wird, kommt es zu einer kurzen Unterbrechung.



Vor dem automatischen Schwenk



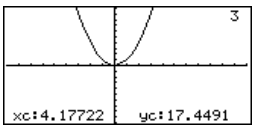
Nach dem automatischen Schwenk

Nach einem automatischen Schwenk wandert der Cursor weiter entlang des Graphen.

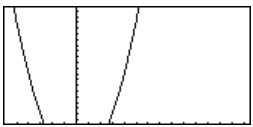
Hinweis: Das automatische Schwenk funktioniert weder, wenn Stat-Plots angezeigt noch, wenn Funktionen schraffiert dargestellt werden.

QuickCenter verwenden

Wenn Sie eine Funktion über den oberen oder unteren Rand des Ansichtfensters hinaus tracen, können Sie [ENTER] drücken, um das Ansichtfenster bezüglich der Cursorposition zu zentrieren.



Vor der Verwendung von QuickCenter



Nach der Verwendung von QuickCenter

Durch die Verwendung von QuickCenter wird das Tracen abgebrochen. Soll der Cursor weitertracen, drücken Sie [F3].

Sie können QuickCenter während des Zeichenvorgangs jederzeit verwenden, selbst wenn sich der Cursor noch auf dem Bildschirm befindet.

Das Tracen abbrechen

Möchten Sie das Tracen abbrechen, drücken Sie zu einem beliebigen Zeitpunkt [ESC].

Das Tracen wird auch durch das Öffnen eines anderen Anwendungsbildschirms, wie beispielsweise des Y= Editor, abgebrochen. Wenn Sie zum Graphikbildschirm zurückkehren und [F3] drücken, um das Tracen zu starten:

- Hat Smart Graph den Graph neu erstellt, erscheint der Cursor auf dem mittleren x-Wert.

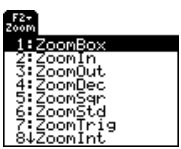
- Hat Smart Graph den Graph nicht neu erstellt, erscheint der Cursor an seiner vorherigen Position (vor dem Anzeigen der anderen Anwendung).

Einen Graphen mit Hilfe von Zoom untersuchen

Das F2 **Zoom**-Menü verfügt über mehrere Werkzeuge zum Einstellen des Ansichtfensters. Sie können ein Ansichtfenster für eine spätere Anwendung auch speichern.

Überblick über das Zoom-Menü

Drücken Sie F2 im Y= Editor, Window-Editor oder Graphikbildschirm.



Die Verwendung von **ZoomBox**, **ZoomIn**, **ZoomOut**, **ZoomStd**, **Memory** und **SetFactors** wird weiter unten in diesem Abschnitt erläutert.

Weitere Angaben zu den übrigen Menüpunkten finden Sie im Modul *Technische Referenz*.

Hinweis: Wählen Sie im Y=Editor oder im Window-Editor ein **Zoom**-Werkzeug, öffnet TI-89 Titanium / Voyage™ 200 Graphischer Rechner automatisch den Graphikbildschirm.

Zoom-Werkzeug	Beschreibung
ZoomBox	Dient zum Zeichnen eines **Zoom**-Rahmens und Vergrößern in diesem Rahmen.
ZoomIn, **ZoomOut**	Sie können einen Punkt wählen und von diesem aus um einen durch **SetFactors** bestimmten Wert vergrößern oder verkleinern.
ZoomDec	Stellt Δx und Δy auf .1 ein und zentriert auf den Ursprung.
ZoomSqr	Paßt die Window-Variablen an, so dass Quadrate oder Kreise im richtigen Verhältnis angezeigt werden (nicht als Rechtecke oder Ellipsen).
ZoomStd	Stellt die Window-Variablen auf ihre Standardwerte ein. xmin = ⁻10 ymin = ⁻10 xres = 2 xmax = 10 ymax = 10 xscl = 1 yscl = 1
ZoomTrig	Stellt die Window-Variablen auf Werte, die sich häufig für die graphische Darstellung trigonometrischer Funktionen eignen. Zentriert den Ursprung und nimmt folgende Einstellungen vor: $\Delta x = \pi/24$ (.130899... rad oder 7.5 Grad) ymin = ⁻4 ymax = 4 xscl = $\pi/2$ (1.570796... rad oder 90 Grad) yscl = 0.5
ZoomInt	Dient zur Wahl eines neuen Zentrums und stellt dann Δx und Δy auf 1 sowie **xscl** und **yscl** auf **10** ein.

Zoom-Werkzeug	Beschreibung
ZoomData	Stellt die Window-Variablen so ein, dass alle gewählten Stat-Plots angezeigt werden.
ZoomFit	Stellt das Ansichtfenster so ein, dass alle abhängigen Variablenwerte der gewählten Funktion angezeigt werden. Beim graphischen Darstellen einer Funktion werden die aktuellen Werte für **xmin** und **xmax** beibehalten, und **ymin** und **ymax** werden angepasst.
Memory	Dient zum Speichern und Abrufen von Window-Variableneinstellungen, so dass Sie ein benutzerdefiniertes Ansichtfenster wiederherstellen können.
SetFactors	Dient zum Einstellen von **Zoom**-Faktoren für **ZoomIn** und **ZoomOut**.

Δx und Δy stellen den Abstand von der Mitte eines Pixels zur Mitte eines angrenzenden Pixels dar.

Vergrößern mit einem Zoom-Rahmen

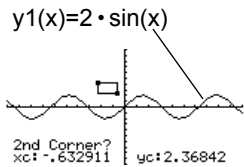
1. Wählen Sie aus dem F2 **Zoom**-Menü **1:ZoomBox**.

   Sie werden nach der ersten Ecke gefragt: **1st Corner?**

2. Setzen Sie den Cursor auf eine beliebige Ecke des Rahmens, den Sie definieren möchten, und drücken Sie dann ENTER.

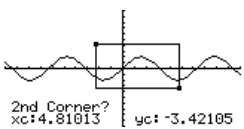
   Der Cursor nimmt die Form eines kleinen Quadrats an, und Sie werden nach der zweiten Ecke gefragt: **2nd Corner?**

   Hinweis: Möchten Sie den Cursor in größeren Schritten bewegen, verwenden Sie 2nd ▶, 2nd ▼ etc.



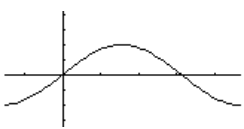
3. Setzen Sie den Cursor auf die gegenüberliegende Ecke des Zoom-Rahmens.

   Mit der Bewegung des Cursors wird der Rahmen verändert.



4. Haben Sie den Bereich, den Sie vergrößern möchten, umfaßt, drücken Sie ENTER.

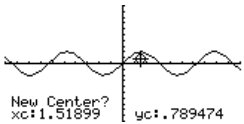
   Der Graphikbildschirm zeigt den gezoomten Bereich an. Sie können **ZoomBox** abbrechen, indem Sie vor ENTER die Taste ESC drücken.



Vergrößern und Verkleinern bezüglich eines Punktes

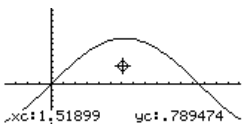
1. Wählen Sie im F2 Zoom-Menü **2:ZoomIn** oder **3:ZoomOut**.

   Ein Cursor erscheint, und Sie werden nach dem neuen Zentrum gefragt: **New Center?**



2. Verschieben Sie den Cursor an den Punkt, an welchem Sie die Vergrößerung oder Verkleinerung orientieren möchten, und drücken Sie dann ENTER.

   Der TI-89 Titanium / Voyage™ 200 ändert die Window-Variablen gemäß den in **SetFactors** bestimmten **Zoom**-Faktoren.



- Bei **ZoomIn** werden die x-Variablen durch **xFact** und die y-Variablen durch **yFact** geteilt.

  new **xmin = xmin/xFact**, etc.

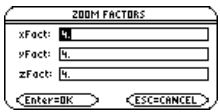
- Bei **ZoomOut** werden die x-Variablen mit **xFact** und die y-Variablen mit **yFact** multipliziert.

  new **xmin = xmin** ∗ **xFact** , etc.

Zoom-Faktoren ändern

Die Zoom-Faktoren bestimmen die Vergrößerung und Verkleinerung durch **ZoomIn** und **ZoomOut**.

1. Wählen Sie im F2 **Zoom**-Menü **C:SetFactors**, um das **ZOOM FACTORS**-Dialogfeld zu öffnen.



Zoom-Faktoren müssen ≥ 1, aber keine ganzen Zahlen sein. Standardeinstellung ist **4**.

Hinweis: Zum Verlassen, ohne Änderungen zu speichern, drücken Sie ESC.

2. Verwenden Sie ⊖ und ⊕, um den zu ändernden Wert zu markieren. Dann:
   - Geben Sie den neuen Wert ein. Der frühere Wert wird mit Beginn der Eingabe automatisch gelöscht.

     – oder –

   - Drücken Sie ◀ oder ▶, um die Markierung zu entfernen, und bearbeiten Sie dann den früheren Wert.
3. Drücken Sie ENTER (nachdem Sie in ein Eingabefeld geschrieben haben, müssen Sie zweimal ENTER drücken), um etwaige Änderungen zu speichern, und verlassen Sie das Dialogfeld.

Ein Ansichtfenster sichern oder abrufen

Nach der Verwendung verschiedener **Zoom**-Werkzeuge können Sie zu einem früheren Ansichtfenster zurückkehren oder das aktuelle Fenster sichern.

1. Wählen Sie im F2 **Zoom**-Menü **B:Memory**, um das Untermenü zu öffnen.

2. Wählen Sie den erforderlichen Menüpunkt.

Wählen Sie:	Zum
1:ZoomPrev	Zurückkehren zum Ansichtfenster, das vor dem Zoomvorgang angezeigt wurde.
2:ZoomSto	Sichern des aktuellen Ansichtfensters. (Die aktuellen Werte der Window-Variablen werden in den System-Variablen **zxmin**, **zxmax** etc. gespeichert.)
3:ZoomRcl	Abrufen des zuletzt durch **ZoomSto** gespeicherten Ansichtfensters.

Hinweis: Sie können nur jeweils einen Window-Variablen-Satz speichern. Durch das Speichern eines neuen Satzes wird der frühere Satz überschrieben.

Das Standard-Ansichtfenster wiederherstellen

Sie können die Standardwerte der Window-Variablen jederzeit wiederherstellen.

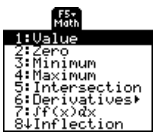
Wählen Sie im F2 **Zoom**-Menü **6:ZoomStd**.

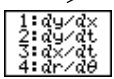
Funktionen mit Hilfe von Mathematik-Werkzeugen analysieren

Das F5 **Math**-Menü auf dem Graphikbildschirm verfügt über verschiedene Werkzeuge zum Analysieren graphisch dargestellter Funktionen.

Überblick über das Math-Menü

Drücken Sie im Graphikbildschirm F5.





Im Untermenü Derivatives ist nur dy/dx für die grafische Darstellung von Funktionen verfügbar. Die anderen Ableitungen sind für andere Graphikmodi verfügbar (Parameter-, Polardarstellung etc.)

Math-Werkzeug	Beschreibung
Value	Berechnet eine gewählte **y(x)**-Funktion an einem bestimmten x-Wert.
Zero, **Minimum**, **Maximum**	Ermittelt eine Nullstelle, Tief- oder Hochpunkt in einem Intervall.
Intersection	Ermittelt den Schnittpunkt zweier Funktionen.
Derivatives	Ermittelt die Ableitung (Steigung) in einem Punkt.

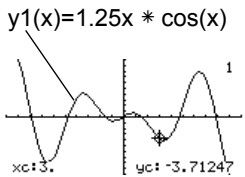
Math-Werkzeug	Beschreibung
∫f(x)dx	Ermittelt das angenäherte numerische Integral über einem Intervall.
Inflection	Ermittelt den Wendepunkt einer Kurve, an welchem deren zweite Ableitung das Vorzeichen ändert (wo die Kurve ihre Krümmung ändert).
Distance	Zeichnet und mißt eine Strecke zwischen zwei Punkten auf derselben Funktion oder auf unterschiedlichen Funktionen.
Tangent	Zeichnet eine Tangente in einem Punkt und zeigt deren Gleichung an.
Arc	Ermittelt die Bogenlänge zwischen zwei Punkten entlang einer Kurve.
Shade	Hängt von der Anzahl graphisch dargestellter Funktionen ab. • Bei nur einer graphisch dargestellten Funktion wird deren Fläche oberhalb oder unterhalb der x-Achse schraffiert. • Bei zwei oder mehr graphisch dargestellten Funktionen wird der Bereich zwischen zwei beliebigen Funktionen in einem Intervall schraffiert.

Hinweis: Bei mathematischen Ergebnissen werden die Cursor-Koordinaten in den Systemvariablen xc und yc (rc und θc bei Polarkoordinaten) gespeichert. Ableitungen, Integrale, Entfernungen etc. werden in der Systemvariablen *sysMath* gespeichert.

y(x) in einem bestimmten Punkt ermitteln

1. Drücken Sie im Graphikbildschirm [F5], und wählen Sie **1:Value**.
2. Geben Sie den x-Wert ein, bei dem es sich um einen reellen Wert zwischen **xmin** und **xmax** handeln muß. Der Wert kann ein Term sein.
3. Drücken Sie [ENTER].

   Der Cursor wird auf diesen x-Wert auf der im Y= Editor zuerst gewählten Funktion gesetzt, und seine Koordinaten werden angezeigt.



4. Drücken Sie ▼ oder ▲, um den Cursor auf den eingegebenen x-Wert anderer Funktionen zu setzen. Der entsprechende y-Wert wird angezeigt.

   Drücken Sie ◀ oder ▶ erscheint der frei bewegliche Cursor. Diesen können Sie nicht mehr auf den eingegebenen x-Wert zurücksetzen.

Sie können Funktionskoordinaten auch anzeigen, indem Sie die Funktion tracen ([F3]), einen x-Wert eingeben und [ENTER] drücken.

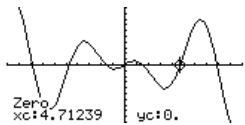
Eine Nullstelle, Tief- oder Hochpunkt in einem Intervall ermitteln

1. Drücken Sie im Graphikbildschirm [F5], und wählen Sie **2:Zero**, **3:Minimum** oder **4:Maximum**.
2. Verwenden Sie notfalls ▼ und ▲ zum Wählen der gewünschten Funktion.

   Hinweis: Grenzen können durch die Eingabe der x-Werte schnell bestimmt werden.

3. Definieren Sie die untere Grenze für x. Setzen Sie den Cursor mit ◄ und ► an die untere Grenze, oder geben Sie deren x-Wert ein.
4. Drücken Sie [ENTER]. Die Marke ▶ am oberen Bildschirmrand kennzeichnet die untere Grenze.
5. Definieren Sie die obere Grenze, und drücken Sie [ENTER].

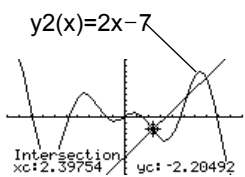
   Der Cursor wird auf die Lösung gesetzt, und seine Koordinaten werden angezeigt.



Den Schnittpunkt zweier Funktionen in einem Intervall ermitteln

1. Drücken Sie im Graphikbildschirm [F5], und wählen Sie **5:Intersection**.
2. Wählen Sie ggf. mit ▼ oder ▲, die erste Funktion, und drücken Sie [ENTER]. Der Cursor geht zur nächsten graphisch dargestellten Funktion über.
3. Wählen Sie die zweite Funktion, und drücken Sie [ENTER].
4. Definieren Sie die untere Grenze für x. Setzen Sie den Cursor mit ◄ und ► an die untere Grenze, oder geben Sie deren x-Wert ein.
5. Drücken Sie [ENTER]. Die Marke ▶ am oberen Bildschirmrand kennzeichnet die untere Grenze.
6. Definieren Sie die obere Grenze, und drücken Sie [ENTER].

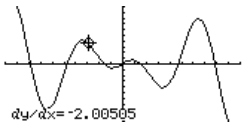
   Der Cursor wird auf den Schnittpunkt gesetzt, und seine Koordinaten werden angezeigt.



Die Ableitung (Steigung) in einem Punkt ermitteln

1. Drücken Sie im Graphikbildschirm F5, und wählen Sie **6:Derivatives**. Wählen Sie dann **1:dy/dx** aus dem Untermenü.
2. Verwenden Sie ggf. ▼ und ▲, um die gewünschte Funktion zu wählen.
3. Definieren Sie den Punkt. Setzen Sie den Cursor auf den Punkt, oder geben Sie dessen x-Wert ein.
4. Drücken Sie ENTER.

   Die Ableitung in diesem Punkt wird angezeigt.



Das numerische Integral über einem Intervall ermitteln

1. Drücken Sie im Graphikbildschirm F5, und wählen Sie **7:∫f(x)dx**.
2. Verwenden Sie ggf. ▼ und ▲, um die gewünschte Funktion zu wählen.

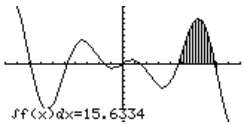
   Hinweis: Die Grenzen können durch die Eingabe der x-Werte schnell bestimmt werden.
3. Definieren Sie die untere Grenze für x. Setzen Sie den Cursor mit ◀ und ▶ an die unter Grenze, oder geben Sie deren x-Wert ein.

4. Drücken Sie [ENTER]. Die Marke ▸ am oberen Bildschirmrand kennzeichnet die untere Grenze.

   Hinweis: Zum Entfernen der schraffierten Fläche drücken Sie [F4] **(ReGraph)**.

5. Definieren Sie die obere Grenze, und drücken Sie [ENTER].

   Das Intervall wird schraffiert und dessen numerisches Integral angezeigt.

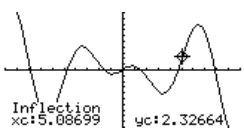


Einen Wendepunkt in einem Intervall ermitteln

1. Drücken Sie im Graphikbildschirm [F5], und wählen Sie **8:Inflection**.
2. Verwenden Sie ggf. ▼ und ▲, um die gewünschte Funktion zu wählen.
3. Definieren Sie die untere Grenze für x. Setzen Sie den Cursor mit ◀ und ▶ an die untere Grenze, oder geben Sie deren x-Wert ein.
4. Drücken Sie [ENTER]. Die Marke ▸ am oberen Bildschirmrand kennzeichnet die untere Grenze.

5. Definieren Sie die obere Grenze, und drücken Sie [ENTER].

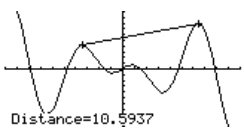
   Der Cursor wird auf den Wendepunkt (falls vorhanden) im Intervall gesetzt, und dessen Koordinaten werden angezeigt.



Die Entfernung zwischen zwei Punkten ermitteln

1. Drücken Sie im Graphikbildschirm F5, und wählen Sie **9:Distance**.
2. Verwenden Sie ggf. ▼ und ▲, um die Funktion für den ersten Punkt zu wählen.
3. Definieren Sie den ersten Punkt. Setzen Sie den Cursor mit ◀ oder ▶ auf den Punkt, oder geben Sie dessen *x*-Wert ein.
4. Drücken Sie ENTER. Das Zeichen **+** markiert den Punkt.
5. Befindet sich der zweite Punkt auf einer anderen Funktion, wählen Sie diese mit ▼ und ▲.
6. Definieren Sie den zweiten Punkt. (Verwenden Sie zum Definieren des Punkts den Cursor, wird durch die Bewegung des Cursors eine Linie gezogen.)
7. Drücken Sie ENTER.

   Die Entfernung der beiden Punkte zueinander wird mit der Verbindungslinie angezeigt.



Eine Tangente zeichnen

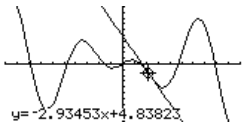
1. Drücken Sie im Graphikbildschirm F5, und wählen Sie **A:Tangent**.

2. Verwenden Sie ggf. ▼ und ▲, um die gewünschte Funktion zu wählen.

   Hinweis: Zum Entfernen einer gezeichneten Tangente drücken Sie F4 **(ReGraph)**.

3. Definieren Sie den Berührpunkt. Setzen Sie den Cursor auf den Punkt, oder geben Sie dessen x-Wert ein.
4. Drücken Sie ENTER.

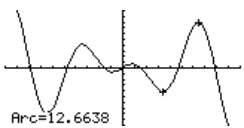
   Die Tangente wird gezeichnet und ihre Gleichung angezeigt.



Eine Bogenlänge ermitteln

1. Drücken Sie im Graphikbildschirm F5, und wählen Sie **B:Arc**.
2. Verwenden Sie ggf. ▼ und ▲, um die gewünschte Funktion zu wählen.
3. Definieren Sie den ersten Punkt des Bogens. Bewegen Sie den Cursor mit ◀ oder ▶, oder geben Sie den x-Wert ein.
4. Drücken Sie ENTER. Das Zeichen **+** markiert den ersten Punkt.
5. Definieren Sie den zweiten Punkt, und drücken Sie ENTER.

   Das Zeichen **+** markiert den zweiten Punkt, und die Bogenlänge wird angezeigt.



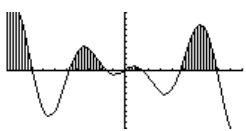
Die Fläche zwischen einer Funktion und der X-Achse schraffieren

Es darf nur eine graphisch dargestellte Funktion vorliegen. Bei zwei oder mehr Graphen schraffiert das Shade-Werkzeug die Fläche zwischen zwei Funktionen.

1. Drücken Sie im Graphikbildschirm [F5], und wählen Sie **C:Shade**. Sie werden gefragt, ob die Fläche über der x-Achse schraffiert werden soll: **Above X axis?**
2. Wählen Sie eine der folgenden Optionen. Zum Schraffieren der Funktionsfläche:
   - oberhalb der x-Achse drücken Sie [ENTER].
   - Unterhalb der x-Achse drücken Sie:
     [alpha] **N**
     N
3. Definieren Sie die untere Grenze für x. Setzen Sie den Cursor mit ◀ und ▶ an die untere Grenze, oder geben Sie deren x-Wert ein.

   Hinweis: Wenn Sie bei der Bestimmung der unteren und oberen Grenze weder ◀ oder ▶ drücken noch einen *x*-Wert eingeben, werden **xmin** und **xmax** als untere bzw. obere Grenze verwendet.
4. Drücken Sie [ENTER]. Die Marke ▸ am oberen Bildschirmrand kennzeichnet die untere Grenze.
5. Definieren Sie die obere Grenze, und drücken Sie [ENTER].

   Die eingeschlossene Fläche wird schraffiert. Zum Entfernen der schraffierten Fläche drücken Sie [F4] **(ReGraph)**.



Die Fläche zwischen zwei Funktionen in einem Intervall schraffieren

Es müssen mindestens zwei graphisch dargestellte Funktionen vorliegen. Bei nur einem Graphen schraffiert das Shade-Werkzeug die Fläche zwischen der Funktion und der x-Achse.

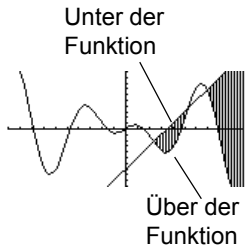
1. Drücken Sie im Graphikbildschirm F5, und wählen Sie **C:Shade**. Sie werden gefragt, ob die Fläche oberhalb der Funktion schraffiert werden soll: **Above?**
2. Verwenden Sie ggf. ▼ oder ▲, um eine Funktion zu wählen. (Die Fläche oberhalb dieser Funktion wird schraffiert.)
3. Drücken Sie ENTER. Der Cursor geht auf den nächsten Graphen über, und Sie werden gefragt, ob die Fläche unterhalb der Funktion schraffiert werden soll: **Below?**
4. Verwenden Sie ggf. ▼ oder ▲, um eine andere Funktion zu wählen. (Die Fläche unterhalb dieser Funktion wird schraffiert.)
5. Drücken Sie ENTER.
6. Definieren Sie die untere Grenze für x. Setzen Sie den Cursor mit ◀ und ▶ an die untere Grenze, oder geben Sie deren *x*-Wert ein.

   Hinweis: Wenn Sie bei der Bestimmung der unteren und oberen Grenze weder ◀ oder ▶ drücken noch einen x-Wert eingeben, werden **xmin** und **xmax** als untere bzw. obere Grenze verwendet.

7. Drücken Sie [ENTER]. Die Marke ▸ am oberen Bildschirmrand kennzeichnet die untere Grenze.

8. Definieren Sie die obere Grenze, und drücken Sie [ENTER].

   Die eingeschlossene Fläche wird schraffiert. Zum Entfernen der schraffierten Fläche drücken Sie [F4] **(ReGraph)**.



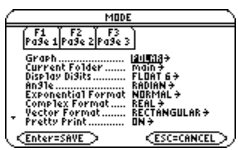
Polardarstellung

Schritte zur Erzeugung von Polardarstellungen

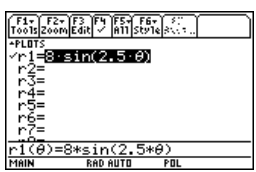
Die Grundschritte für Polardarstellungen sind dieselben, wie für **y(x)**-Funktionen *Grundlagen der graphischen Darstellung von Funktionen* erläutert. Für Polardarstellungen geltende Abweichungen werden im folgenden erläutert.

Polardarstellungen erzeugen

1. **Graph**-Modus (MODE) auf **POLAR** einstellen. Stellen Sie ggf. auch den **Angle**-Modus ein.



2. Definition im Y= Editor vornehmen (◆ [Y=]).
3. Auswahl der (F4) definierten Darstellungen.



   Hinweis: Zum Ausschalten etwaiger Statistik-Plots drücken Sie F5 **5**, oder verwenden Sie F4, um die Auswahl aufzuheben.

4. Anzeigeart einstellen.

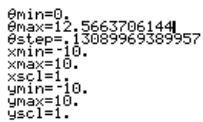
   2nd [F6]

   F6

   Dieser Schritt ist optional. Mehrere Darstellungen können hierdurch optisch besser unterschieden werden.

5. Ansichtfenster definieren (◆ [WINDOW]).

   F2 **Zoom** ändert auch das Ansichtfenster.



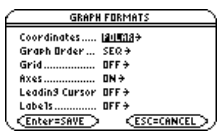
6. Das Graphikformat ggf. ändern.

   F1 **9**

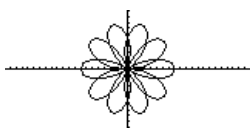
   – oder –

   ◆ I

   ◆ **F**



7. Die Darstellung erzeugen (◆ [GRAPH]).



Den Graphen untersuchen

Im Graphikbildschirm können Sie:

- Mit dem frei beweglichen Cursor die Koordinaten eines beliebigen Pixels und durch Tracen einer Polardarstellung die Koordinaten eines geplotteten Punkts anzeigen.

- Mit dem Menüleisten-Menü F2 **Zoom** einen Abschnitt der Darstellung verkleinern oder vergrößern.
- Mit dem Menüleisten-Menü F5 **Math** Ableitungen, Tangenten etc. ermitteln. Für Polardarstellungen sind einige Menüpunkte nicht verfügbar.

Unterschiede zwischen den Einstellungen Polar und Funktion im Graphik-Modus

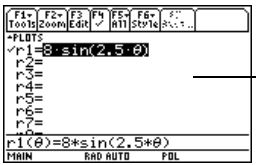
In dieser Einheit wird vorausgesetzt, dass Sie bereits wissen, wie **y(x)**-Funktionen graphisch dargestellt werden (siehe ggf. *Grundlagen der graphischen Darstellung von Funktionen*). Im vorliegenden Abschnitt werden die für Polardarstellungen geltenden Unterschiede beschrieben.

Den Graphikmodus einstellen

Verwenden Sie MODE um **Graph = POLAR** einzustellen, ehe Sie Gleichungen definieren oder Fenstervariable festlegen. Im Y= Editor und im Window-Editor vorgenommene Eingaben gelten nur für die aktuelle **Graph**-Moduseinstellung.

Sie sollten außerdem den **Angle**-Modus auf die Einheit (RADIAN oder DEGREE) einstellen, die Sie für θ verwenden möchten.

Polardarstellungen im Y= Editor definieren



Sie können Polargleichungen für r1(θ) bis r99(θ) definieren.

Sie können zur Definition von Funktionen und Gleichungen für einen beliebigen Graphikmodus, unabhängig vom aktuellen Modus, den Befehl **Define** im Hauptbildschirm verwenden (siehe Modul *Technische Referenz*).

Der Y= Editor führt für jede **Graphik**-Moduseinstellung eine unabhängige Funktionsliste. Beispiel:

- Sie definieren im Graphikmodus FUNCTION einen Satz **y(x)**-Funktionen. Sie gehen über zum Graphikmodus POLAR und erstellen einen Satz **r(**θ**)**-Definitionen.
- Wenn Sie zum Graphikmodus FUNCTION zurückkehren, sind Ihre **y(x)**-Funktionen im Y= Editor nach wie vor definiert. Wenn Sie zum Graphikmodus POLAR zurückkehren, sind Ihre **r(**θ**)**-Definitionen weiterhin vorhanden.

Den Anzeigestil wählen

Die Anzeigearten **Above** und **Below** sind für Polardarstellungen nicht verfügbar und werden deshalb im Menüleisten-Menü **Style** des Y= Editors unscharf angezeigt.

Fenstervariable

Der Window-Editor führt für jede **Graphik**-Moduseinstellung einen eigenen Satz Fenstervariable (so wie der Y= Editor unabhängige Funktionslisten unterhält). Polardarstellungen verwenden folgende Fenstervariablen.

Variable	Beschreibung
θ**min**, θ**max**	Kleinster und größter anzuzeigender θ-Wert.
θ**step**	Schrittweite für den θ-Wert. Polardarstellungsdefinitionen werden ausgewertet bei: r(θmin) r(θmin+θstep) r(θmin+2(θstep)) ... bis nicht größer als ... r(θmax)
xmin, **xmax**, **ymin**, **ymax**	Grenzen des Ansichtfensters.
xscl, **yscl**	Abstand zwischen den Einheiten auf der x- und y-Achse.

Hinweis: Sie können einen negativen θ**step** verwenden. In diesem Fall muß θ**min** größer als θ**max** sein.

Die Standardwerte (eingestellt, wenn Sie **6:ZoomStd** aus dem Menüleisten-Menü F2 **Zoom** wählen) sind:

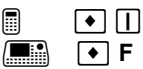
θmin = 0.		xmin = ⁻10.	ymin = ⁻10.
θmax = 2π	(6.2831853... radians oder 360 Grad)	xmax = 10.	ymax = 10.

θstep = π/24	(.1308996... radians oder 7.5 Grad)	xscl = 1.	yscl = 1.

Um sicherzustellen, dass genügend Punkte geplottet werden, müssen Sie ggf. die Standardwerte für die θ-Variablen (θ**min**, θ**max**, θ**step**) ändern.

Das Graphikformat einstellen

Sollen die Koordinaten als **r**- und θ-Werte angezeigt werden, nehmen Sie mit

F1 **9**
– oder –
 **F**

die Einstellung **Coordinates = POLAR** vor. Bei der Einstellung **Coordinates = RECT** werden die Polardarstellungen graphisch korrekt dargestellt, die Koordinaten werden aber als *x* - und *y* -Koordinaten angezeigt.

Wenn Sie eine Polardarstellung tracen, wird selbst bei der Einstellung **Coordinates = RECT** die θ-Koordinate gezeigt.

Einen Graphen untersuchen

Wie beim graphischen Darstellen von Funktionen können Sie auch hier mit den folgenden Tools einen Graphen untersuchen. Die Koordinaten werden gemäß der Einstellung das Graphikformats entweder polar oder kartesisch angezeigt.

Tool	Bei Polargraphen:
Frei beweglicher Cursor	Genau wie für Funktionsgraphen.
F2 **Zoom**	Genau wie für Funktionsgraphen. Nur **x**- (**xmin**, **xmax**, **xscl**) und **y**- (**ymin**, **ymax**, **yscl**) Window-Variablen sind betroffen. Die θ-Window-Variablen (θ**min**, θ**max**, θ**step**) werden erst dann beeinflußt, wenn Sie **6:ZoomStd** wählen (dadurch wird θ**min = 0**, θ**max = 2**π und θ**step =** π**/24** eingestellt).
F3 **Trace**	Dient zum Bewegen des Cursors entlang eines Graphen in Schritten von jeweils einem θ**step**. • Wenn Sie einen Trace-Vorgang beginnen, befindet sich der Cursor auf der ersten gewählten Gleichung bei θ**min**. • QuickCenter gilt für alle Richtungen. Wenn Sie den Cursor über den Bildschirmrand hinaus (oben, unten, links oder rechts) bewegen, drücken Sie ENTER, um das Ansichtfenster bezüglich der Cursorposition zu zentrieren. • Der automatische Schwenk ist nicht verfügbar. Bewegen Sie den Cursor über den linken oder rechten Bildschirmrand hinaus, führt der TI-89 / Voyage™ 200 keinen automatischen Schwenk des Ansichtfensters durch. Sie können jedoch QuickCenter verwenden.

Tool	Bei Polargraphen:
F5 **Math**	Nur **1:Value**, **6:Derivatives**, **9:Distance**, **A:Tangent** und **B:Arc** sind für Polardarstellungen verfügbar. Diese Tools basieren auf θ-Werten. Beispiel: • **1:Value** zeigt für einen bestimmten θ-Wert einen **r**-Wert an (oder x und y, je nach Graphikformat). • **6:Derivatives** ermittelt an einem für einen bestimmten θ-Wert definierten Punkt **dy/dx** oder **dr/d**θ.

Sie können **r(**θ**)** auch während eines **Trace**-Vorgangs auswerten, indem Sie den θ-Wert eingeben und ENTER drücken.

Hinweis: Während eines Trace-Vorgangs können Sie jederzeit QuickCenter verwenden, auch wenn sich der Cursor noch im Bildschirm befindet.

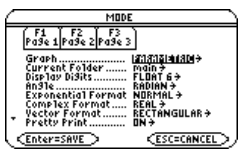
Parameterdarstellung von Kurven

Schritte zur Parameterdarstellung einer Kurve

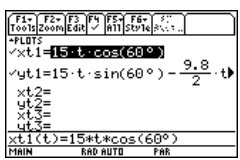
Die Grundschritte für die Parameterdarstellung von Kurven sind dieselben, wie für **y(x)**-Funktionen (in "Grundlagen der graphischen Darstellung von Funktionen", erläutert). Für die Parameterdarstellung geltende Abweichungen werden im folgenden erläutert.

Parameterdarstellung einer Kurve

1. **Graph**-Modus (MODE) auf **PARAMETRIC** setzen. Stellen Sie ggf. auch den **Angle**-Modus ein.



2. x- und y- Komponente im Y= Editor (◆ [Y=]) definieren.
3. Die definierten Komponenten auswählen (F4): x- oder y-Komponente oder beide.



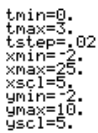
   Hinweis: Um Stat-Daten-Plots auszuschalten, drücken Sie F5 **5**, oder heben Sie ihre Auswahl mit F4 auf.

4. Anzeigestil einstellen; entweder für x- oder für y-Komponente.

   [2nd] [F6]
   [F6]

   Dieser Schritt ist optional. Mehrere Gleichungen können hierdurch optisch besser unterschieden werden.

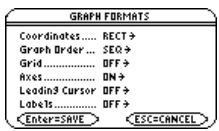
5. Ansichtfenster definieren ([◆] [WINDOW]).



   [F2] **Zoom** ändert auch das Ansichtfenster.

6. Das Graphikformat ggf. ändern.

   [F1] **9**
   – oder –
   [◆] [|]
   [◆] **F**



7. Die ausgewählten Komponenten graphisch darstellen ([◆] [GRAPH]).

Den Graphen untersuchen

Im Graphikbildschirm können Sie:

- Mit dem frei beweglichen Cursor die Koordinaten eines beliebigen Pixels anzeigen und durch Tracen einer Kurve die eines geplotteten Punkts.

- Mit dem Menüleisten-Menü F2 **Zoom** einen Abschnitt des Graphen verkleinern oder vergrößern.
- Mit dem Menüleisten-Menü F5 **Math** Ableitungen, Tangenten etc. ermitteln. Einige Menüpunkte sind für Parameterdarstellungen nicht verfügbar.

Unterschiede zwischen den Einstellungen Parametrisch und Funktion im Graphik-Modus

In diesem Einheit wird vorausgesetzt, daß Sie bereits wissen, wie **y(x)**-Funktionen graphisch dargestellt werden (siehe ggf. *Grundlagen der graphischen Darstellung von Funktionen*). Im vorliegenden Abschnitt werden die für Parameterdarstellungen geltenden Unterschiede beschrieben.

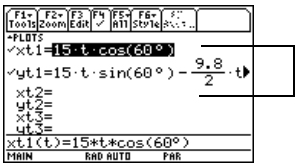
Den Graphikmodus einstellen

Verwenden Sie MODE um **Graph = PARAMETRIC** einzustellen, ehe Sie Gleichungen definieren oder Fenstervariable festlegen. Im Y= Editor und im Window-Editor vorgenommene Eingaben gelten nur für die *aktuelle* **Graphik**-Moduseinstellung.

Komponenten im Y= Editor definieren

Für die Parameterdarstellung einer Kurve müssen Sie sowohl deren x- als auch deren y-Komponente definieren. Bei Definition nur einer Komponente kann die Funktion nicht

gezeichnet werden. (Sie können mit Einzelkomponenten aber, wie in *Tabellen* beschrieben, eine automatische Tabelle erstellen.)



Geben Sie die x- und y-Komponenten in getrennte Zeilen ein.

Sie können xt1(t) bis xt99(t) und yt1(t) bis yt99(t) definieren.

Achten Sie auf implizite Multiplikation im Zusammenhang mit **t**. Zum Beispiel:

Richtige Eingabe:	**Falsche Eingabe:**	**Begründung:**
t∗cos(60)	tcos(60)	**tcos** wird nicht als implizite Multiplikation, sondern als eine benutzerdefinierte Funktion namens **tcos** interpretiert. Meistens ist eine solche Funktion aber gar nicht vorhanden. Deshalb gibt der TI-89 Titanium / Voyage™ 200 Graphischer Rechner anstelle einer Zahl lediglich den Funktionsnamen zurück.

Hinweis: Überprüfen Sie bei Verwendung von **t**, ob eine implizite Multiplikation in der gegebenen Situation korrekt ist. Sie können zum Definieren von Funktionen und Gleichungen für einen beliebigen Graphikmodus, unabhängig vom aktuellen Modus, den Befehl **Define** im Hauptbildschirm verwenden (siehe Modul *Technische Referenz*).

Der Y= Editor führt für jede **Graphik**-Moduseinstellung eine unabhängige Funktionsliste. Beispiel:

- Sie definieren im Graphikmodus FUNCTION einen Satz **y(x)**-Funktionen. Sie gehen über zum Graphikmodus PARAMETRIC und definieren einen Satz x- und y-Komponenten.
- Wenn Sie zum Graphikmodus FUNCTION zurückkehren, sind Ihre **y(x)**-Funktionen im Y= Editor nach wie vor definiert. Wenn Sie zum Graphikmodus PARAMETRIC zurückkehren, sind Ihre x- und y-Komponenten ebenfalls weiterhin definiert.

Parametrische Gleichungen auswählen

Wählen Sie die x- oder die y-Komponente oder beide Komponenten aus, um die Parameterdarstellung einer Kurve zu zeichnen. Beim Eingeben oder Ändern einer Komponente wird diese automatisch ausgewählt.

Es kann für Tabellen (siehe *Tabellen*) nützlich sein, die x- und die y-Komponenten getrennt zu wählen. Bei mehreren Kurven können Sie sämtliche x-Komponenten oder sämtliche y-Komponenten wählen und miteinander vergleichen.

Den Zeichenstil wählen

Sie können den Zeichenstil wahlweise für die x- oder die y-Komponente einstellen. Wenn Sie beispielsweise für die x-Komponente **Dot** einstellen, wendet der TI-89 Titanium / Voyage™ 200 **Dot** automatisch auch auf die y-Komponente an.

Hinweis: Verwenden Sie die Stile **Animate** und **Path**, um interessante Effekte für die Objektbewegung zu erhalten.

Die Stile **Above** und **Below** sind für Parameterdarstellungen nicht verfügbar und werden deshalb im Menüleisten-Menü **Style** des Y= Editors unscharf angezeigt.

Fenstervariable

Der Window-Editor führt für jede **Graph**-Moduseinstellung einen eigenen Satz Fenstervariablen (wie auch der Y= Editor unabhängige Funktionslisten enthält). Für Parameterdarstellungen werden folgende Window-Variablen benutzt.

Hinweis: Sie können einen negativen **tstep** verwenden. In diesem Fall muß **tmin** größer als **tmax** sein.

Variable	Beschreibung
tmin, **tmax**	Kleinster und größter auszuwertender **t**-Wert.
tstep	Schrittweite für den **t**-Wert. Die Komponenten für die Parameterdarstellung einer Kurve werden berechnet bei: x(tmin) y(tmin) x(tmin+tstep) y(tmin+tstep) x(tmin+2(tstep)) y(tmin+2(tstep)) ... bis nicht größer als ... ... bis nicht größer als ... x(tmax) y(tmax)
xmin, **xmax**, **ymin**, **ymax**	Grenzen des Ansichtfensters.
xscl, **yscl**	Abstand zwischen den Einheiten auf der x- und y-Achse.

Standardwerte (eingestellt, wenn Sie **6:ZoomStd** aus dem Menüleisten-Menü F2 **Zoom** wählen) sind:

tmin = 0	xmin = ⁻10.	ymin = ⁻10.

tmax = 2π	(6.2831853... rad oder 360 Grad)	xmax = 10.	ymax = 10.
tstep =π/24	(.1308996... radians or 7.5 degrees)	xscl = 1.	yscl = 1.

Um sicherzustellen, daß genügend Punkte geplottet werden, müssen Sie ggf. die Standardwerte für die **t**-Variablen (**tmin**, **tmax**, **tstep**) ändern.

Einen Graphen untersuchen

Wie beim graphischen Darstellen von Funktionen können Sie mit den folgenden Tools auch die Parameterdarstellung einer Kurve untersuchen.

Hinweis: Sie können **x(t)** und **y(t)** auch während eines Trace-Vorgangs berechnen, indem Sie den **t**-Wert eingeben und ENTER drücken. Während eines Trace-Vorgangs können Sie jederzeit QuickCenter verwenden, auch wenn sich der Cursor noch im Bildschirm befindet.

Tool	Bei Parameterdarstellung einer Kurve:
Frei beweglicher Cursor	Genau wie für Funktionsgraphen.
F2 **Zoom**	Abgesehen von folgenden Ausnahmen, genau wie für Funktionsgraphen: • Nur **x**- (**xmin**, **xmax**, **xscl**) und **y**- (**ymin**, **ymax**, **yscl**) Window-Variablen sind betroffen. • Die **t**-Window-Variablen (**tmin**, **tmax**, **tstep**) werden erst dann beeinflußt, wenn Sie **6:ZoomStd** wählen (dadurch werden die Einstellungen **tmin = 0**, **tmax = 2π** und **tstep = π/24** vorgenommen).

Tool	Bei Parameterdarstellung einer Kurve:
F3 **Trace**	Der Cursor wird in Schritten von jeweils einem **tstep** im Graphen entlangbewegt. • Wenn Sie einen Trace-Vorgang beginnen, befindet sich der Cursor auf der ersten gewählten Parameterdarstellung bei **tmin**. • QuickCenter gilt für alle Richtungen. Wenn Sie den Cursor über den Bildschirmrand hinaus (oben, unten, links oder rechts) bewegen, drücken Sie ENTER, um das Ansichtfenster bezüglich der Cursorposition zu zentrieren. • Der automatische Schwenk ist nicht verfügbar. Bewegen Sie den Cursor über den linken oder rechten Bildschirmrand hinaus, führt der keinen automatischen Schwenk des Ansichtfensters durch. Sie können jedoch QuickCenter verwenden.
F5 **Math**	Nur **1:Value**, **6:Derivatives**, **9:Distance**, **A:Tangent** und **B:Arc** sind für Parameterdarstellungen verfügbar. Diese Tools basieren auf **t**-Werten. Beispiel: • **1:Value** zeigt für einen bestimmten **t**-Wert x- und y-Wert an. • **6:Derivatives** ermittelt an einem für einen bestimmten **t**-Wert definierten Punkt **dy/dx**, **dy/dt** oder **dx/dt**.

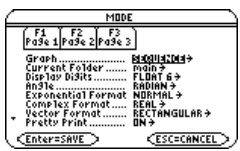
Graphische Darstellung von Folgen

Schritte zur graphischen Darstellung von Folgen

Die Grundschritte für die graphische Darstellung von Folgen sind dieselben, wie für **y(x)**-Funktionen in *Grundlagen der graphischen Darstellung von Funktionen* erläutert. Für Folgen geltende Abweichungen werden nachfolgend erläutert.

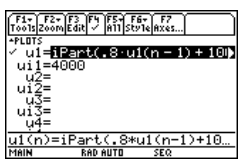
Folgen graphisch darstellen

1. Graph-Modus ([MODE]) auf **SEQUENCE** einstellen. Stellen Sie ggf. auch den **Angle**-Modus ein.



2. Folgen und bei Bedarf Anfangswerte im Y= Editor ([◆] [Y=]) definieren.
3. Die darzustellenden Folgen auswählen ([F4]). Wählen Sie keine Anfangswerte.

   Hinweis: Zum Ausschalten etwaiger Statistik-Plots drücken Sie [F5] **5**, oder verwenden Sie [F4], um die Auswahl aufzuheben.



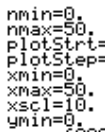
4. Anzeigeart für eine Folge einstellen.

   [2nd] [F6]
   [F6]

   Bei Folgen ist die Voreinstellung **Square**.

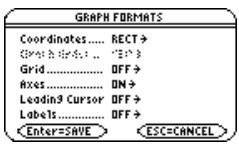
5. Das Ansichtfenster definieren ([♦] [WINDOW]).

   [F2] **Zoom** beeinlußt auch das Anzeigefenster.

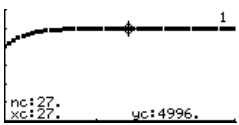


6. Das Graphikformat ggf. ändern.

   [F1] **9**
   — oder —
   [♦] [|]
   [♦] **F**



7. Die ausgewählten Folgen graphisch darstellen ([♦] [GRAPH]).



Den Graphen untersuchen

Im Graphikbildschirm können Sie:

- Mit dem frei beweglichen Cursor die Koordinaten eines beliebigen Pixels und, durch Tracen einer Folge, die eines geplotteten Punkts anzeigen.

- Mit dem Menüleisten-Menü F2 **Zoom** einen Abschnitt der Darstellung verkleinern oder vergrößern.
- Mit dem Menüleisten-Menü F5 **Math** eine Folge berechnen. Bei Folgen ist nur **1:Value** verfügbar.
- Folgen auf **Zeit**- (Standard), **Netz**- oder benutzerspezifische **Achsen** auftragen.

Hinweis: Sie können eine Folge auch während des Trace-Vorgangs berechnen. Geben Sie den n-Wert einfach direkt über die Tastatur ein.

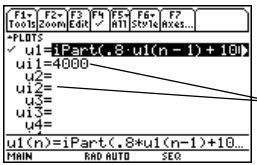
Unterschiede zwischen den Einstellungen Folge und Funktion im Graphik-Modus

In dieser Einheit wird vorausgesetzt, daß Sie bereits wissen, wie **y(x)**-Funktionen graphisch dargestellt werden (siehe ggf. *Grundlagen der graphischen Darstellung von Funktionen*). Im vorliegenden Abschnitt werden die für Folgen geltenden Unterschiede beschrieben.

Den Graphikmodus einstellen

Verwenden Sie MODE um **Graph = SEQUENCE** einzustellen, ehe Sie Gleichungen definieren oder Fenstervariable festlegen. Im Y= Editor und im Window-Editor vorgenommene Eingaben gelten nur für die aktuelle Graph-Moduseinstellung.

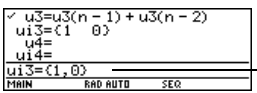
Folgen im Y= Editor definieren



Sie können Folgen von u1(n) bis u99(n) definieren.

Verwenden Sie ui nur für rekursive Folgen, für die ein oder mehrere Anfangswerte erforderlich sind.

Sind für eine Folge mehrere Anfangswerte erforderlich, geben Sie diese als Liste in Klammern { } und durch Kommata voneinander getrennt ein. Zum Eingeben mehrerer Anfangswerte müssen Sie eine Liste verwenden.



Geben Sie {1,0} ein. In der Folgenliste wird {1 0} angezeigt.

Wenn für eine Folge ein Anfangswert erforderlich ist, Sie aber keinen eingeben, wird bei der graphischen Darstellung ein Fehler gemeldet.

Im Y= Editor können Sie mit Axes die Achsen für die graphische Darstellung der Folgen wählen. Bei Folgen (und nur dort) besteht die Möglichkeit, verschiedene Achsen für den Graphen zu wählen. TIME ist die Standardachse.

Achsen	Beschreibung
TIME	Trägt n auf der x-Achse und **u(n)** auf der y-Achse auf.
WEB	Trägt **u(n−1)** auf der x-Achse und **u(n)** auf der y-Achse auf.
CUSTOM	Dient zum Wählen der x- und y-Achse.

Der Y= Editor führt für jede Graph-Moduseinstellung eine unabhängige Funktionsliste. Beispiel:

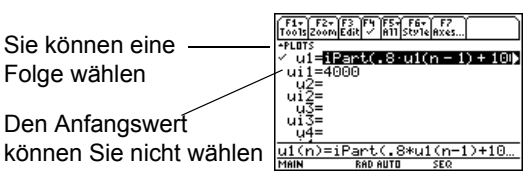
- Sie definieren im Graphikmodus FUNCTION einen Satz **y(x)**-Funktionen. Sie gehen über zum Graphikmodus SEQUENCE und definieren einen Satz **u(n)**-Folgen.
- Wenn Sie zum Graphikmodus FUNCTION zurückkehren, sind Ihre **y(x)**-Funktionen im Y= Editor weiterhin definiert. Wenn Sie zum Graphikmodus SEQUENCE zurückkehren, sind Ihre **u(n)**-Folgen weiterhin definiert.

Hinweis: Sie können zum Definieren von Funktionen und Gleichungen für einen beliebigen Graphikmodus, unabhängig vom aktuellen Modus, den Befehl **Define** im Hauptbildschirm verwenden (siehe Modul *Technische Referenz*).

Folgen auswählen

Bei TIME- (Zeit-) und WEB- (Netz-) Achsen stellt der TI-89 Titanium / Voyage™ 200 nur die ausgewählten Folgen graphisch dar. Haben Sie Folgen eingegeben, für die ein Anfangswert erforderlich ist, müssen Sie auch den entsprechenden ui-Wert eingeben.

Hinweis: Bei TIME- und CUSTOM- (benutzerspezifischen) Achsen werden sämtliche definierten Folgen berechnet, auch wenn sie nicht geplottet werden.



Bei CUSTOM-Achsen wird eine in den benutzerspezifischen Einstellungen angegebene Folge auch dann graphisch dargestellt, wenn sie nicht ausgewählt wurde.

Den Zeichenstil wählen

Für Folgen sind nur die Arten **Line**, **Dot**, **Square** und **Thick** verfügbar. **Dot** und **Square** werden ausschließlich an den diskreten ganzzahligen Werten (in plotstep-Inkrementen) gesetzt, an denen eine Folge abgebildet wird.

Fenstervariable

Der Window-Editor führt für jede Graphik-Moduseinstellung einen eigenen Satz Fenstervariable (so wie der Y= Editor unabhängige Funktionslisten unterhält). Für die Darstellung von Folgen werden folgende Fenstervariablen benutzt.

Variable	Beschreibung
nmin, nmax	Kleinster und größter zu berechnender n-Wert. Folgen werden berechnet bei: u(nmin) u(nmin+1) u(nmin+2) ... bis nicht weiter als ... u(nmax)
plotStrt	Die Nummer des Glieds, das als erstes geplottet wird (je nach plotstep). Soll das Plotten beispielsweise mit dem zweiten Glied der Folge beginnen, stellen Sie **plotstrt = 2** ein. Das erste Glied wird zwar in **nmin** berechnet, aber nicht geplottet.

Variable	Beschreibung
plotStep	Inkrementeller n-Wert nur für die graphische Darstellung. Hat keinen Einfluß auf die Berechnung der Folge, bestimmt nur, welche Punkte geplottet werden. Es sei beispielsweise **plotstep = 2** eingestellt. Die Folge wird dann zwar bei jeder ganzen Zahl berechnet, aber nur in jeder zweiten ganzen Zahl geplottet.
xmin, **xmax**, **ymin**, **ymax**	Grenzen des Ansichtfensters.
xscl, **yscl**	Abstand zwischen den Einheiten auf der x- und y-Achse.

Hinweis: Sowohl **nmin** als auch **nmax** müssen positive ganze Zahlen sein; **nmin** hingegen kann auch Null sein. **nmin**, **nmax**, **plotstrt** und **plotstep** müssen ganze Zahlen ≥ 1 sein. Geben Sie keine ganzen Zahlen ein, wird Ihre Eingabe stets auf eine ganze Zahl gerundet.

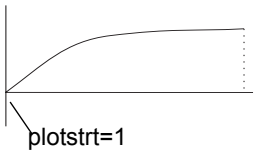
Standardwerte (eingestellt, wenn Sie **6:ZoomStd** aus dem Symbolleistenmenü F2 **Zoom** wählen) sind:

nmin = 1	xmin = ⁻10	ymin = ⁻10
nmax = 10	xmax = 10	ymax = 10
plotStrt = 1	xscl = 1	yscl = 1
plotStep = 1		

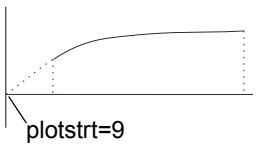
Um sicherzustellen, daß genügend Punkte geplottet werden, müssen Sie ggf. die Standardwerte für die n- und plot-Variable ändern.

In untenstehendem Beispiel für eine rekursive Folge können Sie beobachten, wie ein Graph durch **plotstrt** beeinflußt wird.

Dieser Graph ist vom ersten Glied aus geplottet.



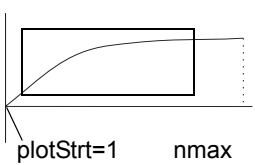
Dieser Graph ist vom neunten Glied aus geplottet.



Hinweis: Bei beiden Graphen werden, abgesehen von **plotstrt**, dieselben Window-Variablen verwendet.

Bei TIME-Achsen (Axes im Y= Editor) können Sie **plotstrt = 1** einstellen und trotzdem nur einen ausgewählten Abschnitt der Folge graphisch darstellen lassen. Definieren Sie einfach ein Ansichtfenster, das nur den gewünschten Bereich der Koordinatenebene anzeigt.

Sie könnten folgende Einstellung vornehmen:



- **xmin** = erster darzustellender n-Wert
- **xmax** = **nmax** (auch andere Werte sind möglich)
- **ymin** und **ymax** = für die Folge erwartete Werte

Das Graphikformat ändern

Das Format Graph Order ist nicht verfügbar.

- Bei TIME- oder CUSTOM-Achsen werden mehrere Folgen stets gleichzeitig geplottet.
- Bei WEB-Achsen werden mehrere Folgen stets nacheinander geplottet.

Einen Graphen untersuchen

Wie beim graphischen Darstellen von Funktionen können Sie auch hier mit den folgenden Tools einen Graphen untersuchen. Die Koordinaten werden gemäß der Einstellung im Graphikformat entweder polar oder kartesisch angezeigt.

Tool	Bei Folgengraphen
Frei beweglicher Cursor	Genau wie für Funktionsgraphen.
F2 **Zoom**	Genau wie für Funktionsgraphen. • Nur **x**- (**xmin**, **xmax**, **xscl**) und **y**- (**ymin**, **ymax**, **yscl**) Window-Variablen sind betroffen. • Die **n**- und plot-Window-Variablen (**nmin**, **nmax**, **plotstrt**, **plotStep**) werden erst dann beeinflußt, wenn Sie **6:ZoomStd** wählen (dadurch werden alle Window-Variablen auf die Standardwerte eingestellt).

Tool	Bei Folgengraphen
F3 **Trace**	Trace funktioniert für TIME-, CUSTOM- und WEB-Achsen sehr unterschiedlich. • Bei TIME- oder CUSTOM-Achsen wird der Cursor in Schritten von jeweils einem plotstep auf der Folge entlangbewegt. Drücken Sie für eine Bewegung um jeweils ca. zehn geplottete Punkte 2nd ▶ oderr 2nd ◀. - Wenn Sie einen Trace-Vorgang beginnen, befindet sich der Cursor auf der ersten gewählten Folge an dem durch plotstrt bestimmten Glied, auch wenn sich dieses außerhalb des Ansichtfensters befindet. - QuickCenter gilt für alle Richtungen. Wenn Sie den Cursor über den Bildschirmrand hinaus (oben, unten, links oder rechts) bewegen, drücken Sie ENTER, um das Ansichtfenster bezüglich der Cursorposition zu zentrieren. • Bei WEB-Achsen verfolgt der Trace-Cursor anstelle der Folge das Netz.
F5 **Math**	Für Folgengraphen ist nur **1:Value** verfügbar. • Bei TIME- und WEB-Achsen wird für einen bestimmten n-Wert der **u(n)**-Wert (dargestellt durch **yc**) angezeigt. • Bei CUSTOM-Achsen hängen die Werte, die x und y entsprechen, von den gewählten Achsen ab.

Sie können eine Folge auch während eines Trace-Vorgangs auswerten, indem Sie einen Wert für n eingeben und ENTER drücken. Während eines Trace-Vorgangs können Sie jederzeit QuickCenter verwenden, auch wenn sich der Cursor noch im Bildschirm befindet.

Achseneinstellung für Zeit-, Netz- oder Eigene-Plots

Nur bei Folgen ist es möglich, für die graphische Darstellung verschiedene Achsenarten zu wählen. Beispiele für diese unterschiedlichen Arten finden Sie gegen Ende der vorliegenden Einheit.

Das Dialogfeld AXES öffnen

In Y= Editor, Option Axes:

- Je nach der aktuellen Axes-Einstellung können einige Punkte unscharf eingeblendet sein.
- Möchten Sie das Dialogfeld verlassen, ohne Änderungen vorzunehmen, drücken Sie [ESC].

Menüpunkt	Beschreibung
Axes	TIME — Trägt **u(n)** auf der y-Achse und n auf der x-Achse auf. WEB — Trägt **u(n)** auf der y-Achse und **u(n−1)** auf der x-Achse auf. CUSTOM — Dient zum Wählen der x- und y-Achse.
Build Web	Nur aktiv bei Axes = WEB; bestimmt, ob ein Netz manuell (TRACE) oder automatisch (AUTO) gezeichnet wird.

Menüpunkt	Beschreibung
X Axis und **Y Axis**	Nur aktiv bei Axes = CUSTOM; hiermit wählen Sie, welcher Wert oder welche Folge auf der x- und der y-Achse geplottet werden soll.

Die Einstellungen können Sie genauso ändern, wie Sie es von anderen Dialogfeldern, z.B. dem Dialogfeld MODE, kennen.

Netz-Plots verwenden

Ein Netz-Plot trägt **u(n)** gegen **u(n−1)** auf, wodurch Sie das Langzeitverhalten einer rekursiven Folge untersuchen können. Die Beispiele in diesem Abschnitt zeigen außerdem, wie der Anfangswert das Verhalten einer Folge beeinflussen kann.

Für Web Plots gültige Funktionen

Eine Folge muß, um auf den WEB-Achsen korrekt dargestellt zu werden, folgenden Bedingungen entsprechen. Die Folge:

- Muß rekursiv sein; aber nur eine Rekursionsebene ist zulässig; **u(n−1)** gilt, **u(n−2)** gilt nicht.
- Darf sich nicht direkt auf n beziehen.
- Darf sich auf keine andere definierte Folge außer auf sich selbst beziehen.

Wenn Sie den Graphikbildschirm öffnen

Nachdem Sie WEB-Achsen gewählt und den Graphikfildschirm geöffnet haben, verhält sich der TI-89 Titanium / Voyage™ 200 folgendermaßen:

- Er zeichnet eine **y=x** -Bezugslinie.
- Er plottet die gewählten Folgendefinitionen als Funktionen mit **u(n−1)** als unabhängiger Variablen. Dadurch wird eine rekursive Folge für die graphische Darstellung in eine nichtrekursive Form umgesetzt.

Gegeben sei beispielsweise die Folge u1(n) = $\sqrt{5-u1(n-1)}$ und der Anfangswert **ui1=1**. Der TI-89 Titanium / Voyage™ 200 zeichnet hier die **y=x** -Bezugslinie und plottet dann y = $y = \sqrt{5-x}$.

Das Netz zeichnen

Nach dem Plotten der Folge kann das Netz, je nach Einstellung von **Build Web** im Dialogfeld AXES, entweder manuell oder automatisch eingeblendet werden.

Build Web =	**Das Netz wird:**
TRACE	Erst dann gezeichnet, wenn Sie F3 drücken. Dann wird das Netz mit der Bewegung des Trace-Cursors Schritt für Schritt aufgebaut (bevor Sie die Trace-Funktion verwenden, muß ein Anfangswert angegeben werden). **Hinweis:** Im Gegensatz zu anderen Graphikmodi können Sie bei WEB-Achsen die Folge nicht tracen.

Build Web =	Das Netz wird:
AUTO	Automatisch gezeichnet. Wenn Sie F3 drücken, können Sie das Netz tracen und dessen Koordinaten anzeigen lassen.

Das Netz:

1. Beginnt auf der x-Achse am Anfangswert ui (wenn **plotstrt = 1**).
2. Bewegt sich in Bezug zur Folge vertikal (auf- oder abwärts).
3. Bewegt sich in Bezug zur **y=x** -Bezugslinie horizontal.
4. Wiederholt diese vertikale und horizontale Bewegung, bis **n=nmax**.

Hinweis: Das Netz startet bei **plotstrt**. Jedesmal, wenn sich das Netz zur Folge bewegt, wird der Wert für n um 1 erhöht (**plotstep** wird nicht berücksichtigt).

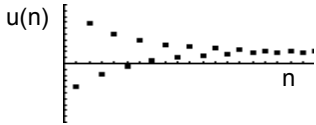
Beispiel: Konvergenz

1. Definieren Sie im Y= Editor (♦ [Y=]) u1(n) = ⁻ .8u1(n−1) + 3.6. Setzen Sie als Anfangswert ui1 = ⁻4.
2. Stellen Sie **Axes = TIME** ein.
3. Stellen Sie im Window-Editor (♦ [WINDOW]) die Fenstervariablen ein.

nmin=1	xmin=0	ymin=⁻10
nmax=25	xmax=25	ymax=10
plotstrt=1	xscl=1	yscl=1
plotstep=1		

4. Stellen Sie die Folge graphisch dar ([♦] [GRAPH]).

   Für Folgen ist **Square** der voreingestellte Anzeigestil.



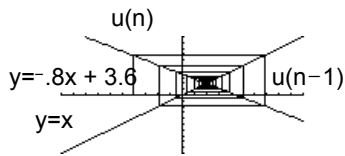
5. Strellen Sie im Y= Editor **Axes = WEB** und **Build Web = AUTO** ein.
6. Ändern Sie im Window-Editor die Fenstervariablen.

nmin=1	xmin=⁻10	ymin=⁻10
nmax=25	xmax=10	ymax=10
plotstrt=1	xscl=1	yscl=1
plotstep=1		

7. Stellen Sie die Folge erneut graphisch dar.

   Unabhängig vom gewählten Anzeigestil werden Web-Plots stets als Linien dargestellt.

   Hinweis: Sie können den Cursor während eines Trace-Vorgangs auf einen bestimmten n-Wert setzen, indem Sie den Wert eingeben und [ENTER] drücken.



8. Drücken Sie [F3]. Mit Betätigung von ⊙ verfolgt der Trace-Cursor das Netz. Auf dem Bildschirm werden die Cursor-Koordinaten nc, xc und yc angezeigt (wobei xc und yc für **u(n−1)** bzw. **u(n)** stehen).

Bei der Bewegung hin zu größeren Werten für nc können Sie die Annäherung von xc und yc an den Konvergenzpunkt beobachten.

Hinweis: Ändert sich der nc-Wert, dann befindet sich der Cursor auf der Folge. Bei der nächsten Betätigung von ▶ bleibt nc gleich, und der Cursor befindet sich auf der **y=x**-Bezugslinie.

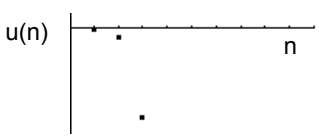
Beispiel: Divergenz

1. Definieren Sie im Y= Editor ([◆] [Y=]) u1(n) = 3.2u1(n−1) − .8(u1(n−1))^2. Bestimmen Sie als Anfangswert ui1 = 4.45.
2. Stellen Sie **Axes = TIME** ein.
3. Stellen Sie im Window-Editor ([◆] [WINDOW]) die Fenstervariablen ein.

nmin=0	xmin=0	ymin=−75
nmax=10	xmax=10	ymax=10
plotstrt=1	xscl=1	yscl=1
plotstep=1		

4. Stellen Sie die Folge graphisch dar ([◆] [GRAPH]).

   Da die Folge rasch zu “großen” negativen Werten divergiert, werden nur wenige Punkte geplottet.



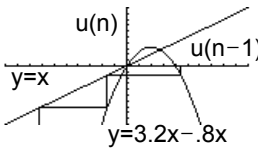
5. Stellen Sie im Y= Editor **Axes = WEB** und **Build Web = AUTO** ein.

6. Ändern Sie im Window-Editor die Fenstervariablen.

nmin=0	xmin=⁻10	ymin=⁻10
nmax=10	xmax=10	ymax=10
plotstrt=1	xscl=1	yscl=1
plotstep=1		

7. Stellen Sie die Folge erneut graphisch dar.

   Der Web-Plot zeigt, wie schnell die Folge zu “großen” negativen Werten divergiert.



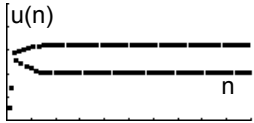
Beispiel: Schwingung

Dieses Beispiel zeigt auf, wie der Anfangswert eine Folge beeinflussen kann.

1. Verwenden Sie im Y= Editor ([♦] [Y=]) die im Beispiel für Divergenz definierte Folge: u1(n) = 3.2u1(n−1) − .8(u1(n−1))^2. Bestimmen Sie für den Anfangswert ui1 = 0.5.
2. Stellen Sie **Axes = TIME** ein.
3. Stellen Sie im Window-Editor ([♦] [WINDOW]) die Fenstervariablen ein.

nmin=1	xmin=0	ymin=0
nmax=100	xmax=100	ymax=5
plotstrt=1	xscl=10	yscl=1
plotstep=1		

4. Stellen Sie die Folge graphisch dar ([♦] [GRAPH]).



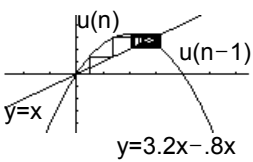
Hinweis: Vergleichen Sie diesen Graphen mit dem Beispiel für Divergenz. Es handelt sich hier um dieselbe Folgenvorschrift mit einem anderen Anfangswert.

5. Stellen Sie im Y= Editor **Axes = WEB** und **Build Web = AUTO** ein.

6. Ändern Sie im Window-Editor die Fenstervariablen.

nmin=1	xmin=2.68	ymin=4.7
nmax=100	xmax=6.47	ymax=47
plotstrt=1	xscl=1	yscl=1
plotstep=1		

7. Stellen Sie die Folge erneut graphisch dar.



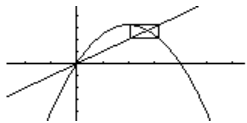
Hinweis: Das Netz oszilliert in einem Orbit zwischen zwei feststehenden Punkten.

8. Drücken Sie F3. Tracen Sie das Netz dann mit ▶.

   Bewegen Sie sich auf größere Werte für nc zu, können Sie die Schwingung von xc und yc zwischen 2.05218 und 3.19782 beobachten.

9. Stellen Sie Window-Editor **plotstrt=50** ein. Stellen Sie dann die Folge erneut graphisch dar.

   Hinweis: Wird der Web-Plot bei einem späteren Glied begonnen, ist die stabile Oszillation klarer erkennbar.



Eigene-Plots verwenden

CUSTOM-Achsen bieten Ihnen bei der graphischen Darstellung von Folgen viel Flexibilität. Aus folgendem Beispiel geht hervor, dass sich die CUSTOM-Achsen besonders zum Zeigen des Zusammenhangs zwischen verschiedenen Folgen eignen.

Beispiel: Räuber-Beute-Modell

Verwenden Sie das Räuber-Beute-Modell der Biologie, um die Anzahl der Kaninchen und Füchse zu ermitteln, die in einem bestimmten Gebiet das Populationsgleichgewicht aufrechterhalten.

R = Anzahl der Kaninchen

M = Zuwachsrate der Kaninchen in Abwesenheit von Füchsen (verwenden Sie .05)

K = Beuterate (verwenden Sie .001)

W = Anzahl der Füchse

G = Zuwachsrate der Füchse in Anwesenheit von Kaninchen (verwenden Sie .0002)

D = Todesrate der Füchse in Abwesenheit von Kaninchen (verwenden Sie .03)

Rn = R_{n-1} (1 + M −K W_{n-1})

Wn = W_{n-1} (1 + G R_{n-1} −D)

1. Definieren Sie im Y= Editor ([♦] [Y=]) die Folge und die Anfangswerte für $\mathbf{R}_n$ und $\mathbf{W}_n$.

   u1(n) = u1(n−1) ∗ (1 + .05 − .001 ∗ u2(n−1))
   ui1 = 200
   u2(n) = u2(n−1) ∗ (1 + .0002 ∗ u1(n−1) − .03)
   ui2 = 50

   Hinweis: Gehen Sie von einer Anfangspopulation von 200 Kaninchen und 50 Füchsen aus.

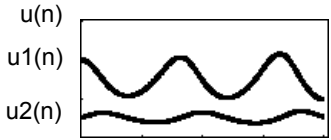
2. Bestimmen Sie für **Axes = TIME**.

3. Stellen Sie im Window-Editor ([♦] [WINDOW]) die Fenstervariablen ein.

nmin=0	xmin=0	ymin=0
nmax=400	xmax=400	ymax=300
plotstrt=1	xscl=100	yscl=100
plotstep=1		

4. Stellen Sie die Folge graphisch dar ([♦] [GRAPH]).

   Hinweis: Verwenden Sie [F3], um die Anzahl der Kaninchen **u1(n)** und der Füchse **u2(n)** über den Zeitraum **(n)** einzeln zu tracen.

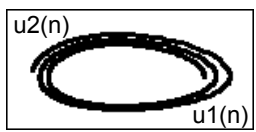


5. Setzen Sie im Y= Editor **Axes = CUSTOM**, **X Axis = u1** und **Y Axis = u2**.
6. Ändern Sie im Window-Editor ([♦] [WINDOW]) die Fenstervariablen.

nmin=0	xmin=84	ymin=25
nmax=400	xmax=237	ymax=75
plotstrt=1	xscl=50	yscl=10
plotstep=1		

7. Stellen Sie die Folge erneut graphisch dar.

   Hinweis: Verwenden Sie [F3], um sowohl die Anzahl der Kaninchen (xc) als auch der Füchse (yc) über 400 Generationen zu tracen.



Eine Tabelle unter Verwendung einer Folge erstellen

In den vorigen Abschnitten wurde beschrieben, wie eine Folge graphisch dargestellt werden kann. Sie können eine Folge auch zur Erstellung einer Tabelle verwenden. Näheres zu Tabellen finden Sie in Tabellen.

Beispiel: Fibonacci-Folge

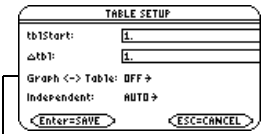
In einer Fibonacci-Folge sind die ersten beiden Glieder 1 und 1. Jedes nachfolgende Glied ist die Summe der beiden unmittelbar vorangehenden Glieder.

1. Definieren Sie im Y= Editor ([♦] [Y=]) die Folge, und stellen Sie wie dargestellt die Anfangswerte ein.

Sie müssen {1,1} eingeben, obwohl in der Folgenliste {1 1} angezeigt wird.

2. Stellen Sie die Tabellenparameter (◆ [TBLSET]) auf:

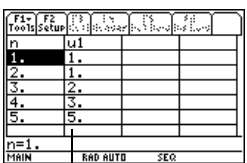
   tblStart = 1
   ∆tbl = 1
   Independent = AUTO



Wenn Sie keine TIME-Achsen verwenden, werden diese Punkte unscharf angezeigt.

3. Stellen Sie die Fenstervariablen (◆ [WINDOW]) so ein, dass **nmin** und **tblStart** denselben Wert aufweisen.

4. Zeigen Sie die Tabelle an (◆ [TABLE]).



Die Fibonacci-Folge befindet sich in Spalte 2.

5. Scrollen Sie abwärts durch die Tabelle (▼ oder [2nd] ▼), um die Folge weiter einsehen zu können.

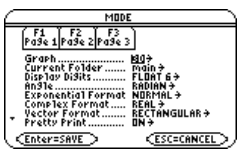
3D-Darstellung

Schritte zur Erzeugung von 3D-Darstellungen

Die Grundschritte für 3D-Darstellungen sind dieselben, wie für **y(x)**-Funktionen (in *Grundlagen der graphischen Darstellung von Funktionen*, erläutert). Für 3D-Darstellungen geltende Abweichungen werden im folgenden erläutert.

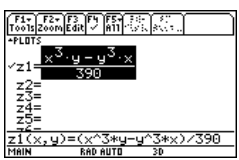
3D-Darstellungen erzeugen

1. **Graph**-Modus (MODE) auf **3D** einstellen. Stellen Sie ggf. auch den **Angle**-Modus ein.



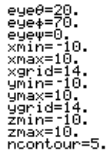
2. 3D-Darstellungen im Y= Editor (♦ Y=). definieren.
3. Darstellung auswählen (F4). Sie können nur eine 3D-Gleichung wählen.

   Zum Ausschalten etwaiger Statistik-Plots drücken Sie F5 **5**, oder verwenden Sie F4, um die Auswahl aufzuheben.



4. Ansichtswürfel definieren (◆ [WINDOW]).

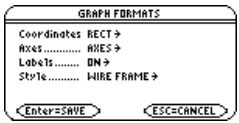
   Bei 3D-Darstellungen heißt das Ansichtfenster “Ansichtswürfel”. F2 **Zoom** ändert auch den Ansichtswürfel.



5. Das Graphikformat ggf. ändern.

   F1 **9**
   – oder –
   ◆ |
   ◆ **F**

   Hinweis: Schalten Sie Axes und Labels ein, um die Ausrichtung des 3D-Graphen besser erkennen zu können.



6. Darstellung erzeugen (◆ [GRAPH]).

   Hinweis: Vor der Anzeige des Graphen wird auf dem Bildschirm der berechnete Anteil in Prozent eingeblendet.

Den Graphen untersuchen

Im Graphikbildschirm können Sie:

- Die Darstellung tracen.
- Mit dem Menüleisten-Menü F2 **Zoom** einen Abschnitt der Darstellung verkleinern oder vergrößern. Einige Menüpunkte sind für 3D-Darstellungen nicht verfügbar und werden deshalb unscharf angezeigt.

- Mit dem Menüleisten-Menü F5 **Math** die Gleichung an einem bestimmten Punkt berechnen. Nur **1:Value** ist für 3D-Darstellungen verfügbar.

Sie können **z(x,y)** auch während des Trace-Vorgangs berechnen. Geben Sie den x-Wert ein, und drücken Sie ENTER; geben Sie dann den y-Wert ein, und drücken Sie ENTER.

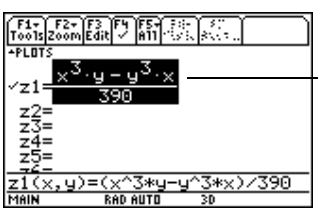
Unterschiede zwischen den Einstellungen 3D und Funktion im Graphik-Modus

In diesem Einheit wird vorausgesetzt, dass Sie bereits wissen, wie **y(x)**-Funktionen graphisch dargestellt werden. Im vorliegenden Abschnitt werden die für 3D-Darstellungen geltenden Unterschiede beschrieben.

Den Graphikmodus einstellen

Verwenden Sie MODE um **Graph = 3D** einzustellen, ehe Sie Gleichungen definieren oder Fenstervariable festlegen. Im Y= Editor und im Window-Editor vorgenommene Eingaben gelten nur für die aktuelle Graphik-Moduseinstellung.

3D-Darstellungen im Y= Editor definieren



Sie können 3D-Darstellungen für **z1(x,y)** bis **z99(x,y)** definieren.

Der Y= Editor führt für jede Graphik-Moduseinstellung eine unabhängige Funktionsliste. Beispiel:

- Sie definieren im Graphikmodus FUNCTION einen Satz **y(x)**-Funktionen. Sie gehen über zum Graphikmodus 3D und erstellen einen Satz **z(x,y)**-Definitionen.
- Wenn Sie zum Graphikmodus FUNCTION zurückkehren, sind Ihre **y(x)**-Funktionen im Y= Editor nach wie vor definiert. Wenn Sie zum Graphikmodus 3D zurückkehren, sind Ihre **z(x,y)**-Definitionen weiterhin vorhanden.

Hinweis: Sie können zum Definieren von Funktionen und Gleichungen für einen beliebigen Graphikmodus, unabhängig vom aktuellen Modus, den Befehl **Define** im Hauptbildschirm verwenden (siehe Modul *Technische Referenz*).

Den Zeichenstil wählen

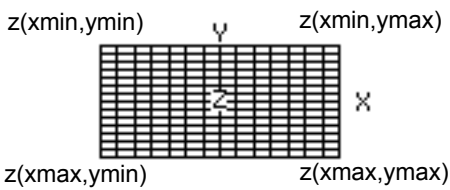
Da Sie nur jeweils eine 3D-Darstellung erzeugen lassen können, sind keine Anzeigestile verfügbar. Das Menüleisten-Menü Style wird im Y= Editor unscharf angezeigt.

Sie können das Style-Format bei 3D-Darstellungen jedoch mit:
F1 **9**
– oder –
◆ I
◆ **F**
auf WIRE FRAME oder HIDDEN SURFACE stellen.

Fenstervariable

Der Window-Editor führt für jede Graph-Moduseinstellung einen eigenen Satz Fenstervariable (so wie der Y= Editor unabhängige Funktionslisten unterhält). 3D-Darstellungen verwenden folgende Fenster-Variablen.

Variable	Beschreibung
eyeθ, **eye**ϕ, **eye**ψ	Winkel (stets in Grad) für die Betrachtung des Graphen.
xmin, **xmax**, **ymin**, **ymax**, **zmin**, **zmax**	Grenzen des Ansichtswürfel.

Variable	Beschreibung
xgrid, **ygrid**	Die Entfernung zwischen **xmin** und **xmax** sowie zwischen **ymin** und **ymax** wird in die angegebene Anzahl Rastereinheiten unterteilt. Die **z(x,y)**-Gleichung wird in jedem Schnittpunkt der Rasterlinien (Gitterlinien) berechnet. Der Inkrement-Wert entlang der x- und der y-Achse wird folgendermaßen berechnet: $\text{x Inkrement} = \frac{\text{xmax} - \text{xmin}}{\text{xgrid}}$ $\text{y Inkrement} = \frac{\text{ymax} - \text{ymin}}{\text{ygrid}}$ Die Anzahl der Gitterlinien ist **xgrid + 1** und **ygrid + 1**. Bei **xgrid = 14** und **ygrid = 14**, besteht das xy-Gitter beispielsweise aus 225 (15 × 15) Gitterpunkten. 
ncontour	Die Anzahl der gleichmäßig entlang des angezeigten Bereichs von z-Werten verteilten Konturlinien.

Hinweis: Geben Sie für **xgrid** oder **ygrid** einen Bruch ein, wird dieser auf die nächste ganze Zahl ≥ 1 gerundet. Im 3D-Modus stehen die Fenstervariablen **scl** nicht zur Verfügung. Es können also keine Teilstriche für die Achsen eingestellt werden.

Standardwerte (eingestellt, wenn Sie **6:ZoomStd** aus dem Menüleisten-Menü F2 **Zoom** wählen) sind:

eyeθ = 20.	xmin = ⁻10.	ymin = ⁻10.	zmin = ⁻10.
eyeϕ = 70.	xmax = 10.	ymax = 10.	zmax = 10.
eyeψ = 0.	xgrid = 14.	ygrid = 14.	ncontour = 5.

Um sicherzustellen, dass genug Punkte geplottet werden, müssen Sie ggf. die Standardwerte für die Gitter-Variablen (**xgrid**, **ygrid**) erhöhen.

Hinweis: Durch Erhöhung der Gitter-Variablen wird die Darstellungsgeschwindigkeit herabgesetzt.

Das Graphikformat einstellen

Die Formate Axes und Style sind für den 3D-Graphikmodus von besonderer Bedeutung.

Einen Graphen untersuchen

Wie beim graphischen Darstellen von Funktionen können Sie auch hier mit den folgenden Tools einen Graphen untersuchen. Die Koordinaten werden gemäß der Einstellung im Graphikformat entweder in kartesischer Form oder als Zylinderkoordinaten angezeigt. Bei der 3D-Darstellung werden Zylinderkoordinaten angezeigt, wenn Sie mit:

F1 **9**
– oder –
◆ |
◆ **F**
die Einstellung **Coordinates = POLAR** vornehmen.

Tool	**Bei 3D-Darstellungen:**
Frei beweglicher Cursor	Der frei bewegliche Cursor ist nicht verfügbar.
F2 **Zoom**	Funktioniert prinzipiell wie bei Funktionsgraphen; beachten Sie aber, dass Sie hier anstatt mit nur zwei mit drei Dimensionen arbeiten. • Es sind nur folgende Zooms verfügbar: **2:ZoomIn**, **3:ZoomOut**, **5:ZoomSqr**, **6:ZoomStd**, **A:ZoomFit**, **B:Memory**, **C:SetFactors** • Nur **x** (**xmin**, **xmax**), **y** (**ymin**, **ymax**), und **z** (**zmin**, **zmax**) Fenstervariablen sind betroffen. • Die **grid-**(**xgrid**, **ygrid**) und **eye** (**eye**θ, **eye**ϕ, **eye**ψ) Fenstervariablen werden erst dann beeinflußt, wenn Sie **6:ZoomStd** wählen (dadurch werden diese Werte auf die Standardwerte zurückgesetzt).

Tool	Bei 3D-Darstellungen:
F3 **Trace**	Der Cursor wird auf der 3D-Fläche von Rasterpunkt zu Rasterpunkt entlang der Rasterlinie bewegt. • Wenn Sie einen Trace-Vorgang beginnen, befindet sich der Cursor im Zentrum des xy-Rasters. • QuickCenter ist verfügbar. Sie können beim Tracen ENTER unabhängig von der Cursor-Position jederzeit drücken, um den Ansichtswürfel bezüglich des Cursors zu zentrieren. • Die Cursor-Bewegung ist in den Richtungen x und y eingeschränkt. Der Cursor kann nicht über die durch **xmin**, **xmax**, **ymin** und **ymax** bestimmten Grenzen des Ansichtswürfel hinaus bewegt werden.
F5 **Math**	Nur **1:Value** ist für 3D-Darstellungen verfügbar. Dieses Tool zeigt den z-Wert für bestimmte x- und y-Werte an. Nachdem Sie **1:Value** gewählt haben, geben Sie den x-Wert ein, und drücken Sie ENTER. Geben Sie dann den y-Wert ein, und drücken Sie ENTER.

Hinweis: Sie können **z(x,y)** auch während eines Trace-Vorgangs berechnen. Geben Sie den x-Wert ein, und drücken Sie ENTER; geben Sie dann den y-Wert ein, und drücken Sie ENTER.

Den Cursor im 3D-Modus bewegen

Wenn Sie den Cursor über die 3D-Fläche bewegen, ist es vielleicht nicht einsichtig, weshalb er sich so bewegt, wie er es tut. 3D-Darstellungen verfügen über zwei unabhängige Variablen (**x,y**) anstatt nur über eine, und die x- und y-Achsen sind anders ausgerichtet als in anderen Graphikmodi.

So wird der Cursor bewegt

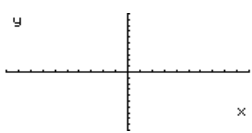
Auf der 3D-Fläche folgt der Cursor in seiner Bewegung stets einer Gitterlinie.

Cursortaste	Der Cursor bewegt sich zum nächsten Gitterpunkt in:
⊙	Positiver x-Richtung
⊙	Negativer x-Richtung
⊙	Positiver y-Richtung
⊙	Negativer y-Richtung

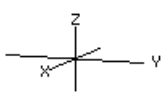
Hinweis: Sie können den Cursor nur innerhalb der durch die Fenstervariablen **xmin**, **xmax**, **ymin** und **ymax** bestimmten Grenzen bewegen.

Obwohl die Regeln sehr klar sind, kann die tatsächliche Bewegung des Cursors verwirrend sein, wenn Sie die Ausrichtung der Achsen nicht kennen.

In zweidimensionalen Graphikmodi haben die x- und y-Achsen bezüglich des Graphikbildschirms stets dieselbe Ausrichtung.



Bei der 3D-Darstellung haben die x- und die y-Achse bezüglich des Graphikbildschirms eine andere Ausrichtung. Außerdem kann man den Betrachtungswinkel drehen und/oder anheben.



eyeθ=20 eyeϕ=70 eyeψ=0

Hinweis: Mit folgenden Tasten werden die Achsen und ihre Beschriftungen im Bildschirm Y= Editor, Window Editor oder Graph angezeigt:

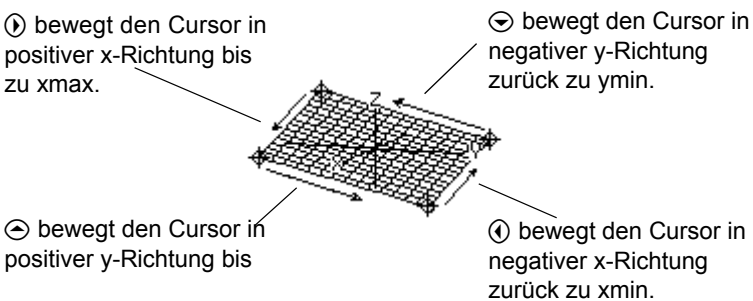
◆ |

◆ **F**

Einfaches Beispiel für die Bewegung des Cursors

Folgender Graph zeigt eine schiefe Ebene mit der Gleichung z1(x,y) = -(x + y) / 2. Sie möchten nun um die angezeigte Grenze herum tracen.

Drücken Sie F3, erscheint der Trace-Cursor im Mittelpunkt des xy-Rasters. Bewegen Sie den Cursor mit Hilfe des Cursorfelds an jeden beliebigen Rand.



Befindet sich der Trace-Cursor an einem Punkt im Innern der angezeigten Ebene, bewegt er sich Gitterpunkt um Gitterpunkt auf einer der Gitterlinien entlang. Das Gitter kann nicht diagonal durchlaufen werden.

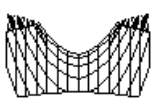
Sie können die Gitterpunkte näher zusammenrücken, indem Sie die Fenstervariablen **xgrid** und **ygrid** erhöhen. Wenn Sie die Achsen anzeigen und beschriften, ist das Prinzip der Cursor-Bewegung klarer erkennbar.

Beachten Sie, dass die Gitterlinien nicht unbedingt parallel zu den Achsen erscheinen müssen.

Beispiel für Cursor auf einer verdeckten Oberfläche

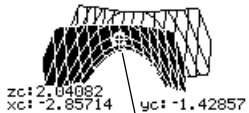
Bei komplexeren Formen kann es aussehen, als ob sich der Cursor nicht auf einem Gitterpunkt befindet. Diese optische Täuschung wird dadurch hervorgerufen, dass der Cursor auf einer verdeckten Oberfläche sitzt.

Gegeben sei beispielsweise eine Sattelform z1(x,y) = $(x^2 - y^2) / 3$. Bei folgendem Graphen wird der Blick entlang der y-Achse abwärts gezeigt.

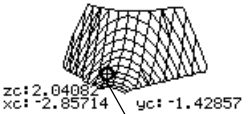


```
eyeθ=90.
eyeϕ=70.
eyeψ=0.
xmin=-10.
xmax=10.
xgrid=14.
ymin=-10.
ymax=10.
ygrid=14.
zmin=-10.
zmax=10.
ncontour=5.
```

Schauen Sie sich dieselbe Form nun aus einem Winkel von 10° zur x-Achse an (eyeθ = 10).



Der Cursor kann so bewegt werden, dass er nicht auf einem Rasterpunkt zu sitzen scheint.



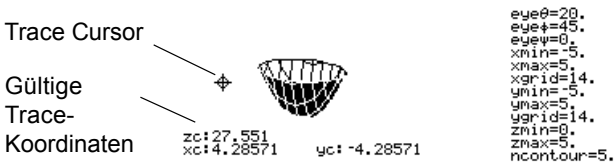
Schneiden Sie die Vorderseite ab, wird erkennbar, dass sich der Cursor auf einem Rasterpunkt der zuvor verdeckten Rückseite befindet.

Hinweis: Schneiden Sie hier die Vorderseite des Sattels ab, indem Sie **xmax=0** einstellen, wodurch nur negative x-Werte angezeigt werden.

Beispiel für einen “nicht auf der Kurve befindlichen” Cursor

Der Cursor kann sich zwar nur entlang der Gitterlinien bewegen, doch Sie werden in vielen Fällen den Eindruck haben, als befände er sich überhaupt nicht auf der 3D-Oberfläche. Dies geschieht, wenn die z-Achse zu kurz ist, um **z(x,y)** für die entsprechenden x- und y-Werte anzuzeigen.

Sie möchten beispielsweise das mit den angegebenen Fenstervariablen dargestellte Paraboloid $z(x,y) = x^2 + .5y^2$ tracen. Dabei können Sie den Cursor leicht in eine Position wie die folgende bringen:



Obgleich der Cursor tatsächlich das Paraboloid traced, scheint er sich nicht auf der Kurve zu befinden, da die Trace-Koordinaten:

- **xc** und **yc** innerhalb des Ansichtswürfel liegen.

  – aber –

- **zc** außerhalb des Ansichtswürfel liegt.

Hinweis: Mit QuickCenter können Sie den Ansichtswürfel bezüglich der Cursorposition zentrieren. Drücken Sie einfach ENTER.

Liegt **zc** außerhalb der z-Grenze des Ansichtswürfel wird der Cursor bei **zmin** oder **zmax** angezeigt (auf dem Bildschirm können aber die korrekten Trace-Koordinaten abgelesen werden).

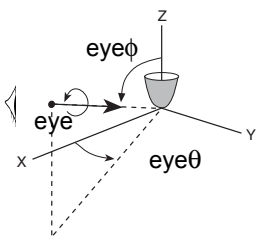
Den Betrachtungswinkel drehen und/oder anheben

Mit den Fenstervariablen **eye**θ und **eye**ϕ können Sie eine 3D-Graphik aus jedem Winkel betrachten. Mit der Fenstervariablen **eye**ψ läßt sich der Graph um diese Betrachtungsachse drehen.

So wird der Betrachtungswinkel gemessen

Der Betrachtungswinkel hat drei Komponenten:

- **eye**θ — Winkel in Grad zur positiven x-Achse.
- **eye**ϕ — Winkel in Grad zur positiven z-Achse.
- **eye**ψ — Winkel in Grad, in welchem der Graph gegen den Uhrzeigersinn um die durch **eye**θ und **eye**ϕ eingestellte Betrachtungsachse gedreht wird.




```
eyeθ=20.
eyeϕ=70.
eyeψ=0.
xmin=-10.
xmax=10.
xgrid=14.
ymin=-10.
ymax=10.
ygrid=14.
zmin=-10.
zmax=10.
ncontour=5.
```

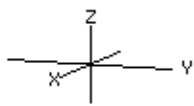
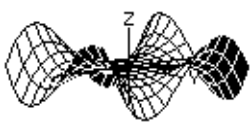
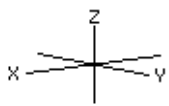
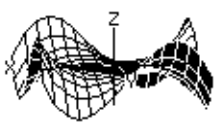
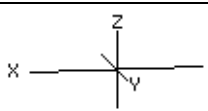
Geben Sie kein °-Symbol ein. Schreiben Sie z.B. 20, 70 und 0, und nicht 20°, 70° und 0°.

Hinweis: Bei **eye**ψ**=0** verläuft die z-Achse senkrecht über den Bildschirm, bei **eye**ψ**=90** wird sie um 90° gegen den Uhrzeigersinn gedreht und verläuft horizontal.

Geben Sie im Window-Editor ([♦] [WINDOW]), unabhängig vom aktuellen Winkelmodus, **eye**θ, **eye**ϕ und **eye**ψ stets in Grad ein.Auswirkung der Änderung von **eye**θ.

Auswirkung der Änderung von eyeθ

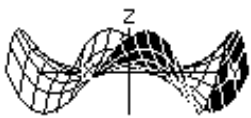
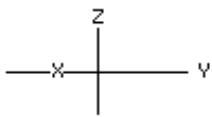
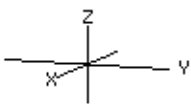
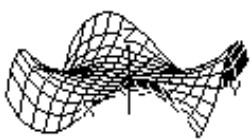
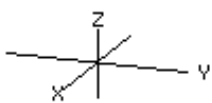
Die Ansicht auf dem Graphikbildschirm richtet sich stets am Betrachtungswinkel aus. Deshalb können Sie durch Änderung von **eye**θ den Betrachtungswinkel um die z-Achse drehen.

z1(x,y) = ($x^3y - y^3x$) / 390		**In diesem Beispiel eyeϕ = 70**
	eyeθ = 20	
	eyeθ = 50	
	eyeθ = 80	

Hinweis: In diesem Beispiel wird **eye**θ um 30 vergrößert.

Auswirkung der Änderung von eyeϕ

Indem Sie **eyeϕ** ändern, können Sie Ihren Betrachtungswinkel über die xy-Ebene verlegen. Bei **90 < eyeϕ < 270** liegt der Betrachtungswinkel unterhalb der xy-Ebene.

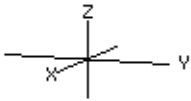
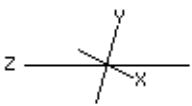
z1(x,y) = ($x^3y - y^3x$) / 390		**In diesem Beispiel eyeθ = 20**
	eyeϕ = 90	
	eyeϕ = 70	
	eyeϕ = 50	

Hinweis: In diesem Beispiel ist die xy-Ebene der Ausgangspunkt (**eyeϕ = 90**), dann wird **eyeϕ** um 20 verringert, um den Betrachtungswinkel nach oben zu versetzen.

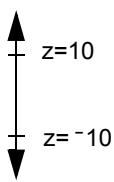
Auswirkung der Änderung von eyeψ

Die Ansicht auf dem Graphikbildschirm richtet sich stets an dem durch **eyeθ** und **eyeϕ** eingestellten Betrachtungswinkel aus. Sie können **eyeψ** ändern, um den Graph um die Betrachtungsachse zu drehen.

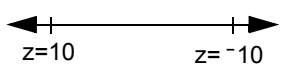
Hinweis: Beim Drehen werden die Achsen zur Anpassung an die Bildschirmbreite- und höhe verlängert bzw. verkürzt. Dadurch entstehen wie im Beispiel Verzerrungen.

z1(x,y)=(x^3y-y^3x) / 390		**In diesem Beispiel eyeθ=20 and eyeϕ=70**
	eyeψ = 0	
	eyeψ = 45	
	eyeψ = 90	

Wenn **eyeψ=0**, verläuft die z-Achse über die Höhe des Bildschirms.



Wenn **eyeψ=90**, verläuft die z-Achse über die Breite des Bildschirms.



Mit der Drehung der z-Achse um 90° wird ihr Bereich (in diesem Beispiel ⁻10 bis 10) auf beinahe das Zweifache ihrer normalen Länge ausgedehnt. Ebenso verlängern bzw. verkürzen sich die Achsen x und y.

Im Hauptbildschirm oder von einem Programm aus

Die für **eye** verwendeten Werte werden in den Systemvariablen **eye**θ, **eye**ϕ und **eye**ψ gespeichert. Sie können bei Bedarf auf diese Variablen zugreifen oder in sie ändern.

Für die Eingabe von ϕ oder ψ drücken Sie ♦ (alpha **[F]** bzw. ♦ (Y. Sie können auch 2nd [CHAR] drücken und das Menü der griechischen Buchstaben verwenden.

Für die Eingabe von ϕ oder ψ, drücken Sie 2nd **G F** bzw. 2nd **G Y**. Sie können auch 2nd [CHAR] drücken und das Menü der griechischen Buchstaben verwenden.

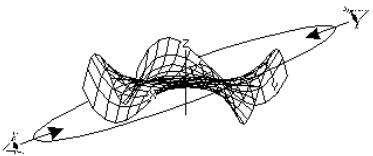
Eine Animation des 3D-Graphen interaktiv erzeugen

Nach dem Plotten eines beliebigen 3D-Graphen können Sie den Betrachtungswinkel mit Hilfe des Cursors interaktiv ändern.

Die Ansichtsspur

Wnn Sie ◀ und ▶ zum Animieren eines Graphen verwenden, denken Sie daran, dass sich der Beobachtungswinkel längs einer “Ansichtsspur” um den Graph bewegt.

Durch die Bewegung entlang dieser Spur kann die z-Achse bei der Animation leicht schwanken.



Hinweis: Die Ansichtsspur wirkt sich unterschiedlich stark auf die Fenstervariablen eye aus.

Eine Graphik animiert anzeigen

Zum:	Gehen Sie wie folgt vor:
Schrittweisen Animieren der Graphik	Drücken Sie kurz auf den Cursor.
Bewegen entlang der Ansichtsspur	◀ oder ▶
Ändern der Neigung der Ansichtsspur (vergößert oder verkleinert in erster Linie eyeϕ)	▲ oder ▼

Zum:	Gehen Sie wie folgt vor:
Der Graph kontinuierlich animieren	Den Cursor ca. 1 Sekunde lang drücken und dann loslassen. Abbruch mit [ESC], [ENTER], [ON], oder [◆] [␣] (Leerzeichen). Abbruch mit [ESC], [ENTER], [ON], oder Leertaste.
Umschalten zwischen vier verschiedenen Animationsgeschwindigkeiten (schrittweise die Fenstervariablen eye vergrößern oder verringern)	[+] oder [−] drücken.
Ändern des Betrachtungswinkels eines nicht-animierten Graphen, um ihn entlang der x-, y- oder z-Achse zu betrachten.	Drücken Sie X, Y bzw. Z.
Zurückkehren zu den Ausgangswerten für eye-Winkel	0 drücken.

Hinweis:

- Bei erweiterter Ansicht des Graphen wird die Anzeige automatisch auf normale Ansicht zurückgeschaltet, sobald Sie eine Cursortaste drücken.
- Nach einer Graphenanimation können Sie den Vorgang stoppen und mit folgenden Tasten die Animation in der gleichen Richtung erneut starten:
  - [ENTER] oder [alpha] [␣]
  - [ENTER] oder Leertaste

- Im Rahmen einer Animation kann der nächste Graphikformatstil mit folgenden Tasten aktiviert werden:

  F
- Sie können eine Graphik anzeigen, in welcher die eye-Winkel sichtbar sind.

Eine Folge von Bildern bewegt anzeigen

Sie können auch eine Animation einer Graphik erzeugen, indem Sie eine Serie von Graphikbildern speichern und dann diese Bilder nacheinander “durchblättern”. Siehe “Eine Folge von Bildern bewegt anzeigen” im Modul *Weitere Darstellungsarten*. Mit dieser Methode haben Sie eine bessere Kontrolle über die Werte der Fenstervariablen, insbesondere von **eye**ψ, welche für die Drehung der Graphik verantwortlich ist.

Das Format für Achsen und Anzeigestil ändern

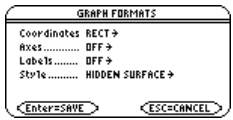
Standardmäßig zeigt der TI-89 Titanium / Voyage™ 200 Graphischer Rechner die verdeckten Oberflächen einer 3D-Darstellung an, die Achsen aber nicht. Sie können das Graphikformat jedoch jederzeit ändern.

Das Dialogfeld GRAPH FORMATS öffnen

Ausgangspunkt ist der Y= Editor, Window-Editor oder der Graphikbildschirm drücken Sie:

F1 **9**
– oder –
◆ I
◆ **F**

- Das Dialogfeld zeigt die aktuellen Graphikformat-Einstellungen.
- Möchten Sie das Dialogfeld verlassen, ohne Änderungen vorzunehmen, drücken Sie ESC.



Die Einstellungen können Sie genauso ändern, wie Sie es von anderen Dialogfeldern, z.B. dem Dialogfeld MODE, kennen.

Beispiel für Achsen-Einstellungen

Markieren Sie die aktuelle Einstellung, und drücken Sie ⊙, um die gültigen **Axes**-Einstellungen anzuzeigen.

$z1(x,y) = x^2+.5y^2$

- AXES — Zeigt Standard-xyz-Achsen.

- BOX — Zeigt dreidimensionale Box-Achsen.

  Die Ränder der Box sind durch die Fenstervariablen **xmin**, **xmax** etc. festgelegt.

  Häufig liegt der Ursprung (0,0,0) im Innern der Box und nicht etwa an einer Ecke. Bei **xmin = ymin = zmin = ⁻10** und **xmax = ymax = zmax = 10** befindet sich der Ursprung z.B. in der Mitte der Box.

Hinweis: Die Einstellung Labels = ON ist bei der Anzeige jedes 3D-Achsentyps hilfreich.

Beispiel für Zeichenstil-Einstellungen

Hinweis: WIRE FRAME läßt sich schneller zeichnen und bietet sich an, wenn Sie mit verschiedenen Formen experimentieren.

Markieren Sie die aktuelle Einstellung, und drücken Sie ⊙, um die gültigen Style-Einstellungen anzuzeigen.

- WIRE FRAME — Zeigt die dreidimensionale Form als durchsichtiges Gittermodell an.

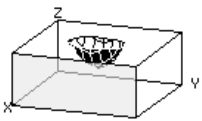
- HIDDEN SURFACES — Setzt die beiden Seiten der dreidimensionalen Form durch Schattierung voneinander ab.

Weiter unten in diesem Modul werden die Einstellungen CONTOUR LEVELS, WIRE AND CONTOUR und IMPLICIT PLOT beschrieben.

Vorsicht vor optischen Täuschungen

Die für die Betrachtung verwendeten eye-Winkel (**eye**θ, **eye**ϕ und **eye**ψ Fenstervariablen) können zu optischen Täuschungen führen, durch welche die Perspektive für Sie unklar werden kann. Die meisten optischen Täuschungen treten bei eye-Winkeln in einem negativen Quadranten des Koordinatensystems auf.

Die optischen Täuschungen machen sich bei Box-Achsen stärker bemerkbar. Es ist beispielsweise nicht sofort klar, welche Seite die “Vorderseite” einer Box ist.

Blick nach unten von oberhalb der xy-Ebene	Blick nach oben von unterhalb der xy-Ebene
eyeθ = 20, eyeϕ = 55, eyeψ = 0	eyeθ = 20, eyeϕ = 120, eyeψ= 0
	

Hinweis: Die ersten beiden Beispiele stellen die Graphen dar, wie sie auf dem Bildschirm angezeigt werden. Die zweiten beiden Beispiele ist die Vorderseite der Box schattiert hervorgehoben (nur zur Verdeutlichung, nicht auf dem Bildschirm angezeigt).

Den Effekt der optischen Täuschungen können Sie einschränken, indem Sie mit dem Dialogfeld GRAPH FORMATS die Einstellung Style = HIDDEN SURFACE vornehmen.

Konturenplots

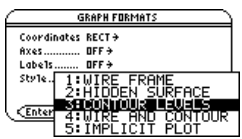
In einem Konturenplot wird eine Linie zur Verbindung nebeneinanderliegender Punkte in der 3D-Graphik gezogen, die denselben z-Wert besitzen (Höhenlinien). In diesem

Abschnitt werden die Graphikformat-Stile CONTOUR LEVELS und WIRE AND CONTOUR behandelt.

Wahl des Graphikformat-Stils

Definieren Sie im 3D-Graphikmodus eine Fläche, und stellen Sie diese, wie Sie es für 3D-Darstellugen gewöhnt sind graphisch dar. Einziger Unterschied zur üblichen Vorgehensweise ist: Zeigen sie das Dialogfeld GRAPH FORMATS an, indem Sie im Y=Editor, Window-Editor oder im Graphikbildschirm F1 **9** drücken. Nehmen Sie dann folgende Einstellungen vor:

Style = CONTOUR LEVELS
– oder –
Style = WIRE AND CONTOUR

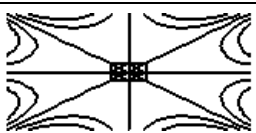
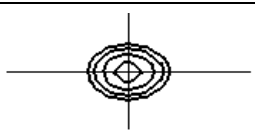
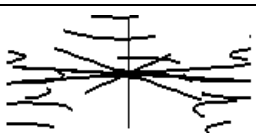
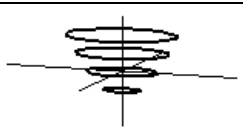


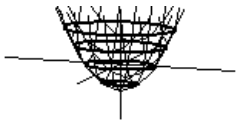
- Mit CONTOUR LEVELS werden nur die Konturen angezeigt.
  - Der Betrachtungswinkel wird zu Beginn so eingestellt, daß Sie entlang der z-Achse abwärts auf die Konturen blicken. Bei Bedarf kann der Betrachtungswinkel geändert werden.
  - Der Graph wird in erweiterter Ansicht angezeigt. Zum Umschalten zwischen erweiterter und normaler Ansicht drücken Sie ×.
  - Das Format Labels wird automatisch auf OFF gesetzt.
- Mit WIRE AND CONTOUR werden die Konturen in einer Drahtmodellansicht gezeichnet. Die vorherigen Einstellungen für Betrachtungswinkel, Ansicht (erweitert oder normal) und das Format Labels werden beibehalten.

Hinweis:

- Im Graphenbildschirm wird mit folgenden Tasten (unter Umgehung von IMPLICIT PLOT) ein neues Graphenformat gewählt:
  [calculator] [|]
  [keyboard] **F**

- Wenn Sie
  [calculator] [|]
  [keyboard] **F**
  zur Wahl von CONTOUR LEVELS drücken, hat dies im Gegensatz zur Verwendung von [|] keinen Einfluß auf den Betrachtungswinkel, die Ansicht oder das Format Labels:
  [calculator] [♦] [|]
  [keyboard] [♦] **F**

Style	**$z1(x,y)=(x^3y-y^3x)/390$**	**$z1(x,y)=x^2+.5y^2-5$**
	Blick entlang der z-Achse abwärts	
CONTOUR LEVELS		
	Mit eyeθ=20, eyeϕ=70, eyeψ=0	
CONTOUR LEVELS		

Style	$z1(x,y)=(x^3y-y^3x)/390$	$z1(x,y)=x^2+.5y^2-5$
WIRE AND CONTOUR		

Hinweis: In diesen Beispielen werden dieselben Werte für die Fenstervariablen x, y und z verwendet wie für einen **ZoomStd**-Ansichtswürfel. Wenn Sie **ZoomStd** verwenden, drücken Sie Z, um entlang der z-Achse abwärts zu blicken. Verwechseln Sie die Konturen nicht mit den Gitterlinien. Die Konturen sind dunkler.

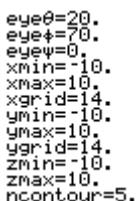
Wie werden z-Werte bestimmt?

Mit den Fenstervariablen ncontour (◆ [WINDOW]) läßt sich die Anzahl der Konturen festlegen, die gleichmäßig entlang des angezeigten Bereichs an z-Werten verteilt werden. Hierbei gilt:

$$\text{Zuwachs} = \frac{\text{zmax} - \text{zmin}}{\text{ncontour} + 1}$$

Die z-Werte für die Konturen sind:

zmin + Zuwachs
zmin + 2(Zuwachs)
zmin + 3(Zuwachs)
⋮
zmin + ncontour(Zuwachs)



Vorgabe ist 5.
Möglicher Einstellungsbereich ist 0 bis 20.

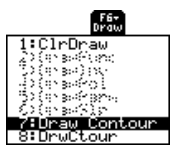
Wenn **ncontour=5** ist und Sie das Standard-Ansichtfenster verwenden (**zmin=⁻10** und **zmax=10**), ist der Zuwachs 3.333. Fünf Konturlinien werden bei z=⁻6.666, ⁻3.333, 0, 3.333 und 6.666 gezeichnet.

Beachten Sie aber, daß für einen z-Wert dann keine Kontur gezeichnet wird, wenn die 3D-Graphik in diesem z-Wert nicht definiert ist.

Interaktives Zeichnen einer Kontur für den z-Wert eines gewählten Punktes

Auf einem aktuellen Kontur-Graph können Sie einen Punkt angeben und eine Kontur für dessen z-Wert zeichnen.

1. Das Zeichenmenü (**Draw)** wird wie folgt aktiviert:
   [2nd] [F6]
   [F6]



2. Wählen Sie **7:Draw Contour**.
3. Sie haben folgende Möglichkeiten:
   - Geben Sie den x-Wert des Punktes ein, und drücken Sie [ENTER]; geben Sie dann den y-Wert ein, und drücken Sie [ENTER].

     – oder –
   - Setzen Sie den Cursor auf den betreffenden Punkt. (Der Cursor bewegt sich entlang der Gitterlinien.) Drücken Sie dann [ENTER].

Gehen Sie beispielsweise von dem Graph $z1(x,y)=x^2+.5y^2-5$ aus. Geben Sie x=2 und y=3 an, so wird eine Kontur für z=3.5 gezeichnet.

Hinweis: Alle vorhandenen Konturen bleiben in dem Graph. Möchten Sie die Standardkonturen entfernen, zeigen Sie den Window-Editor an ([♦] [WINDOW]), und stellen Sie **ncontour=0** ein.

Konturen für angegebene z-Werte zeichnen

Rufen Sie im Graphenbildschirm das Zeichenmenü auf, und wählen Sie die Option **8:DrwCtour**. Es wird automatisch der Hauptbildschirm mit der Angabe **DrwCtour** in der Eingabezeile angezeigt. Dann können Sie entweder einen oder mehrere z-Werte einzeln angeben oder auch eine Folge von z-Werten erzeugen.

Dies sind einige Beispiele:

DrwCtour 5	Zeichnet eine Kontur für z=5.
DrwCtour {1,2,3}	Zeichnet Konturen für z=1, 2 und 3.
DrwCtour seq(n,n,⁻10,10,2)	Zeichnet Konturen für eine Folge von z-Werten von ⁻10 bis 10 in Schritten von 2 (⁻10, ⁻8, ⁻6, etc.).

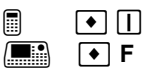
Hinweis: Zum Entfernen der Standardkonturen drücken Sie [♦] [WINDOW] und stellen **ncontour=0** ein.

In der aktuellen 3D-Graphik werden die angegebenen Konturen gezeichnet. (Es wird keine Kontur gezeichnet, wenn der angegebene z-Wert außerhalb des Ansichtswürfels liegt oder die 3D-Graphik in diesem z-Wert nicht definiert ist.)

Anmerkungen zu Konturenplots

Bei einem Konturenplot:

- Können Sie die Cursortasten verwenden, um den Konturenplot bewegt anzuzeigen.
- Können Sie die Konturen selbst nicht tracen (F3). Es ist aber möglich, das Drahtmodell, wie es bei der Einstellung Style=WIRE AND CONTOUR angezeigt wird, zu tracen.
- Kann die Auswertung der Gleichung anfangs ein wenig Zeit beanspruchen.
- Angesichts der möglichen langen Auswertungszeit bietet es sich an, daß Sie zunächst ein wenig mit Ihrer 3D-Gleichung und Style=WIRE FRAME experimentieren. Die Auswertungszeit ist wesentlich kürzer. Sind die Fenstervariablen Ihrer Ansicht nach korrekt, rufen Sie das Dialogfeld für Graphenformate auf, und wählen Sie für Stil die Option CONTOUR LEVELS oder WIRE AND CONTOUR.



Beispiel: Konturen der “Betragsfläche eines komplexen Terms”

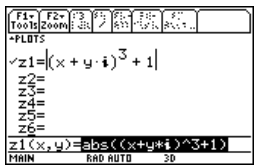
Die durch **z(a,b) = abs(f(a+b*i*))** gegebene Betragsfläche eines komplexen Terms zeigt sämtliche komplexen Nullstellen eines Polynoms **y=f(x)** auf.

Beispiel

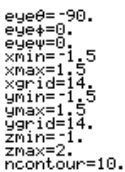
Wählen Sie für dieses Beispiel f(x)= x^3+1. Durch Einsetzen der allgemeinen komplexen Form x+yi für x kann die Gleichung der Fläche als z(x,y)=abs((x+y i)^3+1) ausgedrückt werden.

1. Stellen Sie mit MODE **Graph=3D** ein.
2. Drücken Sie ♦ [Y=], und definieren Sie die Gleichung:

   z1(x,y)=abs((x+y∗*i*)^3+1)



3. Drücken Sie ♦ [WINDOW], und stellen Sie die Fenstervariablen wie gezeigt ein.

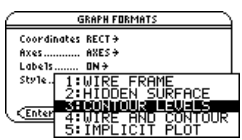


4. Aktivieren Sie das Dialogfeld für Graphenformate wie folgt:

   ♦ |

   ♦ **F**

   Blenden Sie die Achsen ein, wählen Sie für Stil die **Style = CONTOUR LEVELS** und kehren Sie zum Fenster-Editor zurück.



5. Drücken Sie ♦ [GRAPH], um die Gleichung graphisch darzustellen.

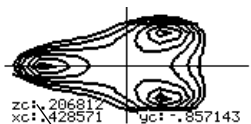
   Nun brauchen Sie ein wenig Geduld, denn die Berechnung des Graphen nimmt einige Zeit in Anspruch. Wenn der Graph angezeigt wird berührt die Betragsfläche

eines komplexen Terms die xy-Ebene genau in den komplexen Nullstellen des Polynoms:

-1, $\frac{1}{2}+\frac{\sqrt{3}}{2}i$, und $\frac{1}{2}-\frac{\sqrt{3}}{2}i$

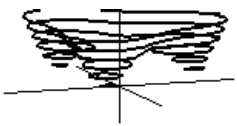
6. Drücken Sie [F3], und setzen Sie den Trace-Cursor auf den Nullpunkt im vierten Quadranten.

   Anhand der Koordinaten kann der Nullpunkt auf .428–.857i geschätzt werden.



Wenn z=0, dann ist der Nullpunkt exakt.

7. Drücken Sie [ESC]. Verwenden Sie dann die Cursortasten, um die Graphik bewegt anzuzeigen und aus unterschiedlichen eye-Winkeln zu betrachten.



Dieses Beispiel zeigt eyeθ=70, eyeϕ=70, und eyeψ=0.

Hinweis:

- Eine größere Genauigkeit läßt sich durch Erhöhung der Fenstervariablen **xgrid** und **ygrid** erzielen. Dadurch wird allerdings die Rechenzeit verlängert.

- Bei einer Animation der Graphik schaltet die Anzeige auf normale Ansicht um. Verwenden Sie [×] zum Umschalten zwischen normaler und erweiterter Ansicht.

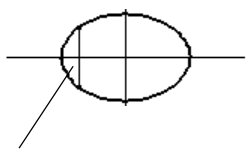
Implizite Plots

Ein impliziter Plot wird hauptsächlich zum Darstellen impliziter Gleichungen verwendet, die im Funktionsdarstellungs-Modus nicht graphisch dargestellt werden können. Technisch gesehen ist ein impliziter Plot ein 3D-Konturenplot mit nur einer einzigen für **z=0** gezeichneten Kontur.

Explizite und implizite Gleichungen

Im Graph-Modus für die zweidimensionale graphische Darstellung von Funktionen haben Gleichungen eine explizite Form **y=f(x)**, wobei y für jeden x-Wert eindeutig ist.

Viele Gleichungen besitzen jedoch eine implizite Form **f(x,y)=g(x,y)**, die nicht explizit nach y oder nach x aufgelöst werden kann.



y ist nicht für jedes x eindeutig, deshalb ist keine Darstellung im Graph-Modus für Funktionen möglich.

Mit Hilfe impliziter Plots im 3D-Graphikmodus können diese impliziten Formen graphisch dargestellt werden, ohne daß nach y oder x aufgelöst wird.

Ordnen Sie die implizite Form als in gleich Null gesetzte Gleichung um.

$$f(x,y)–g(x,y)=0$$

Geben Sie die von Null verschiedene Seite der Gleichung in den Y=Editor ein. Dies ist deshalb zulässig, weil ein impliziter Plot die Gleichung automatisch gleich Null setzt.

$z1(x,y)=f(x,y)–g(x,y)$

Geben Sie für die Ellipsengleichung in nebenstehendem Beispiel die implizite Form in den Y= Editor ein.

Wenn $x^2+.5y^2=30$,
dann $z1(x,y)=x^2+.5y^2–30$.

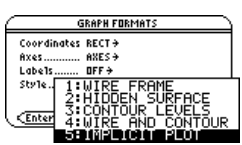
Hinweis: Viele implizite Formen lassen sich außerdem graphisch darstellen, indem Sie:

- Diese entweder als Parametergleichungen ausdrücken.
- Oder sie in explizite Funktionen aufspalten.

Wahl des Graphikformat-Stils

Definieren Sie im 3D-Graphikmodus eine Gleichung, und stellen Sie diese wie Sie es für 3D-Gleichungen gewöhnt sind graphisch dar. Einziger Unterschied zur üblichen Vorgehensweise ist: Zeigen sie das Dialogfeld GRAPH FORMATS vom Y= Editor, Window-Editor oder vom Graphikbildschirm aus an. Nehmen Sie dann folgende Einstellungen vor:

Wählen Sie: **Style = IMPLICIT PLOT**



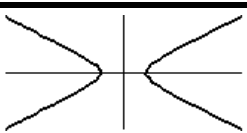
Hinweis: Im Graphikbildschirm erfolgt der Wechsel zu anderen Graphikformatstilen mit folgenden Tasten:

F

So wird wieder IMPLICIT PLOT angezeigt:

F

- Der Betrachtungswinkel wird zu Beginn so eingestellt, daß Sie entlang der z-Achse abwärts auf den Plot blicken. Bei Bedarf kann der Betrachtungswinkel geändert werden.
- Der Plot wird in erweiterter Ansicht angezeigt. Zum Umschalten zwischen erweiterter und normaler Ansicht drücken Sie ⊠.
- Das Format Labels wird automatisch auf OFF gesetzt.

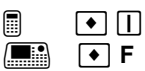
Style	**$x^2-y^2=4$** **z1(x,y)=x^2-y^2-4**	**sin(x)+cos(y)= e(x∗y)** **z1(x,y)=sin(x)+cos(y)–e(x∗y)**
IMPLICIT PLOT		

Hinweis: In diesen Beispielen werden dieselben Werte für die Fenstervariablen x, y und z verwendet wie für einen **ZoomStd**-Ansichtswürfel. Wenn Sie **ZoomStd** verwenden, drücken Sie Z, um entlang der z-Achse abwärts zu blicken.

Anmerkungen zu impliziten Plots

Bei einem impliziten Plot:

- Hat die Fenstervariable **ncontour** keine Wirkung. Unabhängig vom Wert für ncontour wird nur die Kontur **z=0** gezeichnet. Der angezeigte Plot zeigt, wo die Fläche die xy-Ebene schneidet.
- Können Sie die Cursortasten verwenden, um den Plot bewegt anzuzeigen.
- Können Sie den impliziten Plot selbst nicht tracen (F3). Es ist aber möglich, das nicht sichtbare Drahtmodell der 3D-Gleichung zu tracen.
- Kann die Auswertung der Gleichung anfangs ein wenig Zeit beanspruchen.
- Angesichts der möglichen langen Rechenzeit bietet es sich an, daß Sie zunächst ein wenig mit Ihrer 3D-Gleichung und Style=WIRE FRAME experimentieren. Die Rechenzeit ist wesentlich kürzer. Wenn Sie sich dann sicher sind, daß Sie die richtigen Werte für die Fenstervariablen gefunden haben und stellen Sie Style auf IMPLICIT PLOT ein.



Beispiel: Impliziter Plot einer komplizierteren Gleichung

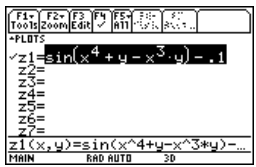
Mit dem Graphikformat-Stil IMPLICIT PLOT lassen sich komplizierte Gleichungen plotten und bewegt anzeigen, für welche keine andere graphische Darstellungsart möglich ist. Das sichtbare Ergebnis rechtfertigt die oft lange Rechenzeit.

Beispiel

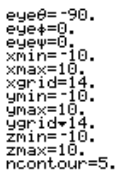
Stellen Sie die Gleichung $\sin(x^4+y-x^3 y) = .1$ graphisch dar.

1. Stellen Sie mit [MODE] **Graph=3D** ein.
2. Drücken Sie [♦] [Y=], und definieren Sie die Gleichung:

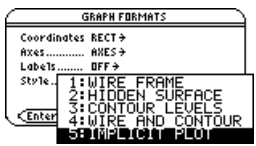
   z1(x,y)=sin(x^4+y – x^3y) –.1



3. Drücken Sie [♦] [WINDOW], und stellen Sie die Fenstervariablen wie gezeigt ein.

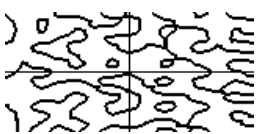


4. Drücken Sie
   [♦] [|]
   [♦] **F**
   schalten Sie die Achsen ein, setzen Sie **Style** auf **IMPLICIT PLOT**, und kehren Sie zum Window-Editor zurück.



5. Drücken Sie [♦] [GRAPH], um die Gleichung graphisch darzustellen.

   Geduld - die Berechnung des Graphen dauert eine Weile.



Der Graph zeigt, wo $\sin(x^4+y-x^3y) = .1$ ist.

6. Verwenden Sie die Cursortasten, um die Graphik bewegt anzuzeigen und aus unterschiedlichen eye-Winkeln zu betrachten.

   Hinweis: Eine größere Genauigkeit läßt sich durch Erhöhung der Fenstervariablen **xgrid** und **ygrid** erzielen. Dadurch wird allerdings die Rechenzeit verlängert.

In der erweiterten Ansicht zeigt dieses Beispiel eyeθ=⁻127.85, eyeϕ=52.86, und eyeψ=⁻18.26.

Hinweis: Bei einer Animation der Graphik schaltet die Anzeige auf normale Ansicht um. Verwenden Sie ⊠ zum Umschalten zwischen normaler und erweiterter Ansicht.

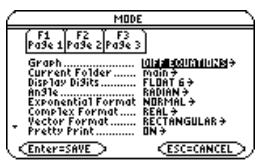
Graphische Darstellung von Differentialgleichungen

Schritte zur graphischen Darstellung von Differentialgleichungen

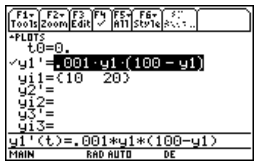
Zur graphischen Darstellung von Differentialgleichungen gehen Sie nach den Hauptschritten vor, wie sie im Modul *Grundlagen der graphischen Darstellung von Funktionen* für **y(x)**-Funktionen beschrieben sind.

Differentialgleichungen graphisch darstellen

1. Stellen Sie **Graph**-Modus (MODE) auf **DIFF EQUATIONS**. Falls erforderlich, auch den **Angle**-Modus einstellen.



2. Definieren Sie im Y= Editor ([♦] [Y=]) die Gleichungen und, falls gewünscht, Anfangsbedingungen.



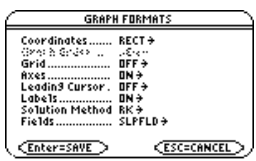
3. Wählen Sie mit ([F4]), welche definierten Gleichungen graphisch dargestellt werden sollen.

   Hinweis: Möchten Sie eventuelle stat-Daten-Plots ausschalten, drücken Sie [F5] **5**, oder [F4], um deren Auswahl aufzuheben.

4. Wählen Sie das Anzeigeformat für eine Gleichung.

   [2nd] [F6]
   [F6]

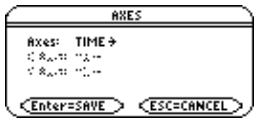
5. Wählen Sie das GraphikFormat. **Solution Method** (**Lösungsverfahren)** und **Fields** (**Felder)** sind bei Differentialgleichungen eindeutig.



   [F1] **9**
   — oder —
   [♦] [|]
   [♦] **F**

   Hinweis: Das Format **Fields** ist wichtig, je nach der Ordnung der Differentialgleichung.

6. Definieren Sie die Achsenbeschriftungen in Abhängigkeit vom Fieldformat.



[2nd] [F7]
[F7]

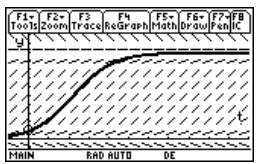
Hinweis: Gültige Einstellungen für **Axes** sind vom Format **Fields** abhängig

7. Definieren Sie das Ansichtfenster ([◆] [WINDOW]).

```
t0=0.
tmax=10.
tstep=.1
tplot=0.
xmin=-10.
xmax=110.
xscl=10.
ymin=-10.
ymax=120.
yscl=10.
ncurves=0.
diftol=.001
fldres=20.
```

Hinweis: Welche Fenstervariablen angezeigt werden, hängt vom Format **Solution Method** und **Fields** ab. [F2] **Zoom** beeinfluß auch das Anzeigefenster.

8. Stellen Sie die Gleichungen graphisch dar ([◆] [GRAPH]).



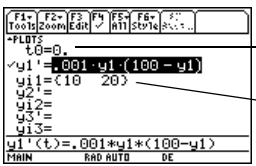
Unterschiede zwischen der graphischen Darstellung von Differentialgleichungen und von Funktionen

In diesem Einheit wird vorausgesetzt, dass Sie bereits wissen, wie **y(x)**-Funktionen gemäß der Beschreibung in *Grundlagen der graphischen Darstellung von Funktionen* graphisch dargestellt werden.

Den Graph-Modus einstellen

Bevor Sie Differentialgleichungen definieren oder Fenstervariable einstellen, nehmen Sie mit MODE die Einstellung **Graph = DIFF EQUATIONS** vor. Mit dem Y= Editor und dem Window-Editor können Sie nur Informationen für die aktuelle Graph-Modus-Einstellung eingeben.

Differentialgleichungen im Y= Editor definieren



Bestimmen Sie mit t0, wenn die Anfangsbedingung auftritt. t0 kann auch im Window-Editor eingestellt werden.

Geben Sie mit yi eine oder mehrere Anfangsbedingungen für die entsprechende Differentialgleichung an.

Sie können Differentialgleichungen von y1'(t) bis y99'(t) definieren.

Hinweis: Zum Definieren von Funktionen und Gleichungen können Sie im Hauptbildschirm den Befehl **Define** verwenden.

Verwenden Sie bei der Eingabe von Gleichungen im Y= Editor keine **y(t)**-Formate, um auf Ergebnisse zu verweisen. Beispiel:

Verwenden Sie keine implizite Multiplikation zwischen einer Variablen und einem Klammerausdruck. Die Eingabe wird sonst als Funktionsaufruf behandelt.

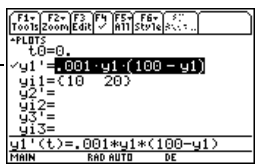
Richtig: y1' = .001y1∗(100−y1)
Falsch: y1' = .001y1(t)∗(100−y1(t))

Im Y= Editor können nur Gleichungen erster Ordnung eingegeben werden. Möchten Sie Gleichungen zweiter oder höherer Ordnung graphisch darstellen, sind diese als ein System von Gleichungen erster Ordnung einzugeben.

Die Einzelheiten zum Einstellen der Ausgangseinstellungen sind ausführlich beschrieben.

Differentialgleichungen auswählen

Mit F4 können Sie eine Differentialgleichung, nicht aber ihre Anfangsbedingung wählen.



Wichtig: Wählen Sie y1', wird in Abhängigkeit von der Achseneinstellung nicht die Ableitung y1', sondern die y1-Lösungskurve graphisch dargestellt.

Wahl des Zeichenstils

Im Stilmenü sind nur die Optionen **Line**, **Dot**, **Square**, **Thick**, **Animate** und **Path** verfügbar. **Dot** und **Square** kennzeichnen nur die diskreten Werte (in **tstep**-Inkrementen), in welchen eine Differentialgleichung geplottet wird.

2nd F6

F6

Graphik-Formate einstellen

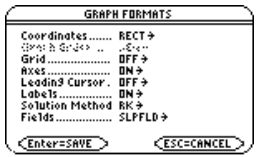
Drücken Sie im Y= Editor, Window-Editor oder Graphikbildschirm:

F1 **9**

— oder —

♦ |

♦ **F**



Folgende Formate sind von Differentialgleichungen betroffen:

Graph-Format	Beschreibung
Graph Order	Nicht verfügbar.
Solution Method	Gibt die zum Lösen der Differentialgleichungen zu verwendende Methode an. • RK — Runge-Kutta-Verfahren. Informationen zu dem für diese Methode verwendeten Algorithmus sind im Modul *Technische Referenz* enthalten. • EULER — Eulersches Verfahren. • Mit dieser Methode können Sie entweder eine höhere Genauigkeit oder Geschwindigkeit wählen. RK ist typischerweise genauer als EULER, die Lösungsfindung dauert aber länger.

Graph-Format	Beschreibung
Fields	Gibt an, ob ein Feld für die Differentialgleichung gezeichnet werden soll. • SLPFLD — Zeichnet ein Steigungsfeld für eine einzige Gleichung erster Ordnung, wobei t auf der x-Achse und die Lösung auf der y-Achse liegt. • DIRFLD — Zeichnet ein Richtungsfeld für eine einzige Gleichung zweiter Ordnung (oder ein System von zwei Gleichungen erster Ordnung), wobei die Achsen durch die vom Benutzer vorgenommenen Achsen-Einstellungen definiert sind. • FLDOFF — Zeigt kein Feld an. Dieses Format ist für Gleichungen jeder Ordnung gültig, muß aber für Gleichungen dritter oder höherer Ordnung verwendet werden. Für jede Gleichung müssen im Y= Editor gleich viele Anfangsbedingungen eingegeben werden.

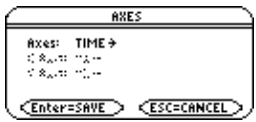
Wichtig: Zum erfolgreichen Darstellen von Differentialgleichungen ist das Graphikformat Fields von grundlegender Bedeutung.

Hinweis: Drücken Sie während des Zeichnens eines Steigungs- oder Richtungsfeldes [ENTER], wird der Graph nach der Zeichnung des Feldes und vor dem Plotten der Lösungen unterbrochen. Zum Fortfahren drücken Sie erneut [ENTER]. Drücken Sie [ON], um das Zeichnen abzubrechen.

Achsen einstellen

Ob **Axes** im Y= Editor verfügbar ist oder nicht, hängt vom aktuellen Graphikformat ab.

Wenn verfügbar, können Sie die für die graphische Darstellung der Differentialgleichungen verwendeten Achsen wählen.



[2nd] [F7]

[F7]

Achsen	Beschreibung
TIME	Plottet t auf der x-Achse und y (die Lösungen der gewählten Differentialgleichungen) auf der y-Achse.
CUSTOM	Dient zum Wählen der x- und y-Achse.

Fenstervariable

Für die Darstellung von Differentialgleichungen werden folgende Fenstervariable verwendet. Ob alle Variablen gleichzeitig im Window-Editor ([♦] [WINDOW]) aufgeführt sind, hängt von den Graphikformaten für **Solution Method** und **Fields** ab.

Variable	Beschreibung
t0	Zeitpunkt, zu welchem die im Y= Editor eingegebenen Anfangsbedingungen auftreten. **t0** kann sowohl im Window-Editor als auch im Y= Editor eingestellt werden (bei Einstellung im Y= Editor wird **tplot** automatisch auf denselben Wert gesetzt).

Variable	Beschreibung
tmax, **tstep**	Dient zur Bestimmung der t-Werte, in welchen die Gleichungen geplottet werden: y'(t0) y'(t0+tstep) y'(t0+2∗tstep) ... bis maximal ... y'(tmax) Wenn **Fields = SLPFLD**, so wird **tmax** ignoriert. Die Gleichungen werden von **t0** aus in **tstep**-Inkrementen zu beiden Bildschirmseiten hin geplottet.
tplot	Der erste geplottete t-Wert. Handelt es sich dabei nicht um ein **tstep**-Inkrement, beginnt der Plot beim nächsten **tstep**-Inkrement. In einigen Situationen sind die ersten berechneten und von **t0** ausgehend geplotteten Punkte für die optische Darstellung nicht interessant. Wird **tplot** größer als **t0** eingestellt, können Sie den Plot im interessanten Bereich beginnen. Dadurch wird der Vorgang beschleunigt und die Anhäufung unnötiger Daten auf dem Graphikbildschirm vermieden.

Hinweis: Wenn **tmax < t0**, so muß **tstep** negativ sein. Wenn **Fields=SLPFLD**, so wird **tplot** ignoriert. Für **tplot** wird dann derselbe Wert wie für **t0** angenommen.

Variable	Beschreibung
xmin, **xmax,** **ymin**, **ymax**	Grenzen des Ansichtfensters.
xscl, **yscl**	Abstand zwischen den Einheiten auf der x- und y-Achse.

Variable	Beschreibung
ncurves	Anzahl der Lösungskurven (0 bis 10), die automatisch gezeichnet werden, wenn Sie keine Anfangsbedingung angeben. Standardgemäß ist **ncurves = 0**.
	Wird **ncurves** verwendet, so wird **t0** kurzzeitig in die Bildschirmmitte verschoben, und die Anfangsbedingungen werden gleichmäßig über die y-Achse verteilt; hierbei gilt: $increment = \frac{ymax - ymin}{ncurves + 1}$ Die y-Werte für die Anfangsbedingungen sind: ymin + increment ymin + 2∗(increment) ⋮ ymin + ncurves∗(increment)
diftol	(Nur bei **Solution Method = RK**) Die beim **RK**-Verfahren für die Wahl einer Schrittweite zum Auflösen der Gleichung verwendete Toleranz; sie muß ≥1E⁻14 sein.
fldres	(Nur bei **Fields = SLPFLD** oder **DIRFLD**) Anzahl der für die Zeichnung eines Steigungs- oder Richtungsfeldes über die gesamte Bildschirmbreite verwendeten Spalten (1 bis 80).
Estep	(Nur bei **Solution Method = EULER**) Eulersche Iterationen zwischen **tstep**-Werten; muß eine ganze Zahl >0 sein. Eine größere Genauigkeit erreichen Sie, wenn Sie **Estep** erhöhen und keine zusätzlichen Punkte plotten.

Variable	Beschreibung
dtime	(Nur bei **Fields = DIRFLD**) Ein Zeitpunkt, zu welchem ein Richtungsfeld gezeichnet wird.

Die Standardeinstellungen (die Sie vorfinden, wenn Sie im F2 **Zoom**-Symbolleistenmenü **6:ZoomStd** wählen) sind:

t0 = 0.	xmin = ⁻1.	ymin = ⁻10.	ncurves = 0.
tmax = 10.	xmax = 10.	ymax = 10.	diftol = .001
tstep = .1	xscl = 1.	yscl = 1.	Estep = 1.
tplot = 0.			fldres = 14.
			dtime = 0.

Um sicherzustellen, dass genügend Punkte geplottet werden, müssen Sie die Standardwerte für die t-Variablen möglicherweise ändern.

Die Systemvariable fldpic

Beim Zeichnen eines Steigungs- oder Richtungsfeldes wird automatisch ein Bild dieses Feldes in eine Systemvariable namens **fldpic** gespeichert. Führen Sie dann eine Operation durch, bei welcher die geplotteten Gleichungen erneut graphisch dargestellt, nicht aber die Felder beeinflußt werden, so verwendet der TI-89 Titanium / Voyage™ 200 Graphischer Rechner das in **fldpic** gespeicherte Bild, anstatt das Feld neu zeichnen zu müssen. Dies kann die Zeit für die graphische Darstellung bedeutend verkürzen.

Wenn Sie den Modus für die graphische Darstellung von Differentialgleichungen verlassen oder einen Graphen mit **Fields = FLDOFF** anzeigen, wird **fldpic** automatisch gelöscht.

Einen Graphen untersuchen

Wie bei der graphischen Darstellung von Funktionen können Sie einen Graphen mit Hilfe folgender Instrumente untersuchen. Koordinaten werden je nach Einstellung im Graphikformat in kartesischer oder polarer Form angezeigt.

Instrument	Für graphische Darstellung von Differentialgleichungen:
Frei beweglicher Cursor	Funktioniert wie bei der graphischen Darstellung von Funktionen.
F2 **Zoom**	Funktioniert wie bei der graphischen Darstellung von Funktionen. • Nur die Fenstervariablen **x** (**xmin**, **xmax**, **xscl**) und **y** (**ymin**, **ymax**, **yscl**) sind betroffen. • Die Fenstervariablen **t** (**t0**, **tmax**, **tstep**, **tplot**) sind nur dann betroffen, wenn Sie **6:ZoomStd** wählen (wodurch alle Fenstervariablen auf ihre Standardwerte zurückgesetzt werden).

Instrument	Für graphische Darstellung von Differentialgleichungen:
F3 **Trace**	Dient zum Bewegen des Cursors entlang der Kurve um jeweils einen **tstep**. Um den Cursor jeweils um ca. zehn geplottete Punkte weiterzubewegen, drücken Sie 2nd ⊙ oder 2nd ⊙. Geben Sie im Y= Editor Anfangsbedingungen ein oder lassen die Fenstervariable **ncurves** automatisch Kurven plotten, können Sie die Kurven tracen. Wählen Sie im Graphikbildschirm: 2nd F8 F8 IC, um interaktiv Anfangsbedingungen zu wählen, so können Sie die Kurven nicht tracen. QuickCenter gilt für alle Richtungen. Wenn Sie den Cursor über den Bildschirmrand hinaus verschieben (oben, unten, rechts oder links), können Sie ENTER drücken, um das Ansichtfenster bezüglich der Cursorposition zu zentrieren. Verwenden Sie ⊙ oder ⊙, um auf alle geplotteten Kurven anzusehen.
F5 **Math**	Nur **1:Value** ist verfügbar. • Mit TIME-Achsen wird der **y(t)**-Lösungswert (dargestellt durch yc) für einen bestimmten t-Wert angezeigt. • Mit CUSTOM-Achsen hängen die x und y entsprechenden Werte von den gewählten Achsen ab.

Hinweis: Beim Tracen können Sie den Cursor auf einen bestimmten Punkt setzen, indem Sie einen Wert für t eingeben und ENTER drücken. Sie können QuickCenter während des Zeichnens jederzeit verwenden, selbst wenn sich der Cursor noch auf dem Bildschirm befindet.

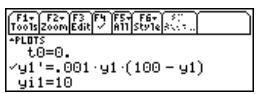
Einstellen der Anfangsbedingungen

Anfangsbedingungen können Sie entweder im Y= Editor eingeben, automatisch vom TI-89 Titanium / Voyage™ 200 Graphischer Rechner berechnen lassen oder aber interaktiv im Graphikbildschirm wählen.

Eingabe von Anfangsbedingungen im Y= Editor

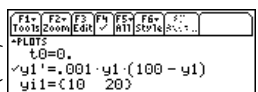
Sie können im Y= Editor eine oder mehrere Anfangsbedingungen angeben. Um mehr als eine anzugeben, müssen Sie diese als Liste in Klammern { } und durch Kommata voneinander getrennt eingeben.

Zur Eingabe von Anfangsbedingungen für die **y1'**-Gleichung verwenden Sie die Zeile **yi1**; etc.



Mit t0 geben Sie an, wann die Anfangsbedingungen auftreten. Dies ist auch das erste für den Graphen ausgewertete t.

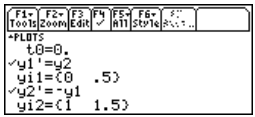
Um eine Lösungsfamilie graphisch darzustellen, geben Sie eine Liste von Anfangsbedingungen ein.



Sie geben {10,20} ein, auch wenn {10 20} angezeigt wird.

Für eine Differentialgleichung zweiter oder höherer Ordnung müssen Sie im Y= Editor ein Gleichungssystem mit Gleichungen erster Ordnung definieren. .

Bei der Eingabe von Anfangsbedingungen ist darauf zu achten, dass für jede Gleichung im System gleich viele Anfangsbedingungen eingegeben werden. Anderenfalls tritt ein Dimension Error auf.



Wenn Sie keine Anfangsbedingung im Y= Editor eingeben

Wenn Sie keine Anfangsbedingungen angeben, bestimmt die Fenstervariable **ncurves** (♦ [WINDOW]), wie viele Lösungskurven automatisch graphisch dargestellt werden. Standardmäßig ist **ncurves = 0**. Es kann ein Wert von 0 bis 10 eingegeben werden. Ob jedoch **ncurves** verwendet wird, hängt vom Graphikformat **Fields** und der Einstellung für **Axes** ab.

Wenn Fields =	Dann:
SLPFLD	Wird **ncurves** , falls nicht auf 0 eingestellt, für die graphische Darstellung der Kurven verwendet.
DIRFLD	Wird **ncurves** ignoriert. Es werden keine Kurven gezeichnet
FLDOFF	Wird **ncurves** verwendet, sofern Axes = TIME (oder wenn Axes = Custom und die x-Achse t ist). Anderenfalls tritt der Fehler Diff Eq setup auf.

Wird **ncurves** verwendet, so wird **t0** kurzzeitig in die Mitte des Graphikbildschirms verschoben. Der im Y= Editor eingestellte Wert von **t0** wird jedoch nicht geändert.

Hinweis:

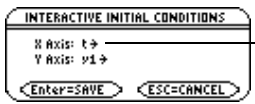
- Um nur ein Steigungs- oder Richtungsfeld anzuzeigen, ohne Anfangsbedingungen einzugeben, verwenden Sie SLPFLD (mit **ncurves=0**) oder DIRFLD.

- SLPFLD gilt nur für eine einzige Gleichung erster Ordnung. DIRFLD gilt nur für eine Gleichung zweiter Ordnung (oder ein System von zwei Gleichungen erster Ordnung).

Eine Anfangsbedingung interaktiv im Graphikbildschirm wählen

Bei der graphischen Darstellung einer Differentialgleichung (unabhängig davon, ob eine Lösungskurve angezeigt wird oder nicht) können Sie im Graphikbildschirm einen Punkt wählen und diesen als Anfangsbedingung verwenden.

Wenn Fields =	Gehen Sie wie folgt vor:
SLPFLD – oder – DIRFLD	Drücken Sie: [2nd] [F8] [F8] Geben Sie eine Anfangsbedingung an. Entweder: • Setzen Sie den Cursor auf den gewünschten Punkt und drücken [ENTER]. – oder – • Sie geben für jede der beiden Koordinaten einen Wert ein und drücken [ENTER]. - Bei SLPFLD (nur 1. Ordnung) geben Sie Werte für **t0** und **y(t0)** ein. - Bei DIRFLD (nur 2. Ordnung oder System von zwei Gleichungen erster Ordnung) geben Sie Werte für beide Anfangsbedingungen **y(t0)** ein, wobei **t0** der im Y= Editor oder Window-Editor eingestellte Wert ist. Die Anfangsbedingung wird mit einem Kreis markiert, und die Lösungskurve wird gezeichnet.

Wenn Fields =	Gehen Sie wie folgt vor:
FLDOFF	• Drücken Sie: [2nd] [F8] [F8] Sie werden aufgefordert, die Achsen zu wählen, für welche Sie Anfangsbedingungen eingeben möchten.  t ist eine gültige Auswahl. Sie können nun einen Wert für t0 angeben. Die gewählten Werte werden als Achsen für den Graphen verwendet. • Sie können die Standardeinstellungen entweder annehmen oder ändern. Drücken Sie dann [ENTER]. • Geben Sie wie für SLPFLD oder DIRFLD beschrieben eine Anfangsbedingung an.

Hinweis: Mit SLPFLD oder DIRFLD können Sie, unabhängig davon, ob Sie im Y= Editor Anfangsbedingungen eingeben, interaktiv Anfangsbedingungen wählen. Bei FLDOFF können Sie die Anfangsbedingungen interaktiv wählen. Wurden jedoch drei oder mehr Gleichungen eingegeben, müssen Sie für jede Gleichung im Y= Editor einzeln einen Wert (keine Liste) als Anfangsbedingung eingeben. Anderenfalls tritt beim Zeichnen ein Dimension-Fehler auf.

Anmerkung zum Tracen einer Lösungskurve

Wenn Sie im Y= Editor Anfangsbedingungen eingeben oder die Lösungskurven mit **ncurves** automatisch graphisch darstellen lassen, können Sie die Kurven mit [F3] tracen.

Kurven, die durch die interaktive Wahl einer Anfangsbedingung gezeichnet wurden, können Sie jedoch nicht verfolgen. Diese Kurven sind nicht geplottet, sondern gezeichnet.

Ein System für Gleichungen höherer Ordnung definieren

Sie müssen alle Differentialgleichungen als Gleichungen erster Ordnung in den Y= Editor eingeben. Eine Gleichung n-ter Ordnung ist in ein System von n Gleichungen erster Ordnung umzuformen.

Umformung einer Gleichung in ein System erster Ordnung

Ein Gleichungssystem kann auf verschiedene Arten definiert werden. Im folgenden wird eine allgemeine Methode dargestellt.

1. Schreiben Sie wenn nötig die ursprüngliche Differentialgleichung um.

   $y'' + y' + y = e^x$

   a) Lösen Sie nach der Ableitung der höchsten Ordnung auf.

   $y'' = e^x - y' - y$

   b) Drücken Sie sie mit y und t aus.

   $y'' = e^t - y' - y$

   c) Substituieren Sie nur auf der rechten Seite, damit keine Bezüge zu Ableitungen bleiben.

   Hinweis: Um eine Gleichung erster Ordnung zu erzeugen darf die rechte Seite nur Variablen enthalten, die nicht zu einer Ableitung gehören.

Ersetzen Sie	Mit:
y	y1
y'	y2
y''	y3
y'''	y4
y(4)	y5
⋮	⋮

$y'' = e^t - y2 - y1$

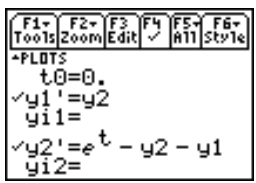
Noch nicht links substituieren.

d) Auf der linken Seite der Gleichung sind die Ableitungen wie folgt zu substituieren.

Ersetzen Sie	Mit:
y’	y1’
y’’	y2’
y’’’	y3’
y(4)	y4’
⋮	⋮

y2' = e^{t} − y2 − y1

2. Definieren Sie das Gleichungssystem in den entsprechenden Zeilen des Y= Editors als:
y1' = y2
y2' = y3
y3' = y4
– bis zu –
yn ' = Ihre Gleichung n-ter Ordnung



Hinweis: Nach obigen Substitutionen stellen die y'-Zeilen im Y= Editor folgendes dar:
y1' = y'
y2' = y''
etc.

Die Gleichung zweiter Ordnung aus diesem Beispiel wird also in die Zeile y2' eingegeben.

In einem solchen System ist die Lösung für die y1'-Gleichung auch die Lösung für die Gleichung n-ter Ordnung. Alle übrigen Gleichungen des Systems sollten Sie abwählen.

Beispiel für eine Gleichung zweiter Ordnung

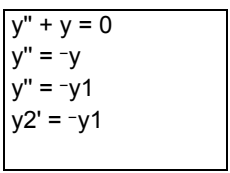
Die Differentialgleichung zweiter Ordnung **y"+y = 0** stellt einen einfachen harmonischen Oszillator dar. Formen Sie diesen in ein System von Gleichungen für den Y= Editor um. Stellen Sie dann für die Anfangsbedingungen **y(0) = 0** und **y'(0) = 1** die Lösungen graphisch dar.

Beispiel

1. Drücken Sie MODE, und nehmen Sie die Einstellung **Graph=DIFF EQUATIONS** vor.

2. Definieren Sie ein Gleichungssystem für die Gleichung zweiter Ordnung.

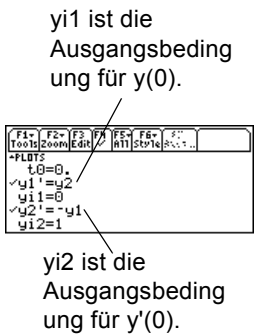
   Schreiben Sie die Gleichung um, und nehmen Sie die erforderlichen Substitutionen vor.



3. Geben Sie im Y= Editor ([♦] [Y=]), das Gleichungssystem ein.

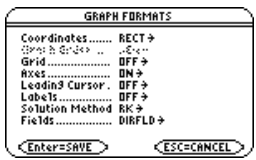
4. Geben Sie die Anfangsbedingungen ein: **yi1=0** und **yi2=1**

   Hinweis: t0 ist der Zeitpunkt, zu welchem die Anfangsbedingungen auftreten. Es ist auch das erste für den Graphen ausgewertete t. Standardmäßig ist **t0=0**.



5. Drücken Sie:
   [F1] **9**
   — oder —
   [♦] [|]
   [♦] **F**
   und nehmen Sie die Einstellungen **Axes = ON**, **Labels = OFF**, **Solution Method = RK** und **Fields = DIRFLD** vor.

   Wichtig: Für Gleichungen zweiter Ordnung muß **Fields** auf **DIRFLD** oder **FLDOFF** eingestellt sein.



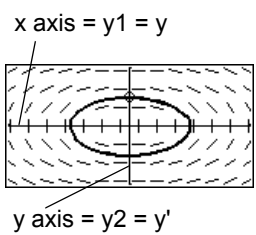
6. Drücken Sie im Y= Editor:
   [2nd] [F7]
   [F7]
   und versichern Sie sich, dass **Axes = CUSTOM** mit **y1** und **y2** als Achsen eingestellt ist.

   Wichtig: Mit **Fields=DIRFLD** kann keine Zeitachse geplottet werden. Wenn **Axes=TIME** oder t als **CUSTOM**-Achse eingestellt ist, tritt der Fehler **Invalid Axes** auf.

7. Stellen Sie im Window-Editor ([♦] [WINDOW]) die Fenstervariablen ein.

t0=0	xmin=⁻2	ncurves=0
tmax=10	xmax=2	diftol=.001
tstep=.1	xscl=1	fldres=14
tplot=0	ymin=⁻2	dtime=0
	ymax=2	
	yscl=1	

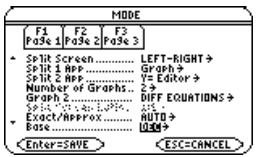
8. Rufen Sie den Graphikbildschirm auf ([♦] [GRAPH]).



Wenn Sie **ZoomSqr** (F2 **5**) wählen, können Sie erkennen, daß das Phasendiagramm eigentlich ein Kreis ist. **ZoomSqr** ändert jedoch Ihre Fenstervariablen.

Zur genaueren Untersuchung dieses harmonischen Oszillators verwenden Sie einen geteilten Bildschirm, um graphisch darzustellen, wie sich y und y' in Abhängigkeit von der Zeit (t) verändern.

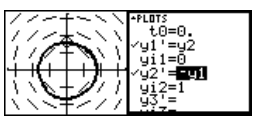
9. Drücken Sie MODE, und ändern Sie die Moduseinstellungen auf **Seite 2** wie hier angegeben. Schließen Sie das Dialogfeld **MODE**. Dadurch wird der Graph erneut gezeichnet.



   Hinweis: Um in jeder Hälfte des geteilten Bildschirms mehrere Graphen anzuzeigen, müssen Sie den Modus **2-Graph** verwenden.

10. Zum Umschalten auf die rechte Hälfte des geteilten Bildschirms drücken Sie 2nd [⊞].

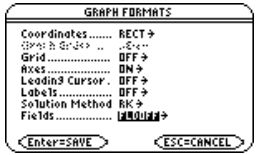
11. Wählen Sie mit F4 **y1'** und **y2'**.



   Die rechte Hälfte verwendet dieselben Gleichungen wie die linke. Zu Beginn sind in der rechten Hälfte jedoch keine Gleichungen gewählt.

12. Drücken Sie:
    F1 **9**
    — oder —
    ♦ |
    ♦ **F**
    und stellen Sie **Fields** auf **FLDOFF** ein.

    Wichtig: Da mit **Fields=DIRFLD** keine Zeitachse geplottet werden kann, muß die Einstellung für Fields geändert werden. **FLDOFF** schaltet alle Felder aus.



13. Drücken Sie im Y= Editor:
    2nd F7
    F7
    und versichern Sie sich, dass **Axes = TIME** eingestellt ist.

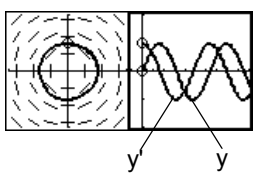
14. Ändern Sie im Window-Editor **ymin** und **ymax** wie hier gezeigt.

    Hinweis: Schalten Sie den Modus **2-Graph** ein, werden die Fenstervariablen für die rechte Hälfte auf ihre Standardeinstellungen gesetzt.

ymin=⁻2.
ymax=2.

15. Drücken Sie ♦ GRAPH, um den Graphikbildschirm für Graph Nr. 2 aufzurufen.

    Die linke Seite zeigt das Phasendiagramm Kreisbahn, die rechte Seite die Lösungskurve und deren Ableitung.



16. Möchten Sie zur Vollbildanzeige des ursprünglichen Graphen zurückkehren, drücken Sie [2nd] [⊟], um auf die linke Hälfte umzuschalten. Drücken Sie dann [MODE], und ändern Sie die Einstellung für **Split Screen**.

Split Screen = FULL

Beispiel für eine Gleichung dritter Ordnung

Geben Sie für die Differentialgleichung dritter Ordnung **y'''+2y''+2y'+y = sin(x)** im Y= Editor ein Gleichungssystem ein. Stellen Sie die Lösung dann als Funktion der Zeit graphisch dar. Verwenden Sie die Anfangsbedingungen **y(0) = 0**, **y'(0) = 1** und **y''(0) = 1**.

Beispiel

1. Drücken Sie [MODE], und nehmen Sie die Einstellung **Graph=DIFF EQUATIONS** vor.

2. Definieren Sie ein Gleichungssystem für die Gleichung dritter Ordnung.

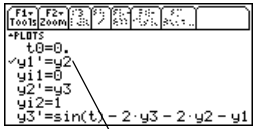
   Schreiben Sie die Gleichung um, und nehmen Sie die erforderlichen Substitutionen vor.

y''' + 2y'' + 2y' + y = sin(x)
y''' = sin(x) − 2y'' − 2y' − y
y''' = sin(t) − 2y'' − 2y' − y
y''' = sin(t) − 2y3 − 2y2 − y1
y3' = sin(t) − 2y3 − 2y2 − y1

3. Geben Sie im Y= Editor ([♦] [Y=]), das Gleichungssystem ein.

4. Geben Sie die Anfangs- bedingungen ein: **yi1=0**, **yi2=1**, und **yi3=1**

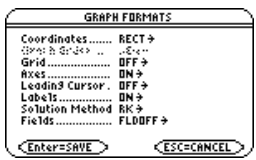
   Hinweis: t0 ist der Zeitpunkt, zu welchem die Anfangsbedingungen auftreten. Standardmäßig ist **t0=0**.



Wichtig: Die Lösung für die Gleichung y1' ist auch die Lösung für die Gleichung dritter Ordnung.

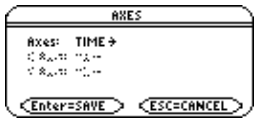
5. Nur **y1'** darf gewählt sein. Heben Sie die etwaige Auswahl anderer Gleichungen mit [F4] auf.

6. Drücken Sie:
   [F1] **9**
   — oder —
   [♦] [|]
   [♦] **F**
   und nehmen Sie die Einstellungen **Axes = ON**, **Labels = ON**, **Solution Method = RK** und **Fields = FLDOFF** vor.



   Wichtig: Für Gleichungen dritter oder höherer Ordnung muß **Fields** auf **FLDOFF** eingestellt sein. Anderenfalls tritt bei der graphischen Darstellung der Fehler **Undefined variable** auf.

7. Drücken Sie im Y= Editor:
   [2nd] [F7]
   [F7]
   und stellen Sie **Axes** auf **TIME** ein.

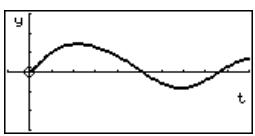


   Hinweis: Mit **Axes=TIME** wird die Lösung für die gewählte Gleichung über der Zeit (t) geplottet.

8. Stellen Sie im Window-Editor ([♦] [WINDOW]) die Fenstervariablen ein.

t0=0	xmin=⁻1	ncurves=0
tmax=10	xmax=10	diftol=.001
tstep=.1	xscl=1.	
tplot=0	ymin=⁻3	
	ymax=3	
	yscl=1	

9. Rufen Sie den Graphikbildschirm auf ([♦] [GRAPH]).



   Hinweis: Möchten Sie die Lösung für einen bestimmten Zeitpunkt finden, tracen Sie den Graphen mit [F3].

Einstellen der Achsen für Zeitplots oder benutzerdefinierte Plots

Die Möglichkeit, die Achsen einzustellen, bietet Ihnen größte Flexibilität beim graphischen Darstellen von Differentialgleichungen. Die benutzerdefinierten Achsen (Custom) sind zum Zeigen verschiedener Beziehungen besonders nützlich.

Das Dialogfeld AXES aufrufen

Drücken Sie im Y= Editor:

[2nd] [F7]

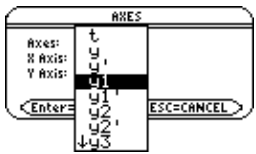
[F7]

Bei **Fields = SLPFLD** ist **Axes** nicht verfügbar.

[2nd] [F7]

[F7]

Menüelement	Beschreibung
Axes	TIME — Plottet t auf der x-Achse und y (Lösungen für alle gewählten Differentialgleichungen) auf der y-Achse. CUSTOM — Dient zum Wählen der x- und y-Achse.

Menüelement	Beschreibung
X Axis, **Y Axis**	Nur aktiv bei Axes = CUSTOM; mit diesen können Sie wählen, was auf der x- und y-Achse geplottet werden soll. 
	t — Zeit
	y — Lösungen (y1, y2 etc.) für alle gewählten Differentialgleichungen
	y' — Werte aller gewählten Differentialgleichungen (y1', y2' etc.)
	y1, y2, etc. — die Lösung für die entsprechende Differentialgleichung, unabhängig davon, ob die Gleichung gewählt ist oder nicht
	y1', y2', etc. — Wert der rechten Seite der entsprechenden Differentialgleichung, unabhängig davon, ob die Gleichung gewählt ist oder nicht

Hinweis: t ist für keine Achse gültig, wenn **Fields=DIRFLD**. Wählen Sie t, tritt bei der graphischen Darstellung der Fehler Invalid axes auf.

Beispiel für Zeitachsen und benutzerdefinierte Achsen

Bestimmen Sie für das Jäger/Beute-Modell aus der Biologie die zur Aufrechterhaltung des Populationsgleichgewichts in einem bestimmtem Gebiet erforderliche Anzahl von Hasen und Füchsen. Stellen Sie die Lösung mit Hilfe von Zeitachsen und benutzerdefinierten Achsen graphisch dar.

Jäger/Beute-Modell

Verwenden Sie die beiden gekoppelten Differentialgleichungen erster Ordnung:

y1' = ⁻y1 + 0.1y1 ∗y2 und y2' = 3y2 −y1 ∗y2

wobei:

y1 = Fuchspopulation

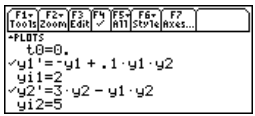
yi1 = Anfängliche Fuchspopulation (2)

y2 = Hasenpopulation

yi2 = Anfängliche Hasenpopulation (5)

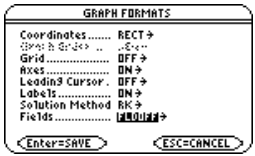
1. Nehmen Sie mit MODE die Einstellung **Graph = DIFF EQUATIONS** vor.

2. Definieren Sie im Y= Editor ([♦] [Y=]) die Differentialgleichungen, und geben Sie die Anfangsbedingungen ein.

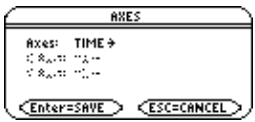


   Hinweis: Zur Verkürzung der Rechenzeit löschen Sie alle anderen Gleichungen im Y= Editor. Mit **FLDOFF** werden alle Gleichungen ausgewertet, auch wenn sie nicht gewählt sind.

3. Drücken Sie:
   [F1] **9**
   — oder —
   [♦] [|]
   [♦] **F**
   und nehmen Sie die Einstellungen **Axes = ON**, **Labels = ON**, **Solution Method = RK** und **Fields = FLDOFF** vor.



4. Drücken Sie im Y= Editor:
   [2nd] [F7]
   [F7]
   und stellen Sie **Axes = TIME** ein.



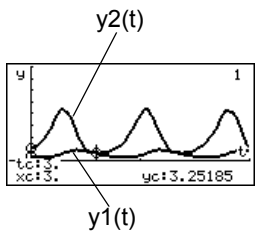
5. Stellen Sie im Window-Editor ([♦] [WINDOW]) die Fenstervariablen ein.

t0=0	xmin=⁻1	ncurves=0
tmax=10	xmax=10	diftol=.001
tstep=π/24	xscl=5	
tplot=0	ymin=⁻10	
	ymax=40	
	yscl=5	

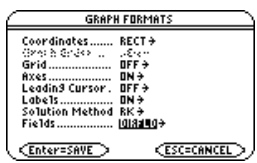
6. Stellen Sie die Differentialgleichungen graphisch dar ([♦] [GRAPH]).

7. Zum Tracen [F3] drücken. Drücken Sie dann **3** [ENTER], um die Fuchs- (**yc** für **y1**) und Hasenanzahl (**yc** für **y2**) bei **t=3** zu sehen.

   Hinweis: Mit ⊙ und ⊙ bewegen Sie den Trace-Cursor zwischen den Kurven für **y1** und **y2** hin und her.



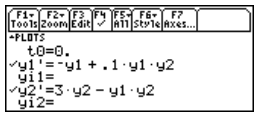
8. Kehren Sie zum Y= Editor zurück, drücken Sie:
   [F1] **9**
   — oder —
   [♦] [|]
   [♦] **F**
   und stellen Sie **Fields = DIRFLD** ein.



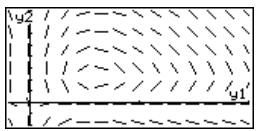
   Hinweis: In diesem Beispiel wird **DIRFLD** für zwei gekoppelte Differentialgleichungen verwendet, die keine Gleichung zweiter Ordnung darstellen.

9. Drücken Sie:
   [2nd] [F7]
   [F7]
   und vergewissern Sie sich, dass die Achsen wie gezeigt eingestellt sind.

10. Löschen Sie im Y= Editor die Anfangsbedingungen für **yi1** und **yi2**.



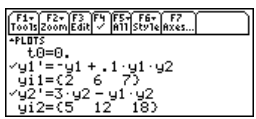
11. Kehren Sie zurück zum Graphikbildschirm, der nur das Richtungsfeld anzeigt.



12. Soll eine Lösungsfamilie graphisch dargestellt werden, kehren Sie zum Y= Editor zurück und geben die folgenden Anfangsbedingungen ein.

    yi1={2,6,7} und **yi2={5,12,18}**

    Hinweis: Mehrere Anfangsbedingungen geben Sie in Form einer Liste an.

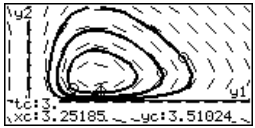


13. Rufen Sie wieder den Graphikbildschirm auf, der für jedes Anfangsbedingungs-Paar eine Kurve anzeigt.

14. Zum Tracen F3 drücken. Drücken Sie dann **3** ENTER, um die Fuchs- (**xc**) und Hasenanzahl (**yc**) bei **t=3** zu sehen.

    Da **t0=0** und **tmax=10**, können Sie innerhalb des Bereichs $0 \le t \le 10$ tracen.

    Hinweis: Mit ▲ und ▼ bewegen Sie den Trace-Cursor zwischen den Anfangsbedingungs-Kurven hin und her.



Beispielvergleich zwischen RK und Euler

Wir betrachten ein logistisches Wachstumsmodell **dP/dt = .001∗P∗(100−P)** mit der Anfangsbedingung **P(0) = 10**. Vergleichen Sie die durch die Lösungsverfahren RK und Euler berechneten Graph-Punkte mit der neuen Anweisung **BldData**. Plotten Sie diese Punkte dann mit einem Graphen für die exakte Lösung der Gleichung.

Beispiel

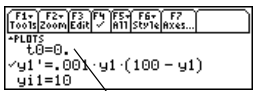
1. Drücken Sie MODE, und nehmen Sie die Einstellung **Graph=DIFF EQUATIONS** vor.

2. Drücken Sie die Gleichung erster Ordnung mit **y1'** und **y1** aus.

y1'=.001y1∗(100−y1)

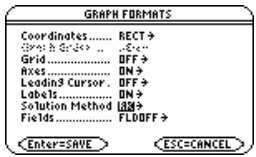
Verwenden Sie keine implizite Multiplikation zwischen Variablen und Klammern, sonst wird die Eingabe als Funktionsaufruf behandelt.

3. Geben Sie im Y= Editor ([♦] [Y=]) die Gleichung ein.
4. Geben Sie folgende Anfangsbedingung ein:
   yi1=10



t0 ist der Zeitpunkt, zu welchem die Anfangsbedingung auftritt. Standardmäßig ist t0=0.

5. Drücken Sie:
   F1 **9**
   — oder —
   ♦ |
   ♦ **F**
   und nehmen Sie die Einstellungen **Solution Method = RK** und **Fields = FLDOFF** vor.



   Hinweis: Zur Verkürzung der Rechenzeit löschen Sie alle anderen Gleichungen im Y= Editor. Mit **FLDOFF** werden alle Gleichungen ausgewertet, auch wenn sie nicht gewählt sind.

6. Stellen Sie im Window-Editor (♦ [WINDOW]) die Fenstervariablen ein.

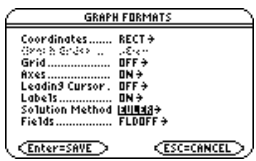
	t0=0.	xmin=⁻1.	ncurves=0.
	tmax=100.	xmax=100.	diftol=.001
❶	tstep=1.	xscl=1.	
	tplot=0.	ymin=⁻10.	
		ymax=10	
		yscl=1.	

❶ **Wichtig:** Ändern Sie **tstep** von .1 (Standardwert) auf 1. Anderenfalls berechnet **BldData** zu viele Zeilen für die Datenvariable, und ein Dimension-Fehler tritt auf.

7. Erstellen Sie mit **BldData** im Hauptbildschirm
   [HOME]
   [◆] [CALC HOME]
   eine Datenvariable, welche die **RK**-Graph-Punkte enthält.

   BldData rklog

8. Kehren Sie zurück zum Y= Editor, drücken Sie:
   [F1] **9**
   — oder —
   [◆] [|]
   [◆] **F**
   und stellen Sie **Solution Method = EULER** ein.

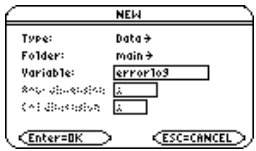
   Hinweis: Sie müssen die Gleichung nicht graphisch darstellen, bevor Sie **BldData** verwenden. Näheres zu **BldData** finden Sie im Modul *Technische Referenz*.

9. Kehren Sie zurück zum Hauptbildschirm, und erstellen Sie mit **BldData** eine Datenvariable, welche die **Euler**-Graph-Punkte enthält.

   BldData eulerlog

10. Verwenden Sie den **Daten/Matrix-Editor** (APPS), um eine neue Datenvariable namens **errorlog** zu erstellen.

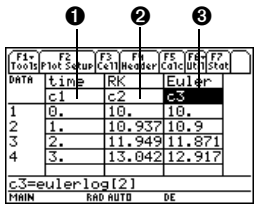
    Hinweis: Mit **errorlog** können Sie die Daten in **rklog** und **eulerlog** kombinieren und die beiden Datensätze gegenüberstellen.



11. Definieren Sie in dieser neuen Datenvariablen die Kopfzeilen der Spalten **c1**, **c2** und **c3**, um Bezüge zu den Daten in **rklog** und **eulerlog** herzustellen. Geben Sie auch wie gezeigt Spaltentitel ein.

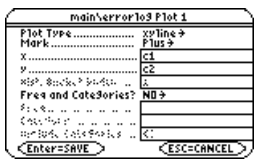
    Zum Definieren einer Kopfzeile setzen Sie den Cursor auf die entsprechende Spalte, drücken F4, geben den Bezugsterm ein (z.B. **rklog[1]** für **c1**) und drücken ENTER.

    Hinweis: rklog[1] und **rklog[2]** beziehen sich auf Spalte 1 bzw. 2 in **rklog**. Gleiches gilt für **eulerlog[2]**.



❶ c1=rklog[1] oder c1=eulerlog[1]
❷ c2=rklog[2]
❸ c3= eulerlog[2]

12. Drücken Sie im **Daten/Matrix-Editor** F2. Drücken Sie dann F1, und definieren Sie, wie rechts gezeigt, **Plot 1** für die **RK**-Daten.



13. Definieren Sie **Plot 2** für die **Euler**-Daten. Verwenden Sie nebenstehende Werte.

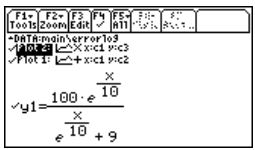
Plot Type=xyline
Mark=Cross
x=c1
y=c3

14. Kehren Sie zurück zum Y= Editor, drücken Sie MODE und stellen Sie **Graph = FUNCTION** ein.

15. Folgendes ist die exakte Lösung für die Differentialgleichung. Geben Sie diese als y1 ein.

    y1 = (100∗e^(x/10))/(e^(x/10)+9)

    Hinweis: Sie können deSolve( ) verwenden, um diese exakte allgemeine Lösung zu finden.

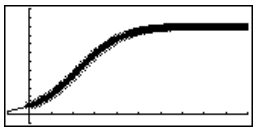


Mit ⊙ können Sie das Bild nach oben scrollen, um Plot 1 und Plot 2 zu betrachten.

16. Stellen Sie im Window-Editor die Fenstervariablen ein.

xmin=⁻10	ymin=⁻10.	xres=2.
xmax=100	ymax=120.	
xscl=10	yscl=10.	

17. Rufen Sie den Graphikbildschirm auf ([♦] [GRAPH]).



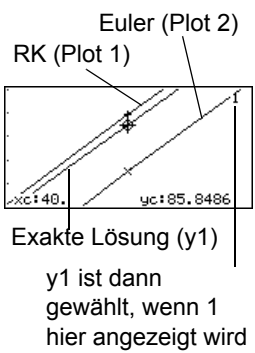
Hinweis: Die unscharfe Linie auf dem Graphen zeigt die Unterschiede zwischen den **RK**- und **Euler**-Werten auf.

18. Stellen Sie im Window-Editor die Fenstervariablen zum Vergrößern ein, so dass Sie die Unterschiede detaillierter untersuchen können.

xmin=39.7	ymin=85.5	xres=2
xmax=40.3	ymax=86	
xscl=.1	yscl=.1	

19. Rufen Sie wieder den Graphikbildschirm auf.

20. Drücken Sie zum Tracen [F3] und dann so lange ▲ oder ▼, bis **y1** gewählt ist (in der oberen rechten Ecke wird 1 angezeigt). Geben Sie dann 40 ein.



Wenn Sie mit dem Trace-Cursor jede Lösung für **xc = 40** verfolgen, stellen Sie folgendes fest:

- Die exakte Lösung (**y1**) ist 85,8486, auf sechs Stellen gerundet.
- Die **RK**-Lösung (**Plot 1**) ist 85,8952.
- Die **Euler**-Lösung (**Plot 2**) ist 85,6527.

Sie können auch im **Daten/Matrix-Editor** die Datenvariable errorlog öffnen und zu **time = 40** scrollen.

Beispiel für die Funktion deSolve( )

Mit der Funktion **deSolve( )** können Sie viele gewöhnliche Differentialgleichungen erster und zweiter Ordnung exakt lösen.

Beispiel

Für eine allgemeine Lösung können Sie folgende Syntax verwenden. Wie Sie für eine bestimmte Lösung vorgehen, erfahren Sie im Modul *Technische Referenz*.

deSolve(*DG1oder2Ordnung, unabhängigeVar, abhängigeVar***)**

Finden Sie für die logistische Differentialgleichung erster Ordnung aus dem Beispiel auf die allgemeine Lösung für y bezüglich t.

deSolve(y' = 1/1000 y∗(100−y),t,y**)**

Zur Eingabe von ', tippen Sie 2nd ['].

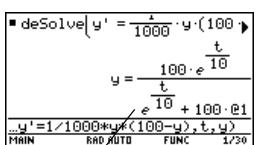
Verwenden Sie keine implizite Multiplikation zwischen der Variablen und Klammern, sonst wird die Eingabe als Funktionsaufruf behandelt.

Hinweis:

- Für eine maximale Genauigkeit verwenden Sie 1/1000 anstelle von ,001. Eine Fließkommazahl kann zu Rundungsfehlern führen.
- In diesem Beispiel wird keine graphische Darstellung durchgeführt, daher können Sie jeden Graph-Modus verwenden.

Löschen Sie vor der Verwendung von **deSolve( )** alle vorhandenen t-und y-Variablen. Anderenfalls tritt ein Fehler auf.

1. Verwenden Sie **deSolve( )** im Hauptbildschirm
   HOME
   ◆ [CALC HOME]
   um die allgemeine Lösung zu finden.



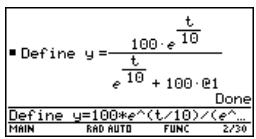
@1 stellt eine Konstante dar. Sie erhalten möglicherweise eine andere Konstante (@2 etc.).

2. Definieren Sie mit der Lösung eine Funktion.

   a) Drücken Sie ⊙ zum Markieren der Lösung im History-Bereich. Drücken Sie dann [ENTER], um sie automatisch in die Eingabezeile einzufügen.

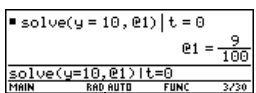
   b) Setzen Sie die Anweisung **Define** am Zeilenanfang ein. Drücken Sie dann [ENTER].

   Hinweis: Drücken Sie [2nd] ◀, um den Cursor zum Anfang der Eingabezeile zu bewegen.



3. Für die Anfangsbedingung **y=10** mit **t=0** finden Sie die @1-Konstante mit **solve( )**.

   Hinweis: Haben Sie eine andere Konstante erhalten (@2 etc.), lösen Sie nach dieser auf.

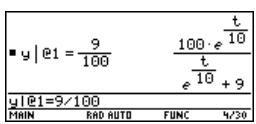


Zur Eingabe von @ tippen Sie

[Taschenrechner] [♦] [STO▶]

[Tastatur] [2nd] **R**

4. Werten Sie die allgemeine Lösung (**y**) mit der Konstanten **@1=9/100** aus, um die hier gezeigte bestimmte Lösung zu erhalten.



Mit **deSolve( )** können Sie die Aufgabe auch direkt lösen. Geben Sie folgendes ein:

deSolve(y' = 1/1000 y∗(100−y) und y(0)=10,t,y**)**

Fehlersuche bei Verwendung des Feld-Graphikformats

Dieser Abschnitt hilft Ihnen, etwaige Schwierigkeiten bei der graphischen Darstellung von Differentialgleichungen zu beheben. Viele dieser Probleme können mit der Einstellung des Graphikformats Fields zusammenhängen.

Einstellung des Graphikformats Fields

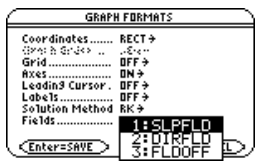
Drücken Sie im Y= Editor, Window-Editor oder Graphikbildschirm:

F1 **9**

— oder —

◆ I

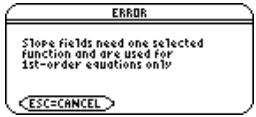
◆ **F**



Welcher Ordnung ist die darzustellende Gleichung?

Handelt es sich um eine Gleichung der:	Sind folgende Einstellungen für Fields zulässig:
1. Ordnung	SLPFLD oder FLDOFF
2. Ordnung (System von zwei Gleichungen erster Ordnung)	DIRFLD oder FLDOFF
3. Oder einer höheren Ordnung (System von drei oder mehr Gleichungen erster Ordnung)	FLDOFF

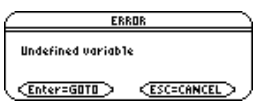
Da **Fields = SLPFLD** die Standardeinstellung ist, wird auf der rechten Seite eine gewöhnliche Fehlermeldung angezeigt.

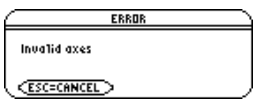
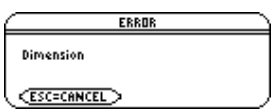


Wenn diese oder eine andere Fehlermeldung angezeigt wird:

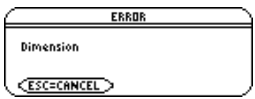
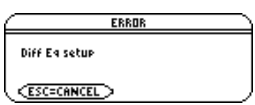
- Ermitteln Sie anhand obiger Tabelle die gültigen Fields-Einstellungen für die Ordnung Ihrer Gleichung, und ändern Sie Ihre Einstellung dementsprechend.
- Lesen Sie die nachfolgenden Informationen für die entsprechende Fields-Einstellung.

Fields=SLPFLD

Im Y= Editor	Wählen Sie mit F4 nur eine einzige Gleichung erster Ordnung. Sie können mehrere Gleichungen eingeben, es kann aber nur jeweils eine gewählt werden. Die gewählte Gleichung darf sich auf keine andere Gleichung im Y= Editor beziehen. Beispiel: Wenn **y1'=y2**, tritt bei der graphischen Darstellung der Fehler Undefined variable auf. 
Im Graphik-bildschirm	Wenn zwar das Steigungsfeld gezeichnet, aber keine Lösungskurve geplottet wird, geben Sie eine Anfangsbedingung an.

Im Y= Editor	Geben Sie ein gültiges System von zwei Gleichungen erster Ordnung ein. Näheres zum Definieren eines gültigen Systems für eine Gleichung zweiter Ordnung finden Sie unter Beispiel für eine Gleichung zweiter Ordnung. Definieren Sie die Achsen wie folgt: **CUSTOM**: 2nd F7 F7 Bei **Axes = TIME** erscheint beim Zeichnen eine Fehlermeldung.  Wenn Sie im Y= Editor Anfangsbedingungen eingeben, so müssen die Gleichungen, auf welche sich die benutzerdefinierten Achsen beziehen, gleich viele Anfangsbedingungen haben. Anderenfalls tritt beim Zeichnen Darstellung ein Dimension Error auf. 

Mit benutzer-definierten Achsen	Stellen Sie für Ihr Gleichungssystem gültige Achsen ein. Wählen Sie für keine Achse t, sonst tritt beim Zeichnen der Fehler Invalid axes auf. Die beiden Achsen müssen sich auf verschiedene Gleichungen in Ihrem Gleichungssystem beziehen. y1 gegen y2 ist zum Beispiel gültig, während y1 gegen y1' einen Invalid axes-Fehler verursacht.
Im Graphik-bildschirm	Wird zwar das Richtungsfeld gezeichnet, aber keine Kurve geplottet, geben Sie im Y= Editor Anfangsbedingungen ein, oder wählen Sie die Anfangsbedingungen im Graphikbildschirm interaktiv. Haben Sie bereits Anfangsbedingungen eingegeben, wählen Sie **ZoomFit**: F2 alpha **A** F2 **A** Die Fenstervariable **ncurves** wird bei DIRFLD ignoriert. Standardkurven werden nicht automatisch gezeichnet.
Hinweise	Bei DIRFLD bestimmen die Gleichungen, auf welche sich die benutzerdefinierten Achsen beziehen, welche Gleichungen graphisch dargestellt werden, unabhängig davon, welche Gleichungen im Y= Editor gewählt sind. Bezieht sich Ihr Gleichungssystem auf t, so wird das Richtungsfeld (nicht die geplotteten Kurven) bezüglich eines bestimmten Zeitpunkts gezeichnet, der durch die Fenstervariable dtime bestimmt ist.

Im Y= Editor	Geben Sie eine Gleichung zweiter oder höherer Ordnung stets als gültiges Gleichungssystem ein, wie es auf beschrieben wurde. Alle Gleichungen (gewählt oder nicht) müssen gleich viele Anfangsbedingungen aufweisen. Anderenfalls tritt bei der graphischen Darstellung ein Dimension Error auf.  **Axes = TIME** oder **CUSTOM** wird wie folgt definiert: 2nd [F7] F7
Mit benutzer-definierten Achsen	Wenn X Axis nicht t ist, müssen Sie für jede Gleichung im Y= Editor (unabhängig davon, ob die Gleichungen gewählt sind oder nicht) mindestens eine Anfangsbedingung eingeben. Anderenfalls tritt bei der graphischen Darstellung der Fehler Diff Eq setup auf. 

Im Graphik-bildschirm	Wird keine Kurve gezeichnet, geben Sie eine Anfangsbedingung ein. Haben Sie bereits im Y= Editor Anfangsbedingungen eingegeben, wählen Sie **ZoomFit**: F2 alpha **A** F2 **A** Mit FLDOFF kann eine Gleichung erster Ordnung anders aussehen als mit SLPFLD, da FLDOFF die Fenstervariablen **tplot** und **tmax** (page 9) verwendet, welche mit SLPFLD ignoriert werden.
Hinweise	Verwenden Sie für Gleichungen erster Ordnung FLDOFF und Axes = Custom, um Achsen zu plotten, die mit SLPFLD nicht geplottet werden können. Sie können zum Beispiel t gegen y1' plotten (wobei SLPFLD t gegen y1 plottet). Bei der Eingabe mehrerer Gleichungen erster Ordnung können Sie eine Gleichung oder ihre Lösung gegen eine andere plotten, indem Sie diese als Achsen definieren.

Wenn Sie Differentialgleichungen im Tabellenbildschirm anzeigen

Im Tabellenbildschirm können die Punkte für den Graphen einer Differentialgleichung angezeigt werden. Die Tabelle zeigt aber möglicherweise andere Gleichungen als die graphisch dargestellten. Unabhängig davon, ob die Gleichungen mit den aktuellen Einstellungen für **Fields** und **Axes** geplottet werden können, zeigt sie lediglich die gewählten Gleichungen an.

Tabellen

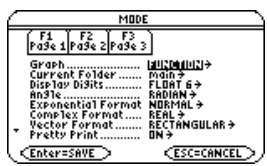
Schritte zum Erstellen einer Tabelle

Zum Erstellen einer Tabelle mit Werten für eine oder mehrere Funktionen gehen Sie gemäß den im folgenden dargestellten Schritten vor. Spezifische Angaben zur Einstellung der Tabellenparameter und zur Anzeige der Tabelle finden Sie auf den nachfolgenden Seiten.

Eine Tabelle erstellen

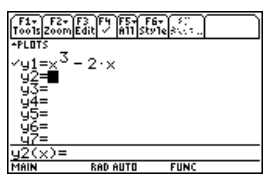
1. Stellen Sie **Graph**-Modus und, wenn nötig, **Angle** Modus ein (MODE).

   Hinweis: Im **3D Graph**-Modus sind Tabellen nicht verfügbar.



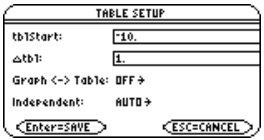
2. Bestimmen Sie Funktionen im Y= Editor ([◆] [Y=]).
3. Wählen Sie (F4), welche definierten Funktionen in der Tabelle angezeigt werden sollen.

   Hinweis: Angaben zum Definieren und Auswählen von Funktionen anhand des Y= Editors finden Sie in *Grundlagen der graphischen Darstellung van Funktionen*.

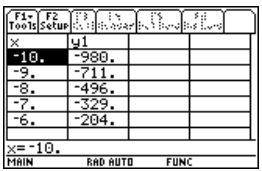


4. Stellen Sie die Anfangsparameter für die Tabelle ein (♦ [TBLSET]).

   Hinweis: Sie können eine automatische Tabelle angeben, die auf Startwerten basiert oder einer Grafik bzw. manuellen (abgefragten) Tabelle entspricht.



5. Zeigen Sie die Tabelle an (♦ [TABLE]).



Die Tabelle untersuchen

Im Tabellen-Bildschirm können Sie:

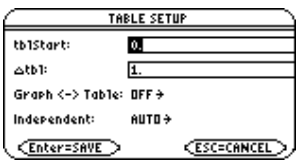
- Die Tabelle durchlaufen, um sich Werte auf anderen Seiten anzusehen.
- Eine Zelle markieren, um ihren vollständigen Wert anzuzeigen.
- Die Setup-Parameter der Tabelle ändern. Durch Änderung des Anfangs- oder Inkrementwertes für die unabhängige Variable können Sie die Tabelle in einzelnen Punkten genauer, oder über größere Bereiche umfassender betrachten, und dadurch unterschiedliche Detail-Ebenen anschauen.
- Die Zellenbreite ändern.
- Gewählte Funktionen bearbeiten.
- Eine manuelle Tabelle erstellen oder bearbeiten, in der nur einzeln spezifierte Werte angezeigt werden.

Die Tabellenparameter einrichten

Zum Einrichten der Anfangsparameter für eine Tabelle verwenden Sie das **TABLE SETUP**-Dialogfeld. Nach Anzeige der Tabelle können Sie die Parameter mit Hilfe dieses Dialogfelds auch ändern.

Das TABLE SETUP-Dialogfeld öffnen

Drücken Sie zum Öffnen des TABLE SETUP-Dialogfelds ♦ [TBLSET]. Im Tabellen-Bildschirm können Sie auch F2 drücken.



Setup-Parameter	Beschreibung
tblStart	Bei **Independent = AUTO** und **Graph <‑> Table = OFF** wird hierdurch der Anfangswert für die unabhängige Variable bestimmt.
Δtbl	Bei **Independent = AUTO** und **Graph <‑> Table = OFF** wird hierdurch der Inkrementwert für die unabhängige Variable bestimmt. **Δtbl** kann positiv oder negativ, aber nicht Null sein.

Setup-Parameter	Beschreibung
Graph < - > Table	Bei **Independent = AUTO**: OFF — Die Tabelle verwendet die für **tblStart** und **Δtbl** eingegebenen Werte. ON — Die Tabelle verwendet die unabhänigen Variablen, die für die graphische Darstellung der Funktionen auf dem Graphikbildschirm verwendet werden. Diese Werte hängen von den im Window-Editor eingestellten Window-Variablen sowie von der Größe des geteilten Bildschirms (*Geteilte Bildschirme*) ab.
Independent	AUTO — Der TI-89 / Voyage™ 200 erstellt auf Grundlage von **tblStart**, **Δtbl** und **Graph < - > Table** automatisch eine Wertefolge für die unabhängige Variable. ASK — Dient zum manuellen Erstellen einer Tabelle durch Eingabe spezifischer Werte für die unabhängige Variable.

Hinweis: Die Tabelle beginnt anfangs bei **tblStart**; Sie können aber mit ⊙ davorliegende Werte erreichen.

Einsatz der verschiedenen Setup-Parameter

Zu erstellende Tabelle	tblStart	Δtbl	Graph < - > Table	Independent
Automatische Tabelle				
• Auf Grundlage von Anfangswerten	Wert	Wert	**OFF**	**AUTO**

Zu erstellende Tabelle	tblStart	Δtbl	Graph <–> Table	Independent
• In Anpassung an den Graphikbildschirm	–	–	**ON**	**AUTO**
Manuelle Tabelle	–	–	**–**	**ASK**

Hinweis: "—" bedeutet, dass ein für diesen Parameter eingegebener Wert bei der angegebenen Tabellenart nicht berücksichtigt wird.

Verwenden Sie im **SEQUENCE**-Graphik-Modus für **tblStart** und Δ**tbl** ganze Zahlen.

Die Setup-Parameter ändern

Ausgangspunkt ist das **TABLE SETUP**-Dialogfeld:

1. Markieren Sie mit ⊙ und ⊙ den Wert oder die Einstellung, den/die Sie ändern möchten.

2. Geben Sie den neuen Wert / die neue Einstellung an.

Ändern	Vorgehensweise
tblStart oder **Δtbl**	Geben Sie den neuen Wert ein. Der bestehende Wert wird gelöscht, sobald sie schreiben. — oder — Drücken Sie ◀ oder ▶, um die Markierung zu entfernen. Bearbeiten Sie dann den bestehenden Wert.
Graph <–> Table oder **Independent**	Drücken Sie ◀ oder ▶, um ein Menü mit gültigen Einstellungen zu öffnen. Dann: • Bewegen Sie den Cursor zum Markieren der Einstellung, und drücken Sie ENTER. — oder — • Drücken Sie die Nummer dieser Einstellung.

Hinweis: Möchten Sie ein Menü schließen oder das Dialogfeld verlassen, ohne Änderungen zu speichern, drücken Sie ESC anstatt ENTER.

3. Wenn Sie alle entsprechenden Werte oder Einstellungen geändert haben, drücken Sie ENTER, um Ihre Änderungen zu speichern, und schließen Sie das Dialogfeld.

Im Hauptbildschirm oder in einem Programm

Sie können die Parameter einer Tabelle vom Hauptbildschirm oder von einem Programm aus einstellen. Sie können:

- Werte direkt in den Systemvariablen **tblStart** und **Δtbl** speichern. Siehe “Variablenwerte speichern und abrufen” in *Bedienung des Taschenrechners*.

- **Graph <‑> Table** und Independent anhand der **setTable**-Funktion einstellen. Siehe Modul *Technische Referenz*.

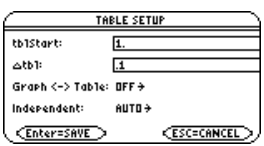
Eine automatische Tabelle anzeigen

Wenn im **TABLE SETUP**-Dialogfeld **Independent = AUTO**, dann wird beim Öffnen des Tabellen-Bildschirms automatisch eine Tabelle erstellt. Wenn **Graph <‑> Table = ON**, paßt sich die Tabelle an die Parameter des Graphikbildschirms an. Wenn **Graph <‑> Table = OFF**, beruht die Tabelle auf den für **tblStart** und **Δtbl** eingegebenen Werten.

Vorbereitungen

Definieren und wählen Sie die gewünschten Funktionen im Y= Editor ([♦] [Y=]). In diesem Beispiel wird **y1(x) = x^3−x/3** verwendet.

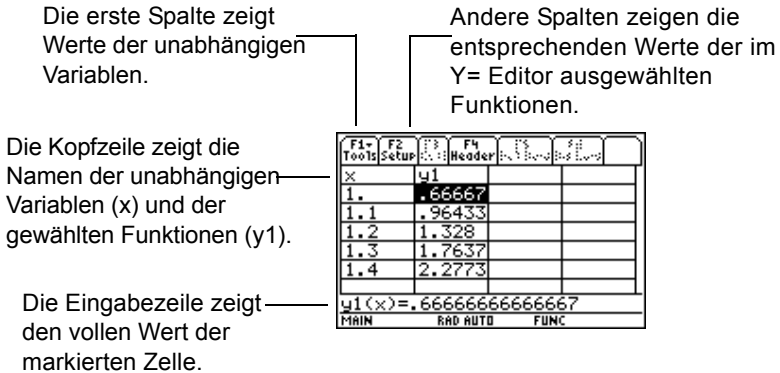
Geben Sie dann die Anfangsparameter für die Tabelle ein ([♦] [TBLSET]).



Den Tabellen-Bildschirm anzeigen

Drücken Sie zum Öffnen des Tabellen-Bildschirms [♦] [TABLE] oder [APPS] **5**.

Der Cursor markiert zunächst die Zelle mit dem Anfangswert der unabhängigen Variablen. Sie können den Cursor auf jede Zelle setzen, die einen Wert enthält.



Hinweis: Vom Anfangswert aus können Sie mit ⊙ oder 2nd ⊙ rückwärts scrollen.

Bewegung des Cursors:	Drücken Sie
Um je eine Zelle	⊙, ⊙, ⊙, oder ⊙
Um je eine Seite	2nd und dann ⊙, ⊙, ⊙, oder ⊙

Sowohl die Kopfzeile als auch die erste Spalte sind feststehend und können nicht über den Bildschirmrand hinausscrollen.

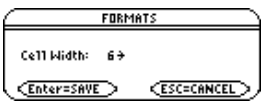
- Beim Aufwärts- oder Abwärtsscrollen sind die Variablen- und Funktionsnamen stets am oberen Bildschirmrand sichtbar.
- Beim Scrollen nach rechts oder links sind die Werte der unabhängigen Variablen stets am linken Bildschirmrand sichtbar.

Die Zellenbreite ändern

Die Zellenbreite bestimmt, wieviele Stellen und Symbole (Komma, Minuszeichen und "E" für wissenschaftliche Schreibweise) maximal in einer Zelle angezeigt werden können. Alle Zellen einer Tabelle haben dieselbe Breite.

Hinweis: Standardeinstellung für die Zellenbreite ist **6**.

So ändern Sie die Zellenbreite im **Tabellen**-Bildschirm:



1. Press F1 **9**
   — oder —
   [calculator] ◆ |
   [keyboard] ◆ **F**
2. Drücken Sie ▶ oder ◀, um ein Menü mit gültigen Breiten zu öffnen (**3–12**).
3. Bewegen Sie den Cursor, um eine Zahl zu markieren, und drücken Sie dann ENTER. (Bei einstelligen Zahlen können Sie die Zahl eingeben und ENTER drücken).
4. Drücken Sie ENTER zum Schließen des Dialogfelds und zum Aktualisieren der Tabelle.

Wie Zahlen in einer Zelle angezeigt werden

Zahlen werden, wo dies möglich ist, gemäß dem aktuell gewählten Anzeigemodus wiedergegeben (Display Digits, Exponential Format etc.). Nötigenfalls wird die Zahl gerundet. Beachten Sie jedoch:

- Ist der Absolutwert einer Zahl zu groß für die aktuelle Zellenbreite, wird die Zahl gerundet und in wissenschaftlicher Schreibweise wiedergegeben.
- Ist die Zelle sogar für die wissenschaftliche Schreibweise zu schmal, wird "..." angezeigt.

Hinweis:

- Ist eine Funktion in einem bestimmten Wert undefiniert, wird in der Zelle undef angezeigt.

- Stellen Sie die Anzeigemodi anhand von [MODE] ein.

Die Standardeinstellung ist **Display Digits = FLOAT 6**. Bei dieser Moduseinstellung werden bis zu sechs Stellen einer Zahl angezeigt, auch wenn die Zellenbreite die Anzeige weiterer Stellen zulassen würde. Andere Einstellungen haben eine dementsprechende Auswirkung auf die Anzeige einer Zahl.

	Zellenbreite			
Volle Präzision	**3**	**6**	**9**	**12**
1.2345678901	1.2	1.2346	1.23457	1.23457*
-123456.78	...	-1.2E5	-123457.	-123457.*
.000005	...	5.E 6	.000005	.000005
1.2345678E19	...	1.2E19	1.2346E19	1.23457E19*
-1.23456789012E 200	...	...	-1.2E 200	-1.2346E 200

***Hinweis:** Je nach Anzeigenmodus-Einstellung werden einige Werte selbst bei ausreichender Zellenbreite nicht in voller Präzision angezeigt.

Hinweis: Die Zahl in voller Präzision sehen Sie in der Eingabezeile, wenn Sie die entsprechende Zelle markieren.

Bei komplexen Zahlen als Ergebnis

In einer Zelle wird eine komplexe Zahl so weit wie möglich wiedergegeben (je nach aktuellem Anzeigemodus), und "..." steht hinter dem angezeigten Teil für den Rest der Zahl.

Markieren Sie eine Zelle, in welcher sich eine komplexe Zahl befindet, wird in der Eingabezeile der Real- und der Imaginärteil mit jeweils bis zu vier Stellen angezeigt (FLOAT 4).

Eine gewählte Funktion bearbeiten

Sie können eine gewählte Funktion von der Tabelle aus ändern, ohne dass Sie auf den Y= Editor zurückgreifen müssen.

1. Setzen Sie den Cursor auf eine beliebige Zelle in der Spalte der betreffenden Funktion. In der Kopfzeile der Tabelle werden die Funktionsnamen angezeigt (**y1** etc.).
2. Drücken Sie F4, um den Cursor auf die Eingabezeile zu setzen, wo die Funktion angezeigt und markiert wird.

   Hinweis: Mit dieser Gerätefunktion können Sie sich eine Funktion ansehen, ohne die Tabelle verlassen zu müssen.
3. Nehmen Sie die erforderlichen Änderungen vor.

- Geben Sie die neue Funktion ein. Die alte Funktion wird gelöscht, sobald Sie die Eingabe beginnen.

  — oder —

- Drücken Sie [CLEAR], um die alte Funktion zu löschen. Geben Sie dann die neue ein.

  — oder —

- Drücken Sie ◄ oder ► zum Entfernen der Markierung. Bearbeiten Sie dann die Funktion.

Hinweis: Möchten Sie etwaige Änderungen annullieren und mit dem Cursor zur Tabelle zurückkehren, drücken Sie [ESC] anstatt [ENTER].

4. Drücken Sie [ENTER], um die bearbeitete Funktion zu speichern und die Tabelle zu aktualisieren. Die bearbeitete Funktion wird auch im Y= Editor gespeichert.

Die Setup-Parameter sollen geändert werden

Nach der Erstellung einer automatischen Tabelle können Sie bei Bedarf die Setup-Parameter ändern.

Drücken Sie [F2] oder [♦] [TBLSET], um das TABLE SETUP-Dialogfeld zu öffnen. Nehmen Sie dann Ihre Änderungen vor.

Eine manuelle (Ask-) Tabelle erstellen

Wenn im TABLE SETUP-Dialogfeld **Independent = ASK**, dann können Sie mit dem TI-89 / Voyage™ 200 durch die Eingabe einzelner Werte für die unabhängige Variable eine Tabelle manuell erstellen.

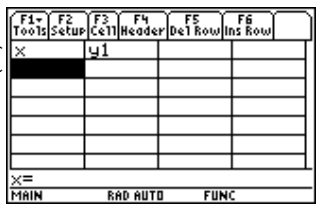
Den Tabellen-Bildschirm öffnen

Drücken Sie ♦ [TABLE], um den Tabellen-Bildschirm zu öffnen.

Bei der Einstellung **Independent = ASK** (durch ♦ [TBLSET]) wird vor der ersten Anzeige einer Tabelle ein leerer Bildschirm geöffnet. Der Cursor markiert die erste Zelle in der Spalte der unabhängigen Variablen.

Die Kopfzeile zeigt die Namen der unabhängigen Variablen (x) und der gewählten Funktionen (y1).

Geben Sie hier einen Wert ein.



Zeigen Sie zunächst eine automatische Tabelle an und ändern sie dann in **Independent = ASK**, enthält die Tabelle weiterhin dieselben Werte. Sie können jedoch keine zusätzlichen Werte mehr durch das Auf- oder Abwärtsscrollen über den Bildschirmrand hinaus erhalten.

Einen Wert für eine unabhängige Variable eingeben oder bearbeiten

Einen Wert können Sie nur in Spalte 1 (unabhängige Variable) eingeben.

1. Bewegen Sie den Cursor, um die Zelle für die Eingabe oder Bearbeitung zu markieren.
   - Beginnen Sie mit einer leeren Tabelle, können Sie die Werte ausschließlich in aufeinanderfolgende Zellen eingeben (Zeile 1, Zeile 2 etc.). Sie können Zellen nicht überspringen (Zeile 1, Zeile 3).

- Enthält eine Zelle in Spalte 1 bereits einen Wert, können Sie diesen bearbeiten.

2. Drücken Sie [F3], um den Cursor auf die Eingabezeile zu setzen.
3. Geben Sie einen neuen Wert oder Term ein, oder bearbeiten Sie den vorhandenen Wert.
4. Drücken Sie [ENTER], um den Wert in die Tabelle zu verschieben und die entsprechenden Funktionswerte zu aktualisieren.

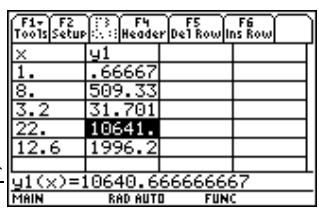
   Hinweis: Es ist nicht erforderlich, [F3] zu drücken, um einen neuen Wert in eine Zelle einzugeben. Beginnen Sie einfach die Eingabe.

Der Cursor kehrt zur Zelle mit der neuen Eingabe zurück. Mit ⊛ können Sie zur nächsten Zeile übergehen.

Geben Sie Werte in beliebiger numerischer Reihenfolge ein.

Geben Sie hier einen neuen Wert ein.

Zeigt den ganzen Wert der markierten Zelle.



Hinweis: In diesem Beispiel können Sie den Cursor auf Spalte 2 setzen, aber Werte nur in Spalte 1 eingeben.

Eine Liste in die Spalte der unabhängigen Variablen eingeben

1. Bewegen Sie den Cursor, um eine beliebige Zelle in der Spalte der unabhängigen Variablen zu markieren.
2. Drücken Sie [F4], um den Cursor auf die Eingabezeile zu setzen.

3. Geben Sie in geschweiften Klammern { } eine Liste durch Kommata voneinander getrennter Werte ein. Beispiel:

   ```
   x={1,1.5,1.75,2}
   ```

   Sie können auch eine Liste als Variable oder einen Term, der eine Liste ergibt, eingeben.

   Hinweis: Enthält die Spalte der unabhängigen Variablen bereits Werte, werden diese als Liste dargestellt (die Sie bearbeiten können).

4. Drücken Sie [ENTER], um die Werte in die Spalte der unabhängigen Variablen zu verschieben. Die Tabelle wird aktualisiert, und die entsprechenden Funktionswerte werden angezeigt.

Einfügen, Löschen oder Entfernen

Wirkung	Vorgehensweise:
Einfügen einer neuen Zeile über einer bestimmten Zeile	Markieren Sie eine Zelle in der angegebenen Reihe, und drücken Sie folgende Tasten: [2nd] [F6] [F6] Die neue Reihe bleibt undefiniert (undef), bis ein Wert für die unabhängige Variable eingegeben wird.
Löschen einer Zeile	Markieren Sie eine Zelle der Zeile, und drücken Sie [F5]. Wenn Sie eine Zelle in der Spalte der unabhängigen Variablen markieren, können Sie auch [←] drücken.

Wirkung	Vorgehensweise:
Entfernen der gesamten Tabelle (nicht aber der gewählten Y= Funktionen)	Drücken Sie F1 **8**. Werden Sie zum Bestätigen aufgefordert, drücken Sie ENTER.

Zellenbreite und Anzeigeformate

Wie Zahlen in einer Tabelle angezeigt werden, hängt von verschiedenen Faktoren ab. .

Im Hauptbildschirm oder in einem Programm

Die Systemvariable **tblInput** enthält eine Liste aller Werte unabhängiger Variablen, die in die Tabelle eingegeben wurden, einschließlich der momentan nicht angezeigten. **tblInput** wird auch für automatische Tabellen verwendet, doch sind dann nur die momentan angezeigten Werte der unabhängigen Variablen enthalten.

Vor der Anzeige einer Tabelle können Sie direkt in der **tblInput**-Systemvariablen eine Werteliste speichern.

Weiteres zur graphischen Darstellung

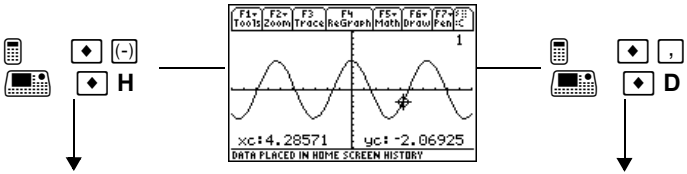
Datenpunkte eines Graphen aufnehmen

Mit Hilfe des Graphikbildschirms können Sie Koordinaten-Werte und/oder mathematische Ergebnisse für eine spätere Untersuchung speichern. Sie können die Informationen als einreihige Matrix (Vektor) im Hauptbildschirm oder als Datenpunkte in einer Systemdatenvariablen, die im Daten/Matrix-Editor geöffnet werden kann, speichern.

Die Punkte aufnehmen

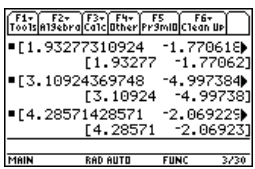
1. Zeigen Sie den Graphen an. (In diesem Beispiel **y1(x)=5∗cos(x)**.)
2. Zeigen Sie die aufzunehmenden Koordinaten oder mathematischen Ergebnisse an.
3. Drücken Sie ◆ (-) oder ◆ , um die Informationen in den Hauptbildschirm bzw. die *sysData* -Variable abzulegen.
   - ◆ (-) (Hauptbildschirm) oder ◆ , (*sysData* als Variable)
   - ◆ **H** (Hauptbildschirm) oder ◆ **D** (*sysData* als Variable)
4. Wiederholen Sie diesen Vorgang so oft wie nötig.

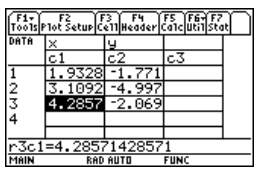
   Hinweis: Tracen Sie eine Funktion mit F3, oder führen Sie eine F5 **Math**-Operation durch (z.B. **Minimum** oder **Maximum**), um Koordinaten oder mathematische Ergebnisse anzuzeigen. Sie können auch den frei beweglichen Cursor verwenden.



Angezeigte Koordinaten werden als einreihige Matrix oder Vektor in den Protokoll-Bereich des Hauptbildschirms übertragen (nicht in die Eingabezeile).

Angezeigte Koordinaten werden in einer Datenvariablen namens *sysData* gespeichert, die im Daten/Matrix-Editor geöffnet werden kann.





Hinweis: Verwenden Sie einen geteilten Bildschirm, um eine Graphik und den Hauptbildschirm bzw. den Daten/Matrix-Editor gleichzeitig anzuzeigen.

Anmerkungen zur SysData-Variablen

- Mit den Tasten:

  - wird *sysData*, sollte sie noch nicht bestehen, im Verzeichnis **MAIN** erzeugt.
  - Sollte *sysData* bereits vorhanden sein, werden neue Daten an das Ende der bestehenden Datei angefügt. Bestehende Titel oder Kopfzeilen (der betroffenen Spalten) werden gelöscht; Titel werden durch die Titel für die neuen Daten ersetzt.

- Die *sysData* -Variable kann gelöscht, entfernt usw. werden, wie jede andere Datenvariable. Sie kann aber nicht gesperrt werden.
- Enthält der Graphikbildschirm eine Funktion oder einen Statistik-Plot mit Bezug auf den aktuellen Inhalt von *sysData*, funktioniert diese Anweisung nicht.

Eine im Hauptbildschirm definierte Funktion graphisch darstellen

Es kann häufig vorkommen, dass Sie eine Funktion oder einen Term im Hauptbildschirm erzeugen und diese(n) dann graphisch darstellen möchten. Sie können einen Term in den Y= Editor kopieren oder, ohne den Y= Editor zu verwenden, direkt vom Hauptbildschirm aus graphisch darstellen.

Was ist die “standardmäßig zugeordnete” unabhängige Variable?

Im Y= Editor müssen alle Funktionen bezüglich der dem aktuellen Graphikmodus “standardmäßig zugeordneten” unabhängigen Variablen definiert werden.

Graphikmodus	Standardmäßig zugeordnete unabhängige Variable
Function	x
Parametric	t
Polar	θ
Sequence	n

Graphikmodus	Standardmäßig zugeordnete unabhängige Variable
3D	x, y
Differential Equation	t

Aus dem Hauptbildschirm in den Y= Editor kopieren

Einen Term im Hauptbildschirm können Sie mit jedem der unten beschriebenen Verfahren in den Y= Editor kopieren.

Verfahren	Beschreibung
Kopieren und Einfügen	1. Markieren Sie den Term im Hauptbildschirm. Drücken Sie [F1], und wählen Sie **5:Copy**. 2. Öffnen Sie den Y= Editor, markieren Sie die gewünschte Funktion, und drücken Sie [ENTER]. 3. Drücken Sie [F1], und wählen Sie **6:Paste**. Drücken Sie dann [ENTER]. **Hinweis:** Alternative zur Verwendung von [F1] **5** oder [F1] **6** zum Kopieren und Einfügen: [♦] [COPY] oder [♦] [PASTE]. [♦] **C** (copy) oder [♦] **V** (paste).

Verfahren	Beschreibung
STO▸	Speichern Sie den Term in einen Y= Funktionsnamen. `2x^3+3x^2−4x+12→y1(x)` Den vollständigen Funktionsnamen verwenden: y1(x), und nicht nur y1. **Hinweis:** Kopieren Sie einen Term aus dem Protokoll-Bereich des Hauptbildschirms mit der automatischen Einfüge-Funktion oder durch Kopieren und Einfügen in die Eingabezeile.
Define Befehl	Definieren Sie den Term als benutzerdefinierte Y= Funktion. `Define y1(x)=2x^3+3x^2−4x+12` **Hinweis: Define** ist im Menüleisten-Menü F4 des Hauptbildschirms verfügbar.

Verfahren	Beschreibung
2nd [RCL]	Wenn der Term bereits in eine Variable gespeichert wurde: 1. Öffnen Sie den Y= Editor, markieren Sie die gewünschte Funktion, und drücken Sie ENTER. 2. Drücken Sie 2nd [RCL]. Geben Sie den Namen der Variablen an, die den Term enthält, und drücken Sie ENTER zweimal. **Wichtig:** Geben Sie zum Abrufen einer Funktionsvariablen wie z.B. **f1(x)** nur **f1** und nicht den ganzen Funktionsnamen ein. 3. Drücken Sie ENTER, um den abgerufenen Term in der Funktionsliste des Y= Editors zu speichern. **Hinweis:** 2nd [RCL] ist hilfreich, wenn ein Term in einer Variablen oder einer Funktion gespeichert ist, die nicht mit dem Y= Editor übereinstimmt, wie z.B. **a1** oder **f1(x)**.

Direkt vom Hauptbildschirm aus zeichnen

Mit dem **Graph**-Befehl können Sie einen Term direkt vom Hauptbildschirm aus graphisch darstellen, ohne den Y= Editor benutzen zu müssen. Im Gegensatz zum Y= Editor ermöglicht **Graph** die Angabe eines Terms mit einer beliebigen unabhängigen Variablen, wobei der aktuelle Graphikmodus keine Rolle spielt.

Angabe des Terms durch:	Verwenden Sie den Befehl Graph wie hier gezeigt:
Die standardmäßig zugeordnete unabhängige Variable	Graph 1.25x∗cos(x) Für die graphische Darstellung von Funktionen ist x die zugeordnete Variable.

Angabe des Terms durch:	Verwenden Sie den Befehl Graph wie hier gezeigt:
Eine nicht-zugeordnete unabhängige Variable	Graph 1.25a∗cos(a),a Geben Sie die unabhängige Variable an, sonst könnte ein Fehler entstehen.

Hinweis: Graph verwendet die aktuellen Window-Variablen-Einstellungen. **Graph** ist im Menüleisten-Menü F4 des Hauptbildschirms verfügbar.

Folgen oder Differentialgleichungen können Sie auf diese Weise nicht mit **Graph** darstellen. Verwenden Sie für Parameter-, Polar- und 3D-Darstellungen die folgenden Varianten.

Im PARAMETRIC Graphikmodus	**Graph** *xExpr*, *yExpr*, *t*
Im POLAR Graphikmodus	**Graph** *expr*, θ
Im 3D Graphikmodus	**Graph** *expr*, *x*, *y*

Hinweis: Verwenden Sie für die Erstellung einer Tabelle im Hauptbildschirm den Befehl Table. Er gleicht dem Befehl **Graph**. Beide verwenden dieselben Terme.

Durch **Graph** wird der Term nicht in den Y= Editor kopiert. Statt dessen werden etwaige im Y= Editor ausgewählte Funktionen zeitweilig ausgesetzt. Sie können **Graph**-Terme genau wie Y= Editor-Funktionen tracen, zoomen oder im Tabellenbildschirm anzeigen und bearbeiten.

Den Graphikbildschirm löschen

Jedesmal, wenn Sie **Graph** ausführen, wird der neue Term zu den bereits bestehenden hinzugefügt. So löschen Sie die Darstellungen:

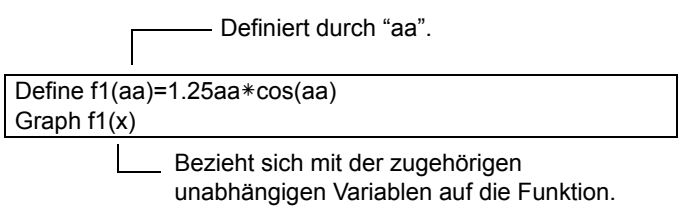
- Führen Sie den Befehl **ClrGraph** aus (verfügbar im Menüleisten-Menü F4 **Other** des Hauptbildschirms).

  – oder –

- Aktivieren Sie den Y= Editor. Wenn Sie den Graphikbildschirm wieder öffnen, verwendet er die im Y= Editor gewählten Funktionen.

Besondere Vorteile benutzerdefinierter Funktionen

Sie können eine benutzerdefinierte Funktion mit jeder beliebigen unabhängigen Variablen definieren. Beispiel:



und:

```
Define f1(aa)=1.25aa∗cos(aa)
f1(x)→y1(x)
```

Eine stückweise definierte Funktion graphisch darstellen

Möchten Sie eine stückweise definierte Funktion graphisch darstellen, müssen Sie diese zunächst durch Angabe von Grenzen und Termen für jeden Teil definieren. Für zweiteilige Funktionen ist die **when**-Funktion besonders nützlich. Für drei oder mehr Teile bietet es sich an, eine benutzerdefinierte Funktion mit mehreren Anweisungen zu erstellen.

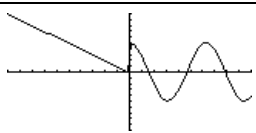
Die When-Funktion verwenden

Verwenden Sie für das Definieren einer zweiteiligen Funktion folgende Syntax:

when(*Bedingung, wahrerTerm, falscherTerm***)**

Sie möchten beispielsweise eine zweiteilige Funktion graphisch darstellen.

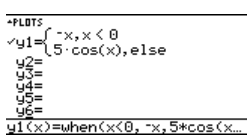
Wenn:	**Verwenden Sie den Term:**
$x < 0$	$-x$
$x \geq 0$	$5\cos(x)$



Im Y= Editor:

Die Funktion wird in dieser Form angezeigt.

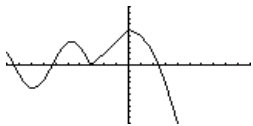
Geben Sie die Funktion so ein.



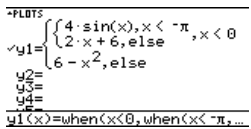
Für drei oder mehr Teile können Sie geschachtelte **when**-Funktionen verwenden.

Hinweis: when wird über die Tastatur oder das Dialogfeld **CATALOG** eingegeben.

Wenn:	**Verwenden Sie den Term:**
$x < -\pi$	$4 \sin(x)$
$x \geq -\pi$ and $x < 0$	$2x + 6$
$x \geq 0$	$6 - x^2$



Im Y= Editor



wobei:

```
y1(x)=when(x<0,when(x<-π,4*sin(x),2x+6),6-x^2)
```

Diese geschachtelte Funktion gilt, wenn x<0.

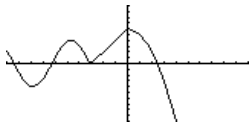
Geschachtelte Funktionen werden schnell komplex und schlecht zu überblicken.

Eine benutzerdefinierte Funktion mit mehreren Anweisungen verwenden

Für drei oder mehr Teile sollten Sie eine benutzerdefinierte Funktion mit mehreren Anweisungen erstellen.

Gehen Sie beispielsweise von der vorigen dreiteiligen Funktion aus.

Wenn:	Verwenden Sie den Term:
$x < -\pi$	4 sin(x)
$x \geq -\pi$ and $x < 0$	2x + 6



Hinweis: Näheres über Ähnlichkeiten und Unterschiede von Funktionen und Programmen finden Sie *Programmierung*.

Eine benutzerdefinierte Funktion mit mehreren Anweisungen kann viele der beim Programmieren verwendeten Steuer- und Entscheidungsstrukturen enthalten (**If**, **ElseIf**, **Return** etc.).Wenn Sie die Struktur einer Funktion erstellen, könnte es hilfreich sein, sich diese zuerst in Blockform zu vergegenwärtigen.

```
❶ Func
    If x<-π Then
    Return 4*sin(x)
  ElseIf x>=-π and x<0 Then
    Return 2x+6
  Else
    Return 6-x^2
  EndIf
❶ EndFunc
```

❶ **Func** und **EndFunc** müssen am Anfang/Ende der Funktion stehen.

Im Y= Editor oder im Hauptbildschirm müssen Sie eine aus mehreren Anweisungen bestehende Funktion in eine einzige Zeile eingeben.

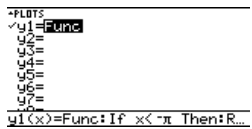
Trennen Sie die einzelnen Anweisungen durch Doppelpunkte (:) voneinander ab.

```
Func:If x<⁻π Then:Return 4∗sin(x): ... :EndIf:EndFunc
```

Im Y= Editor:

Bei einer Funktion mit mehreren Anweisungen wird nur **Func** angezeigt.

Geben Sie eine Funktion mit mehreren Anweisungen in eine einzige Zeile ein. Denken Sie an die Doppelpunkte.



Im Hauptbildschirm oder von einem Programm aus

Im Hauptbildschirm können Sie zum Erstellen einer benutzerdefinierten Funktion mit mehreren Anweisungen auch den Befehl **Define** verwenden. Weitere Angaben zum Kopieren einer Funktion aus dem Hauptbildschirm in den Y= Editor sind verfügbar. .

Im Programmeditor können Sie ebenfalls benutzerdefinierte Funktionen erstellen. Verwenden Sie den Programmeditor beispielsweise zum Erstellen einer Funktion namens **f1(xx)**. Setzen Sie im Y= Editor **y1(x) = f1(x)**.

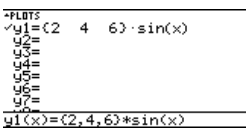
Eine Kurvenschar graphisch darstellen

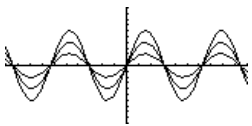
Wenn Sie eine Liste in einen Term einfügen, können Sie für jeden Wert der Liste eine einzelne Funktion plotten. (Eine Kurvenschar kann weder im SEQUENCE- noch im 3D-Graphikmodus dargestellt werden.)

Beispiele zur Verwendung des Y= Editors

Geben Sie den Term **{2,4,6} sin(x)** ein, und stellen Sie die Funktionen graphisch dar.

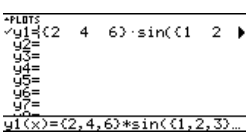
Hinweis: Klammern Sie Listenelemente ein (2nd [{] und 2nd [}]), und trennen Sie sie durch Kommata voneinander ab.

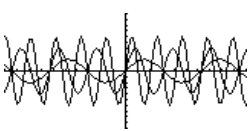




Stellt drei Funktionen graphisch dar:
2 sin(x), 4 sin(x), 6 sin(x)

Geben Sie den Term **{2,4,6} sin({1,2,3} x)** ein, und stellen Sie die Funktionen graphisch dar.





Stellt drei Funktionen graphisch dar:
2 sin(x), 4 sin(2x), 6 sin(3x)

Hinweis: Die Kommata werden zwar in der Eingabezeile angegeben, aber in der Funktionsliste nicht angezeigt.

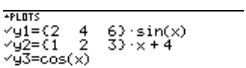
Beispiel für die Verwendung des Graph-Befehls

Analog dazu können Sie im Hauptbildschirm oder von einem Programm aus den Graph-Befehl verwenden.

```
graph {2,4,6}sin(x)
graph {2,4,6}sin({1,2,3}x)
```

Gleichzeitige graphische Darstellung mit Listen

Bei der Graphikformat-Einstellung **Graph Order = SIMUL** werden die Funktionen gruppenweise graphisch dargestellt, wobei Elemente mit derselben Position in der Liste in einer Gruppe zusammengefaßt werden.



Für diese Funktionen zeichnet der TI-89 Titanium / Voyage™ 200 Graphischer Rechner drei Gruppen.

- 2 sin(x), x+4, cos(x)
- 4 sin(x), 2x+4
- 6 sin(x), 3x+4

Die Funktionen innerhalb einer Gruppe werden gleichzeitig gezeichnet, die einzelnen Gruppen nacheinander.

Hinweis: Graphenformate im Bildschirm Y= Editor, Window Editor oder Graphik-Bildschirm werden wie folgt eingestellt:

Tracen einer Kurvenschar

Mit ⊙ oder ⊙ wird der Trace-Cursor zuerst zur nächsten oder vorigen Kurve derselben Schar bewegt und dann zur nächsten oder vorigen gewählten Funktion.

Den 2-Graphen-Modus verwenden

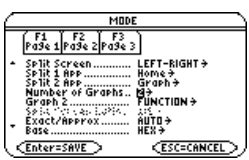
Im 2-Graphen-Modus sind die Graphikfunktionen des TI-89 Titanium / Voyage™ 200 Graphischer Rechner verdoppelt, so daß Ihnen zwei unabhängige Graphik-Rechner zur Verfügung stehen. Der 2-Graphen-Modus ist nur bei geteilten Bildschirmen verfügbar. Näheres zu geteilten Bildschirmen finden Sie unter *Geteilte Bildschirme*.

Den Modus einstellen

Mehrere Moduseinstellungen wirken sich auf den 2-Graphen-Modus aus, es sind aber nur zwei Einstellungen erforderlich. Beide befinden sich auf Seite 2 des Dialogfeldes **MODE**.

1. Drücken Sie MODE. Drücken Sie dann F2, um Seite 2 zu öffnen.

2. Diese Modi müssen Sie einstellen.
   - **Split Screen = TOP-BOTTOM** oder **LEFT-RIGHT**
   - **Number of Graphs = 2**



3. Folgende Modi können Sie nach Bedarf einstellen.

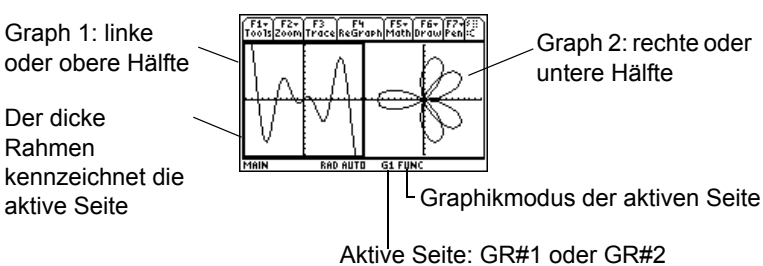
Seite 1:
- **Graph = Graph**-Modus für die obere oder linke Hälfte des geteilten Bildschirms

Seite 2:
- **Split 1 App** = Anwendung für obere oder linke Hälfte
- **Split 2 App** = Anwendung für untere oder rechte Hälfte
- **Graph 2 = Graph** Graph-Modus für untere oder rechte Hälfte
- **Split Screen Ratio** = Größenverhältnis der beiden Hälften (nur Voyage™ 200)

4. Drücken Sie ENTER, um das Dialogfeld zu schließen.

Der 2-Graphen-Bildschirm

Ein 2-Graphen-Bildschirm gleicht dem normalen geteilten Bildschirm.



Unabhängige Graphikmerkmale

Die Bildschirmteile Graph 1 und Graph 2 verfügen über unabhängige:

- Graph-Modi (FUNCTION, POLAR etc.). Andere Modi, wie beispielsweise **Angle**, **Display Digits** etc., betreffen beide Graphen.
- Window-Editor-Variablen.
- Tabellen-Setup-Parameter und Tabellenbildschirme.
- Graphikformate, wie z.B. **Coordinates**, **Axes** etc.
- Graphikbildschirme.
- Y= Editoren. Über die Funktions- und Statistik-Plot-Definitionen verfügen die Graphen jedoch gemeinsam.

  Hinweis: Der Y= Editor ist nur dann vollkommen unabhängig, wenn für die Seiten unterschiedliche Graphikmodi verwendet werden (siehe unten).

Unabhängige graphikbezogene Anwendungen (Y= Editor, Graphikbildschirm, etc.) können gleichzeitig auf beiden Seiten des Bildschirms geöffnet werden.

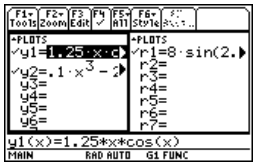
Die nicht-graphikbezogenen Anwendungen (Hauptbildschirm, Daten/Matrix-Editor, etc.) werden gemeinsam genutzt und können deshalb nur jeweils auf einer Seite geöffnet werden.

Der Y= Editor im 2-Graphen-Modus

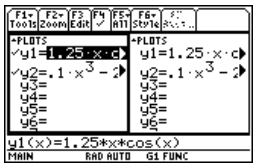
Im 2-Graphen-Modus ist eigentlich nur ein Y= Editor vorhanden, der für jede Graph-Moduseinstellung eine eigene Funktionsliste verwaltet. Wenn beide Seiten denselben

Graphikmodus verwenden, können aus dieser Liste unterschiedliche Funktionen für jede Seite gewählt werden.

- Wenn die Seiten unterschiedliche Graphikmodi verwenden, werden unterschiedliche Funktionslisten angezeigt.



- Wenn die Seiten denselben Graphikmodus verwenden, wird auf jeder Seite dieselbe Funktionsliste angezeigt.
  - Mit F4 können Sie für jede Seite unterschiedliche Funktionen und Statistik-Plots wählen (gekennzeichnet durch ✓).
  - Wählen Sie für eine Funktion eine bestimmte Anzeigeart, so wird diese auf beiden Seiten verwendet
    [2nd] [F6]
    [F6]



- Wenn Graph 1 und Graph 2 für die graphische Darstellung von Funktionen eingestellt sind, können Sie für die Darstellung unterschiedliche Funktionen wählen (✓), obwohl auf beiden Seiten dieselbe Funktionsliste angezeigt wird.

Hinweis: Eine Änderung im aktiven Y= Editor (Neudefinieren einer Funktion, Ändern eines Stils etc.), wird auf der inaktiven Seite erst dann wiedergegeben, wenn Sie auf diese umschalten.

Umgang mit geteilten Bildschirmen

Näheres zu geteilten Bildschirmen finden Sie in *Geteilte Bildschirme*.

- Drücken Sie zum Umschalten zwischen den Seiten 2nd [⊞] (Zweitfunktion von APPS).
- So öffnen Sie unterschiedliche Anwendungen:
  - Schalten Sie auf die gewünschte Seite, und öffnen Sie die Anwendung wie gewohnt.

    – oder –
  - Verwenden Sie MODE, um **Split 1 App** und/oder **Split 2 App** zu ändern.
- So verlassen Sie den 2-Graphen-Modus:
  - Nehmen Sie mit MODE die Einstellung **Number of Graphs = 1** vor, oder verlassen Sie den geteilten Bildschirm, indem Sie **Split Screen = FULL** einstellen.

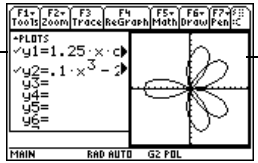
    – oder –
  - Drücken Sie 2nd [QUIT] zweimal. Dadurch können Sie stets einen geteilten Bildschirm verlassen und zum ungeteilten Hauptbildschirm zurückkehren.

Hinweis: Nicht-graphikbezogene Anwendungen (z.B. den Hauptbildschirm) können Sie jeweils nur auf einer Seite öffnen.

Beachten Sie, daß die beiden Seiten unabhängig voneinander sind

Im 2-Graphen-Modus können die beiden Seiten den Anschein erwecken, miteinander zusammenzuhängen, was aber in Wirklichkeit nicht zutrifft. Beispiel:

Für Graph 1 listet der Y= Editor y(x)-Funktionen auf.



Für Graph 2 werden bei der polaren Darstellung r(θ) - Gleichungen verwendet, die aber nicht angezeigt sind.

Im Hauptbildschirm oder von einem Programm aus

Nach der Einrichtung des 2-Graphen-Modus beziehen sich graphikbezogene Operationen auf die aktive Seite. Beispiel:

```
10→xmax
```

betrifft die bei Ausführung des Befehls aktive Seite; dies kann entweder Graph 1 oder Graph 2 sein.

Drücken Sie 2nd [⇔] um die aktivierten Seiten zu wechseln, oder verwenden Sie die **switch**-Funktion **switch(1)** bzw. **switch(2)**.

Eine Funktion oder Inverse in einer Graphik zeichnen

Zu Vergleichszwecken können Sie über Ihre aktuelle Darstellung eine weitere Darstellung zeichnen. Letztere ist meistens eine Variation der ersten Darstellung. Sie können auch die Inverse zeichnen. (Diese Operationen sind für 3D-Darstellungen nicht möglich.)

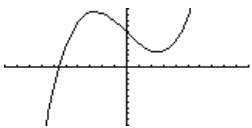
Eine Funktion, eine Parameter- oder eine Polardarstellung zeichnen

Führen Sie im Hauptbildschirm oder von einem Programm aus **DrawFunc**, **DrawParm** oder **DrawPol** aus. Im Graphikbildschirm können Funktionen oder Gleichungen nicht interaktiv gezeichnet werden.

DrawFunc *Term*
DrawParm *Term1, Term2* [,*tmin*] [,*tmax*] [,*tstep*]
DrawPol *Term* [,θ*min*] [,θ*max*] [,θ*step*]

Beispiel:

1. Definieren Sie im Y= Editor **y1(x)=.1x^3–2x+6**, und stellen Sie die Funktion graphisch dar.



2. Drücken Sie im Graphikbildschirm:
   [2nd] [F6]
   [F6]
   und wählen Sie **2:DrawFunc**.

   Mit folgenden Tasten wird der Hauptbildschirm angezeigt und **DrawFunc** in die Eingabezeile übertragen:
   [2nd] [F6] **2**
   [F6] **2**

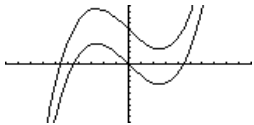
3. Geben Sie im Hauptbildschirm die zu zeichnende Funktion an.

```
DrawFunc y1(x)−6
```

4. Drücken Sie [ENTER], um die Funktion auf dem Graphikbildschirm zu zeichnen.

   Sie können keine Trace-, Zoom- oder mathematische Operation an einer gezeichneten Funktion vornehmen.

   Hinweis: Zum Löschen der gezeichneten Funktion drücken Sie [F4]
   – oder –
   [2nd] [F6] und wählen Sie **1:ClrDraw**
   [F6] und wählen Sie **1:ClrDraw**



Die Inverse einer Funktion zeichnen

Führen Sie im Hauptbildschirm oder von einem Programm aus den Befehl **DrawInv** aus. Im Graphikbildschirm können inverse Funktionen nicht interaktiv gezeichnet werden.

DrawInv *Term*

Verwenden Sie z.B. den oben gezeigten Graphen von **y1(x)=.1x^3−2x+6**.

1. Drücken Sie folgende Tasten im Bildschirm zur Graphenerzeugung:
   [2nd] [F6]
   [F6]
   and select **3:DrawInv**

   Mit folgenden Tasten wird der Hauptbildschirm angezeigt und **DrawInv** in die Eingabezeile übertragen:

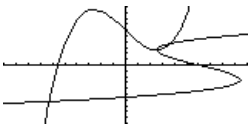
2nd [F6] **3**
F6 **3**

2. Geben Sie im Hauptbildschirm die inverse Funktion an.

```
DrawInv y1(x)
```

3. Drücken Sie ENTER.

   Die Inverse wird als **(y,x)** anstatt als **(x,y)** geplottet.



Eine Graphik mit Linien, Kreisen oder einem Text ergänzen

Sie können, normalerweise für Vergleichszwecke, ein oder mehrere Objekte auf den Graphikbildschirm zeichnen. Zeichnen Sie z.B. eine horizontale Linie, um zu zeigen, dass zwei Seiten eines Graphen denselben y-Wert haben. (Einige Objekte können bei 3D-Darstellungen nicht verwendet werden.)

Alle Zeichnungen löschen

Ein zusätzlich gezeichnetes Objekt ist kein Bestandteil der ursprünglichen Graphik. Es wird “darüber” gezeichnet und bleibt auf dem Bildschirm eingeblendet, bis Sie es löschen.

Im Graphikbildschirm:

- [2nd] [F6]
  [F6]
  und wählen Sie **1:ClrDraw**.

  – oder –

- Drücken Sie [F4], um das Objekt erneut zu zeichnen.

Hinweis: Sie können **ClrDraw** auch in die Eingabezeile des Hauptbildschirms eingeben.

Sie können auch eine beliebige Aktion durchführen, welche die Smart Graph-Funktion dazu veranlaßt, das Objekt erneut graphisch darzustellen (z.B. die Window-Variablen ändern oder die Auswahl einer gewählten Funktion im Y= Editor aufheben).

Einen Punkt oder eine Freihand-Linie zeichnen

Im Graphikbildschirm:

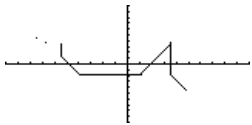
1. [2nd] [F7]
   [F7]
   und wählen Sie: **1:Pencil**.
2. Setzen Sie den Cursor auf die gewünschte Stelle.

Zum Zeichnen:	Vorgehensweise:
eines Punkts (in Pixelgröße)	Drücken Sie [ENTER].
einer Freihand-Linie	Drücken und halten Sie die Taste [⇧], und bewegen Sie den Cursor, um die Linie zu zeichnen. Drücken und halten Sie die Taste [⎋], und zeichnen Sie die Linie mit dem Cursor. Beenden Sie den Vorgang durch Loslassen von [⇧] oder [⎋].

Hinweis: Beim freihändigen Zeichnen einer Linie können Sie den Cursor diagonal

bewegen.

Nach dem Zeichnen des Punkts/ der Linie befinden Sie sich weiterhin im “Bleistift”-Modus.



- Wenn Sie weiter zeichnen möchten, bewegen Sie den Cursor an einen anderen Punkt.
- Zum Abbrechen drücken Sie [ESC].

Hinweis: Beginnen Sie den Zeichenvorgang auf einem weißen Pixel, zeichnet der Bleistift eine schwarze Linie/einen schwarzen Punkt. Beginnen Sie auf einem schwarzen Pixel, wird eine weiße Linie/ein weißer Punkt gezeichnet (kann als Radiergummi wirken).

Einzelne Teile eines Zeichenobjekts entfernen

Drücken Sie im Graphik-Bildschirm folgende Tasten:

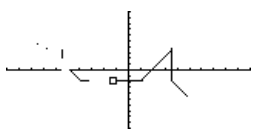
1. [2nd] [F7]
   [F7]
   und wählen Sie **2:Eraser**. Der Cursor wird als kleines Viereck angezeigt.

2. Setzen Sie den Cursor auf die gewünschte Stelle.

Entfernen:	Vorgehensweise:
Fläche unter dem Kästchen	Drücken Sie [ENTER].
Abschnitt einer Freihand-Linie	Drücken Sie die Taste [↑], halten Sie sie gedrückt, und löschen Sie die Linie mit dem Cursor. Drücken Sie die Taste [⎋], halten Sie sie gedrückt, und löschen Sie die Linie mit dem Cursor. Beenden Sie den Vorgang durch Loslassen von [↑] oder [⎋].

Hinweis: Mit diesen Verfahren können auch Teile graphisch dargestellter Funktionen entfernt werden.

Nach dem Entfernen befinden Sie sich weiterhin im “Radiergummi”-Modus.



- Möchten Sie weiter “radieren”, bewegen Sie den quadratischen Cursor an eine andere Stelle.
- Zum Abbrechen drücken Sie [ESC].

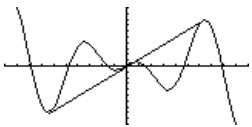
Eine Linie zwischen zwei Punkten zeichnen

Drücken Sie im Graphik-Bildschirm folgende Tasten:

1. [2nd] [F7]
   [F7]
   und wählen Sie **3:Line**.
2. Setzen Sie den Cursor auf den ersten Punkt, und drücken Sie [ENTER].
3. Bewegen Sie ihn zum zweiten Punkt, und drücken Sie [ENTER]. (Bei der Bewegung entsteht zwischen dem ersten Punkt und dem Cursor eine Linie.)

   Hinweis: Verwenden Sie [2nd], um den Cursor in größeren Schritten zu bewegen; [2nd] ▶ etc.

Nach dem Zeichnen der Linie befinden Sie sich weiterhin im “Linien”-Modus.



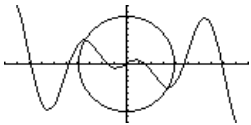
- Möchten Sie eine weitere Linie zeichnen, setzen Sie den Cursor auf einen neuen ersten Punkt.
- Zum Abbrechen drücken Sie [ESC].

Einen Kreis zeichnen

Drücken Sie im Graphik-Bildschirm folgende Tasten:

1. [2nd] [F7]
   [F7]
   und wählen Sie **4:Circle**.

2. Setzen Sie den Cursor an die Stelle für den Kreismittelpunkt, und drücken Sie [ENTER].



3. Bewegen Sie den Cursor, um den Radius zu bestimmen, und drücken Sie [ENTER].

Hinweis: Verwenden Sie [2nd], um den Cursor in größeren Schritten zu bewegen; [2nd] ◗ etc.

Eine horizontale oder vertikale Linie zeichnen

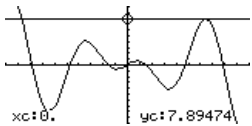
Ausgangspunkt ist der Graphikbildschirm:

1. [2nd] [F7]
   [F7]
   und wählen Sie **5:Horizontal** oder **6:Vertical**. Auf dem Bildschirm wird eine horizontale oder vertikale Linie sowie ein blinkender Cursor angezeigt.

   Befindet sich die Linie anfangs auf einer Achse, ist sie kaum sichtbar. Den blinkenden Cursor können Sie jedoch problemlos erkennen.

2. Verschieben Sie die Linie mit den Cursortasten an die gewünschte Stelle. Drücken Sie dann [ENTER].

   Nach dem Zeichnen der Linie befinden Sie sich weiterhin im “Linien”-Modus.

   - Möchten Sie weiter zeichnen, setzen Sie den Cursor auf eine andere Stelle.
   - Zum Abbrechen drücken Sie [ESC].

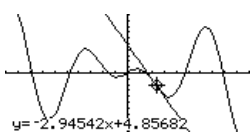
   Hinweis: Verwenden Sie [2nd], um den Cursor in größeren Schritten zu bewegen; [2nd] ⊙ etc.

Eine Tangente zeichnen

Verwenden Sie zum Zeichnen einer Tangente das Menüleisten-Menü [F5] **Math**. Im Graphikbildschirm:

1. Drücken Sie [F5], und wählen Sie **A:Tangent**.
2. Verwenden Sie ggf. ⊙ und ⊙, um die gewünschte Funktion zu wählen.
3. Setzen Sie den Cursor auf den Berührpunkt, und drücken Sie [ENTER].

   Die Tangente wird gezeichnet und ihre Gleichung eingeblendet.

   Hinweis: Zum Bestimmen des Berührpunkts können Sie auch dessen x-Wert eingeben und [ENTER] drücken.

Eine Linie durch einen Punkt mit einer Steigung zeichnen

Zum Zeichnen einer Linie durch einen bestimmten Punkt mit einer bestimmten Steigung führen Sie im Hauptbildschirm oder von einem Programm aus den **DrawSlp**-Befehl aus. Verwenden Sie folgende Syntax:

DrawSlp *x, y, Steigung*

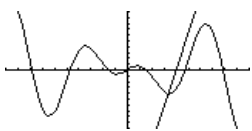
Sie können auf **DrawSlp** auch vom Graphikbildschirm aus zugreifen.

1. Drücken Sie
   [2nd] [F6]
   [F6]
   und wählen Sie **6:DrawSlp**. Dadurch wird auf den Hauptbildschirm umgeschaltet, und **DrawSlp** wird in die Eingabezeile gesetzt.

2. Vervollständigen Sie den Befehl, und drücken Sie [ENTER].

   Der TI-89 Titanium / Voyage™ 200 Graphischer Rechner schaltet automatisch auf den Graphikbildschirm um und zeichnet die Linie.

```
DrawSlp 4,0,6.37
```



Beschriftungen eingeben

Ausgangspunkt ist der Graphikbildschirm:

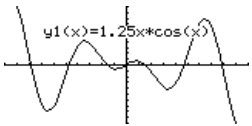
1. [2nd] [F7]
   [F7]
   und wählen Sie **7:Text**.

2. Setzen Sie den Text-Cursor an die Stelle, an welcher Sie zu schreiben beginnen möchten.

3. Geben Sie den Beschriftungstext ein.

   Nach Eingabe des Textes befinden Sie sich weiterhin im “Text”-Modus.

   - Möchten Sie fortfahren, setzen Sie den Cursor an eine andere Stelle.
   - Zum Abbrechen drücken Sie ENTER oder ESC.



Hinweis: Der Text-Cursor steht an der Position, an der die obere linke Ecke des Zeichens erscheint, das als nächstes geschrieben wird.

Im Hauptbildschirm oder von einem Programm aus

Es sind Befehle zum Zeichnen aller in diesem Abschnitt beschriebenen Objekte verfügbar. Mit bestimmten Befehlen (z.B. **PxlOn**, **PxlLine**, etc.) können Sie Objekte zeichnen, indem Sie genaue Pixel-Positionen auf dem Bildschirm angeben.

Eine Liste der verfügbaren Zeichenbefehle finden Sie in “Im Graphikbildschirm zeichnen” in *Programmierung*.

Ein Graphikbild speichern und öffnen

Sie können ein Bild des aktuellen Graphikbildschirms in einer PICTURE- (PIC-) Variablen speichern. Später können Sie dann die Variable öffnen und das Bild anzeigen.

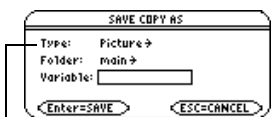
Dabei wird nur das Bild gespeichert, nicht die Graphik-Einstellungen, die zu dessen Erstellung verwendet werden.

Ein Bild des gesamten Graphikbildschirms speichern

Das Bild umfaßt alle geplotteten Funktionen, Achsen, Einheiten und zusätzlich gezeichneten Objekte. Das Bild beinhaltet weder die obere/untere Grenzmarke noch Eingabeaufforderungen oder Cursor-Koordinaten.

Zeigen Sie den Graphikbildschirm an, wie Sie ihn speichern möchten. Dann:

1. Drücken Sie F1, und wählen Sie **2:Save Copy As**.
2. Geben Sie den Typ (**Picture**), das Verzeichnis und einen eindeutigen Variablennamen an.
3. Drücken Sie ENTER. Nach dem Ausfüllen eines Eingabefelds müssen Sie ENTER zweimal drücken.



Wichtig: Standardmäßig gilt Type = GDB (für Graphik-Einstellungen). Sie müssen Type = Picture einstellen.

Einen Teil des Graphikbildschirms speichern

Sie können ein rechteckiges Kästchen definieren, das den zu speichernden Ausschnitt des Graphikbildschirms umschließt.

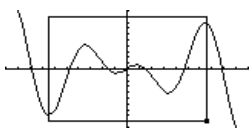
1. [2nd] [F7]
   [F7]
   und wählen Sie **8:Save Picture**.

   Um den äußeren Bildschirmrand herum erscheint ein Kästchen.

   Hinweis: Sie können keinen Ausschnitt einer 3D-Darstellung speichern.

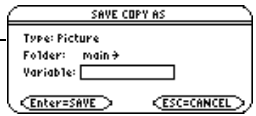
2. Legen Sie die erste Ecke des Kästchens durch Verschieben der oberen und der linken Seite fest. Drücken Sie dann [ENTER].

   Hinweis: Verwenden Sie ▼ und ▲ zum Verschieben der Ober- bzw. Unterseite und ▶ und ◀ zum Verschieben der Seiten.



3. Legen Sie die zweite Ecke des Kästchens durch Verschieben der unteren und der rechten Seite fest. Drücken Sie dann [ENTER].

4. Geben Sie das Verzeichnis und einen eindeutigen Variablennamen an.
5. Drücken Sie [ENTER]. Nach dem Ausfüllen eines Eingabefelds, wie z.B. **Variable**, müssen Sie [ENTER] zweimal drücken.



Hinweis: Beim Speichern eines Graph-Ausschnitts ist für Type automatisch Picture eingestellt.

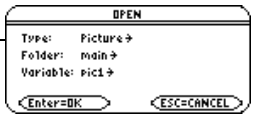
Ein Bild öffnen

Wenn Sie ein Bild öffnen, überlagert es den aktuellen Graphikbildschirm. Um ausschließlich das Bild anzuzeigen, heben Sie im Y= Editor vor dem Öffnen des Bildes die Auswahl aller anderen Funktionen auf.

Im Graphikbildschirm:

1. Drücken Sie [F1], und wählen Sie **1:Open**.
2. Wählen Sie den Typ (**Picture**), das Verzeichnis und die Variable mit der zu öffnenden Graphik.

   Hinweis: Wird im Dialogfeld kein Variablenname angezeigt, dann enthält der Verzeichnis keine Graphikbilder.
3. Drücken Sie [ENTER].



Wichtig: Standardmäßig gilt Type = GDB (für Graph-Einstellung). Stellen Sie unbedingt Type = Picture ein.

Ein Bild ist ein Zeichenobjekt. Kurven in Bildern kann man nicht tracen.

Gespeicherte Ausschnitte des Graphikbildschirms

Drücken Sie F1 und wählen **1:Open**, wird das Bild ausgehend von der oberen linken Ecke des Graphikbildschirms über diesem eingeblendet. Handelt es sich um einen gespeicherten Ausschnitt des Graphikbildschirms, kann es wie eine Verschiebung des darunterliegenden Bildschirminhalts wirken.

Ein Bild löschen

Unbenötigte Picture-Variablen belegen Speicherplatz im Rechner. Verwenden Sie gemäß der Beschreibung in *Speicher und Variablen-Verwaltung* den VAR-LINK-Bildschirm 2nd [VAR-LINK]), um eine Variable zu löschen.

Von einem Programm aus oder im Hauptbildschirm

Verwenden Sie zum Sichern (speichern) und Öffnen (abrufen) eines Bilds die im Modul *Technische Referenz* beschriebenen Befehle **StoPic**, **RclPic**, **AndPic**, **XorPic** und **RplcPic**.

Mit dem Befehl **CyclePic** können Sie eine Folge von Bildern als Animation anzeigen. Ein Beispiel finden Sie auf CyclePic-Befehl.

Eine animierte Bildfolge zeigen

Wie in diesem Kapitel bereits beschrieben wurde, können Sie eine Graphik als Bild speichern. Mit dem Befehl **CyclePic** können Sie durch eine Folge von Bildern "blättern" und so eine Animation erzeugen.

CyclePic-Befehl

Bevor Sie **CyclePic** verwenden können, muß Ihnen eine Reihe von Bildern mit gleichen Grundnamen vorliegen, die von 1 ausgehend durchnumeriert sind (z.B. pic1, pic2, pic3 . . .).

Verwenden Sie zum umlaufenden Anzeigen der Bilder folgende Syntax:

CyclePic *BildNameString, n* [,*Wartezeit*] [,*Zyklen*] [,*Richtung*]
❶ ❷ ❸ ❹ ❺

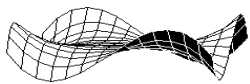
❶ Grundname der Bilder in Anführungszeichen, z.B. "pic"
❷ Anzahl zu durchlaufende Bilder
❸ Sekunden zwischen Bildern
❹ Anzahl der Zyklus

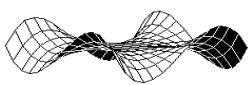
Beispiel

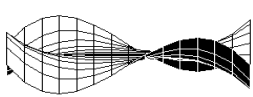
Das Programm (namens **cyc**) in diesem Beispiel erzeugt 10 Ansichten einer 3D-Darstellung, wobei jede Ansicht bezüglich der vorangehenden um 10° um die z-Achse gedreht ist. Näheres zu den einzelnen Befehlen finden Sie im Modul *Technische Referenz*. Erläuterungen zur Verwendung des Programmeditors finden Sie in *Programmierung*.

Programmliste	Jede zweite Graphik des Programms

```
:cyc()
:Prgm
:local I
:©Set mode and Window variables
:setMode("graph","3d")
:70→eyeϕ
:-10→xmin
:10→xmax
:14→xgrid
:-10→ymin
:10→ymax
:14→ygrid
:-10→zmin
:10→zmax
:1→zscl
:©Define the function
:(x^3*y-y^3*x)/390→z1(x,y)
:©Generate pics and rotate
:For i,1,10,1
:  i*10→eyeθ
:  DispG
:  StoPic #("pic" & string(i))
:EndFor
:©Display animation
:CyclePic "pic",10,.5,5,-1
:EndPrgm
```







Kommentare beginnen mit ◉. Drücken Sie:

[♦] [)]

[2nd] **X**

Hinweis: Aufgrund seiner Komplexität nimmt der Ablauf dieses Programms einige Minuten in Anspruch.

Gehen Sie nach Eingabe dieses Programms im Programmeditor zum Hauptbildschirm über, und geben Sie **cyc( )** ein.

Graphik-Einstellungen speichern und öffnen

Die Graphik-Einstellungen bilden die Gesamtheit aller Elemente, die eine bestimmte graphische Darstellung definieren. Wenn Sie eine Graphik-Einstellungen als GDB-Variable speichern, können Sie diese graphische Darstellung zu einem späteren Zeitpunkt durch Öffnen ihrer gespeicherten Einstellungs-Variablen wiederherstellen.

Elemente der Graphik-Einstellungen

Eine Graphik-Einstellung besteht aus:

- Moduseinstellungen ([MODE]) für **Graph**, **Angle**, **Complex Format** und **Split Screen** (nur bei Verwendung des 2-Graphen-Modus).
- Sämtlichen Funktionen im Y= Editor ([♦] [Y=]), einschließlich der Anzeigearten und der Information, welche Funktionen gewählt sind.

- Tabellen-Parametern ([♦] [TBLSET]), Window-Variablen ([♦] [WINDOW]) und Graphikformaten:
  ([F1] **9**
  – oder –
  [♦] [|]
  [♦] **F**

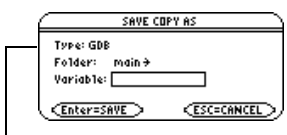
In die Graphik-Einstellungen werden weder zusätzlich gezeichnete Objekte noch Statistik-Plots übernommen.

Hinweis: Im 2-Graphik-Modus werden die Elemente für beide Graphiken in einer einzigen Datenbank gespeichert.

Die aktuelle Graphik-Einstellungen speichern

Ausgangspunkt ist der Y= Editor, Window-Editor, Tabellen-Bildschirm oder Graphikbildschirm:

1. Drücken Sie [F1], und wählen Sie **2:Save Copy As**.
2. Geben Sie das Verzeichnis und einen eindeutigen Variablennamen an.
3. Drücken Sie [ENTER]. Nach dem Ausfüllen eines Eingabefelds, wie beispielsweise Variable, müssen Sie [ENTER] zweimal drücken.



Hinweis: Starten Sie im Graphikbildschirm, müssen Sie Type=GDB einstellen.

Eine Graphik-Einstellungen öffnen

Achtung: Wenn Sie Graphik-Einstellungen öffnen, werden sämtliche in den aktuellen Daten enthaltenen Informationen ersetzt. Sie sollten die aktuelle Graphik-Einstellung speichern, bevor Sie eine gespeicherte Einstellung öffnen.

Ausgangspunkt ist der Y= Editor, Window-Editor, Tabellen-Bildschirm oder der Graphikbildschirm:

1. Drücken Sie [F1], und wählen Sie **1:Open**.
2. Wählen Sie das Verzeichnis und die Variable mit den zu öffnenden Graphik-Einstellungen.
3. Drücken Sie [ENTER].

Hinweis: Starten Sie im Graphikbildschirm, müssen Sie Type=GDB einstellen.

Eine Graphik-Einstellung löschen

Unbenötigte GDB-Variablen belegen Speicherplatz im Rechner. Verwenden Sie gemäß der Beschreibung in *Speicher und Variablen-Verwaltung* den VAR-LINK-Bildschirm ([2nd] [VAR-LINK]), um sie zu löschen.

Von einem Programm aus oder im Hauptbildschirm

Verwenden Sie zum Sichern (speichern) und öffnen (abrufen) einer Graphikdatenbank die im Modul *Technische Referenz* beschriebenen Befehle **StoGDB** und **RclGDB**.

Split-Screens

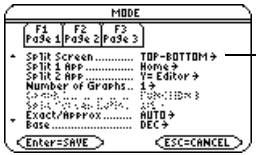
Den Split-Screen-Modus einstellen und beenden

Bestimmen Sie die gewünschten Modus-Einstellungen bei der Einrichtung eines geteilten Bildschirms über das MODE-Dialogfeld. Nach Einrichtung des geteilten Bildschirms bleibt dieser bis zu weiteren Änderungen aktiv.

Den Split-Screen-Modus einstellen

1. Drücken Sie MODE, um das **MODE**-Dialogfeld zu öffnen.
2. Die Modi für die geteilten Bildschirme sind auf der zweiten Seite des **MODE**-Dialogfelds aufgelistet. Deshalb:
   - Verwenden Sie ⊙ zum Abwärtsscrollen.
     — oder —
   - Drücken Sie F2 zum Öffnen der Seite 2.
3. Nehmen Sie für den **Split-Screen**-Modus eine der beiden folgenden Einstellungen vor. Näheres über die Vorgehensweise zum Ändern einer Modus-Einstellung finden Sie in *Bedienung des Taschenrechners*.

Split-Screen-Einstellungen
TOP-BOTTOM (OBEN-UNTEN)
LEFT-RIGHT (LINKS-RECHTS



Bei der Einstellung Split Screen = TOP-BOTTOM oder LEFT-RIGHT werden zuvor unscharf angezeigte Modi, wie z.B. Split 2 App aktiv.

Die Anfangs-Anwendungen einstellen

Bevor Sie das MODE-Dialogfeld mit $\bar{x}$ schließen, können Sie anhand der **Split 1 App-** und **Split 2 App-**Modi die gewünschten Anwendungen wählen.

Modus	Bestimmt die Anwendung im:
Split 1 App	Oberen oder linken Teil des geteilten Bildschirms.
Split 2 App	Unteren oder rechten Teil des geteilten Bildschirms.

Stellen Sie **Split 1 App** und **Split 2 App** auf dieselbe Anwendung ein, verläßt der Handheld den Split-Screen-Modus und zeigt die Anwendung auf dem vollen Bildschirm an.

Sie können, nach Anzeige des geteilten Bildschirms verschiedene Anwendungen öffnen.

Hinweis: Bei dem in *Weitere Darstellungsarten* beschriebenen Zwei-Graph-Modus kann sich dieselbe Anwendung in beiden Hälften des geteilten Bildschirms befinden.

Weitere Modi, die einen geteilten Bildschirm beeinflussen

Modus	Beschreibung
Number of Graphs **Hinweis:** Lassen Sie diese Einstellung solange auf 1, bis Sie den entsprechenden Abschnitt in *Weitere Darstellungsarten* gelesen haben.	Dient zum Einrichten und Anzeigen zweier unabhängiger Graphensätze. Dies ist eine fortgeschrittene Graphikfunktion, die unter “Den Zwei-Graph-Modus verwenden” in *Weitere Darstellungsarten* beschrieben wird.

Geteilte Bildschirme und Pixel-Koordinaten

Der Handheld verfügt über Befehle, die zum Zeichnen von Linien, Kreisen etc. auf dem Graphik-Bildschirm mit Pixel-Koordinaten arbeiten. Folgende Tabelle zeigt, wie die Modus-Einstellungen für **Split Screen-** und **Split Screen-Ratio** die auf dem Graphik-Bildschirm verfügbare Pixelanzahl beeinflussen.

Hinweis:

- Eine Liste der Zeichenbefehle finden Sie unter “Im Graphik-Bildschirm zeichnen” in *Programmierung*.
- Aufgrund des Rahmens um die jeweils aktive Anwendung verfügen geteilte Bildschirme über eine kleinere Anzeigefläche als der nicht geteilte Bildschirm.

TI-89 Titanium:

		Split 1 App		Split 2 App	
Teilen	**Verh.**	**x**	**y**	**x**	**y**
FULL	N/A	0 – 158	0 – 76	N/A	N/A
TOP–BOTTOM	1:1	0 – 154	0 – 34	0 – 154	0 – 34
LEFT–RIGHT	1:1	0 – 76	0 – 72	0 – 76	0 – 72

Voyage™ 200:

		Split 1 App		Split 2 App	
Teilen	**Verh.**	**x**	**y**	**x**	**y**
FULL	N/A	0 – 238	0 –102	N/A	N/A
TOP–BOTTOM	1:1	0 – 234	0 – 46	0 – 234	0 – 46
	1:2	0 – 234	0 – 26	0 – 234	0 – 68
	2:1	0 – 234	0 – 68	0 – 234	0 – 26
LEFT–RIGHT	1:1	0 – 116	0 – 98	0 –116	0 – 98
	1:2	0 – 76	0 – 98	0 – 156	0 – 98
	2:1	0 – 156	0 – 98	0 – 76	0 – 98

Den Split-Screen-Modus beenden

Method 1: Drücken Sie [MODE], um das MODE-Dialogfeld zu öffnen. Nehmen Sie dann die Einstellung **Split Screen = FULL** vor. Wenn Sie zum Schließen des Dialogfelds x̄ drücken, zeigt der ungeteilte Bildschirm die für **Split 1 App** angegebene Anwendung an.

Method 2: Drücken Sie [2nd] [QUIT] zweimal, um den Hauptbildschirm in voller Größe zu öffnen.

Beim Ausschalten des Handhelds

Durch das Ausschalten des Handhelds wird der Split-Screen-Modus nicht beendet.

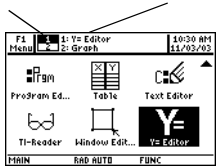
Handheld wurde ausgeschaltet:	Wenn Sie den Handheld wieder einschalten:
Durch Betätigung von [2nd] [OFF]	Der geteilte Bildschirm ist weiterhin wirksam, doch wird an Stelle der Anwendung, die vor der Betätigung von [2nd] [OFF] aktiv war, stets der Hauptbildschirm angezeigt.
Durch die Abschaltautomatik (APD™) oder durch Betätigung von [♦] [OFF].	Der geteilte Bildschirm befindet sich in dem Zustand, in welchen Sie ihn verlassen haben.

Die Split-Screen-Statusanzeige auf der Apps-Arbeitsfläche

Um zur Apps-Arbeitsflächezurückzukehren, drücken Sie APPS. Oben auf der Apps-Arbeitsfläche wird die Split-Screen-Statusanzeige mit den Namen der offenen Apps und den Displaybereichen angezeigt, in denen die betreffende App angezeigt wird.

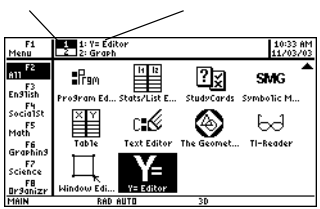
Hinweis: Die Apps-Arbeitsfläche wird immer in der Vollbildansicht angezeigt.

Split-Screen-Indikator Namen der offenen Apps



Split-Screen-Statusanzeige auf der Apps-Arbeitsfläche des TI-89 Titanium

Split-Screen-Indikator Namen der offenen Apps



Split-Screen-Statusanzeige auf der Apps-Arbeitsfläche des Voyage™ 200

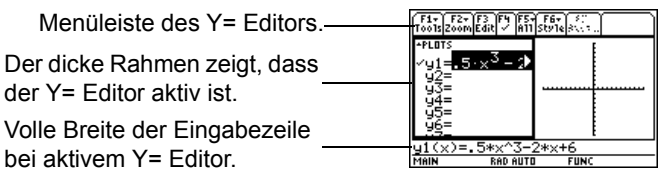
Split-Screen-Statusanzeige	Beschreibung
	Oben-Unten Split-Screen • **1** gibt die Anwendung an, die im oberen Bereich des Displays angezeigt wird. • **2** gibt die Anwendung an, die im unteren Bereich des Displays angezeigt wird. Die hervorgehobene Nummer kennzeichnet den aktiven Bereich des geteilten Bildschirms.
	Links-Rechts Split-Screen • **1** gibt die Anwendung an, die im linken Bereich des Displays angezeigt wird. • **2** gibt die Anwendung an, die im rechten Bereich des Displays angezeigt wird. Die hervorgehobene Nummer kennzeichnet den aktiven Bereich des geteilten Bildschirms.

Die aktive Anwendung wählen

Bei einem geteilten Bildschirm kann nur jeweils eine der beiden Anwendungen aktiv sein. Sie können problemlos von einer Anwendung zur anderen umschalten oder auch eine neue Anwendung öffnen.

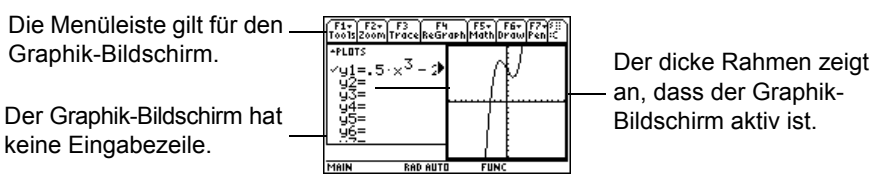
Die aktive Anwendung

- Die aktive Anwendung wird durch einen dicken Rahmen gekennzeichnet.
- Menüleiste und Statuszeile nehmen stets die gesamte Displaybreite ein und beziehen sich auf die aktive Anwendung.
- Bei Anwendungen mit Eingabezeile (wie beispielsweise der Hauptbildschirm und der Y= Editor) nimmt die Eingabezeile nur dann die gesamte Displaybreite ein, wenn diese Anwendung aktiv ist.



Von einer Anwendung auf eine andere umschalten

Drücken Sie [2nd] [⊞] (Zweitfunktion von [APPS]) zum Umschalten von einer Anwendung auf eine andere.



Eine neue Anwendung öffnen

Methode 1:

1. Wechseln Sie mit 2nd [⊡] zu der Anwendung, die ersetzt werden soll.
2. Wählen Sie mit APPS oder ON (beispielsweise ON [WINDOW]) die neue Anwendung aus.

Wählen Sie dabei eine Anwendung, die bereits geöffnet ist, schaltet der TI-89 / Voyage™ 200 auf diese um.

Methode 2:

3. Drücken Sie MODE und dann F2.
4. Ändern Sie **Split 1 App** und/oder **Split 2 App**.

   Stellen Sie **Split 1 App** und **Split 2 App** auf dieselbe Anwendung ein, beendet der TI-89 / Voyage™ 200 den **Split-Screen-**Modus und zeigt die Anwendung auf dem ungeteilten Bildschirm an.

Hinweis: Bei dem in *Weitere Darstellungsarten* beschriebenen Zwei-Graph-Modus kann sich dieselbe Anwendung in beiden Hälften des geteilten Bildschirms befinden.

Den Hauptbildschirm mit 2nd Quit öffnen

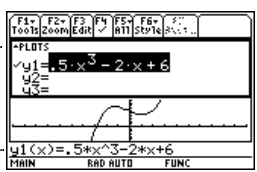
Hinweis: Durch zweimalige Betätigung von 2nd [QUIT] wird der Split-Screen-Modus stets beendet.

Hauptbildschirm:	**Wirkung von 2nd [QUIT]:**
Ist noch nicht geöffnet	Der Hauptbildschirm wird an Stelle der aktiven Anwendung geöffnet.
Ist geöffnet, ist aber nicht die aktive Anwendung	Umschalten auf den Hauptbildschirm, der dadurch zur aktiven Anwendung wird.
Ist die aktive Anwendung	Beendet den Split-Screen-Modus und öffnet den nicht geteilten Hauptbildschirm.

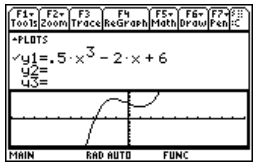
Bei Unterteilung in obere und untere Hälfte

Beachten Sie auch bei Auswahl einer TOP-BOTTOM-Unterteilung (Unterteilung in obere und untere Hälfte), dass sich Eingabezeile und Menüleiste stets auf die aktive Anwendung beziehen. Beispiel:

Die Eingabezeile gilt für den aktiven Y=Editor, *nicht* für den Grafikbildschirm.



Die Menüleiste gilt für den aktiven Grafikbildschirm, *nicht* für den Y=Editor.



Hinweis: Für **Top-Bottom-** und **Left-Right-**Unterteilungen gilt dieselbe Methode zum Wählen einer Anwendung.

Daten/Matrix-Editor

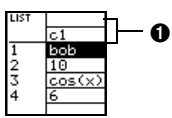
Überblick über Listen-, Daten- und Matrizenvariablen

Zur effektiven Verwendung des Daten/Matrix-Editors ist es erforderlich, dass Sie die Bedeutung von Listen-, Daten- und Matrizenvariablen verstehen.

Listenvariable

Eine Liste ist eine Folge von Objekten (Zahlen, Termen oder Zeichenfolgen), die miteinander in Verbindung stehen können oder nicht. Jedes Objekt wird Element genannt. Im Daten/Matrix-Editor gilt für eine Listenvariable:

- Sie wird als einzelne Spalte angezeigt, deren Elemente sich jeweils in einer einzelnen Zelle befinden.
- Sie muß fortlaufend sein; freie oder leere Zellen sind innerhalb der Liste nicht erlaubt.
- Sie kann aus bis zu 999 Elementen bestehen.
-

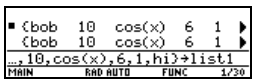


❶ Titel-(oben) und Kopfzeile-Zellen (unten) werden nicht als Teil der Liste ge-speichert.

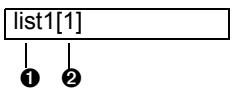
Geben Sie mehr als eine Spalte mit Elementen in eine Listenvariable ein, wird diese automatisch in eine Datenvariable umgewandelt.

Im Hauptbildschirm (oder überall dort, wo Sie eine Liste verwenden können) können Sie eine Liste als eine Folge von Elementen eingeben, die in geschweifte Klammern { } gefaßt und durch Kommas getrennt sind.

Zur Trennung der Elemente müssen Sie bei der Eingabe in die Eingabezeile zwar Kommata verwenden, aber im Protokoll-Bereich werden die Elemente durch Leerstellen getrennt angezeigt.



Auf ein bestimmtes Listenelement beziehen Sie sich mit Hilfe des rechts abgebildeten Formats.



❶ Name der Listenvariablen; ❷ Elementnummer (oder Indexnummer)

Hinweis: Wenn Sie im Daten/Matrix-Editor eine Liste erstellt haben, können Sie diese Liste in jeder Anwendung (z.B. dem Hauptbildschirm) verwenden.

Datenvariable

Bei einer Datenvariablen handelt es sich im Grunde genommen um eine Sammlung von Listen, die miteinander in Verbindung stehen können oder nicht. Im Daten/Matrix-Editor gilt für eine Datenvariable:

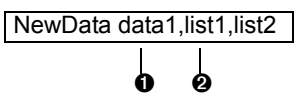
- Sie kann bis zu 99 Spalten haben.
- Sie kann in jeder Spalte bis zu 999 Elemente enthalten. Je nach Datenart müssen die Spalten nicht alle gleich lang sein.
- Die Spalten müssen fortlaufend sein; freie oder leere Zellen sind innerhalb einer Spalte nicht erlaubt.

DATA

	c1	c2	c3
1	fred	stone	95
2	sally	ross	75
3	jane	smith	97
4	nick	castle	83

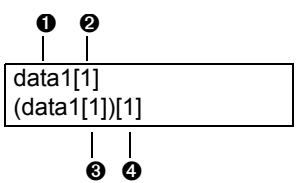
Hinweis: Für statistische Berechnungen müssen die Spalten die gleiche Länge besitzen.

Auf dem Hauptbildschirm oder von einem Programm aus können Sie mit dem Befehl **NewData** eine Datenvariable aus bereits vorhandenen Listen erstellen.



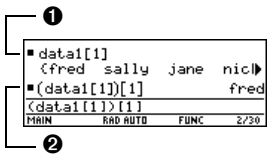
❶ Name der zu erstellenden Datenvariablen
❷ Namen vorhandener Listenvariablen.

Obwohl man eine Datenvariable im Hauptbildschirm nicht direkt anzeigen kann, kann man sich einzelne Spalten oder Elemente ansehen.



❶ Name der Datenvariablen
❷ Spaltennummer
❸ Spaltennummerr
❹ Elementnummer in der Spalte

Beispiel:

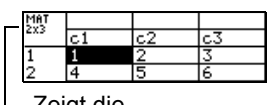


❶ Zeigt Spalte 1 der Datenvariablen 1 an.
❷ Zeigt Element 1 in Spalte 1 der Datenvariablen 1 an.

Matrizenvariable

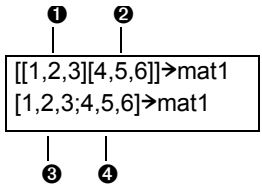
Eine Matrix ist ein rechteckiges Feld aus Elementen. Bei der Erstellung einer Matrix im Daten/Matrix-Editor müssen Sie die Anzahl der Zeilen und Spalten festlegen (Sie können jedoch später Zeilen und Spalten hinzufügen oder entfernen). Im Daten/Matrix-Editor gilt für eine Matrizenvariable:

- Sie sieht ähnlich wie eine Datenvariable aus, aber alle Spalten müssen gleich lang sein.
- Bei ihrer Erstellung erscheint der Wert 0 in jeder Zelle. Sie können dann 0 durch den gewünschten Wert ersetzen.



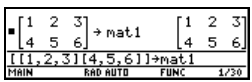
Zeigt die Matrizengröße.

Mit [STO▸] können Sie vom Hauptbildschirm oder von einem Programm aus eine Matrix in einer der nebenstehenden gleichwertigen Formen speichern.



❶ Zeile 1
❷ Zeile 2
❸ Zeile 1
❹ Zeile 2

Sie geben die Matrix zwar in der oben gezeigten Form ein, aber im Protokoll-Bereich wird sie in der herkömmlichen Matrizenform angezeigt.



Wenn Sie im Daten/Matrix-Editor eine Matrix erstellt haben, können Sie diese Matrix in jeder Anwendung (z.B. dem Hauptbildschirm) verwenden.

Hinweis: Verwenden Sie Klammern, um ein bestimmtes Element in einer Matrix anzugeben. Hier ein Beispiel: Geben Sie **mat1[2,1]** ein, um auf das erste Element in der zweiten Zeile zuzugreifen.

Eine Daten/Matrix-Editor-Sitzung beginnen

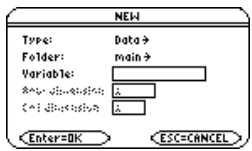
Jedesmal, wenn Sie den Daten/Matrix-Editor starten, können Sie eine neue Variable erstellen, die aktuelle Variable (die bei der letzten Verwendung des Daten/Matrix-Editors zuletzt angezeigte) weiterverwenden oder eine bestehende Variable öffnen.

Eine neue Daten-, Matrizen- oder Listenvariable erstellen

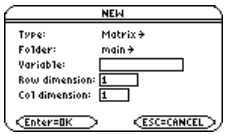
1. Drücken Sie APPS, und wählen Sie **Data/Matrix Editor**.

2. Wählen Sie **3:New**.

3. Machen Sie die entsprechenden Angaben für die neue Variable.



Menüpunkt	Dient zum:
Type	Wählen der zu erstellenden Variablenart. Drücken Sie ▶, um ein Menü mit den verfügbaren Typen zu öffnen.
Folder	Wählen des Verzeichnisses, in welchem die neue Variable gespeichert wird. Drücken Sie ▶, um ein Menü mit den vorhandenen Verzeichnissen zu öffnen. Nähere Informationen zu Verzeichnissen finden Sie in *Weitere Funktionen des Hauptbildschirms*.
Variable	Geben Sie einen Namen für die neue Variable ein. Bei Angabe einer bereits vorhandenen Variablen wird eine Fehlermeldung angezeigt, sobald Sie ENTER drücken. Wenn Sie die Anzeige des Fehlers mit ESC oder ENTER quittieren, wird wieder das Dialogfeld **NEW** angezeigt.

Menüpunkt	Dient zum:	
Row dimension und Col dimension	Wenn Type = Matrix, geben Sie Zeilen- und Spaltenanzahl der Matrix ein.	

Hinweis: Wenn Sie keinen Variablennamen eingeben, öffnet der TI-89 Titanium / Voyage™ 200 Graphischer Rechner den Hauptbildschirm.

4. Drücken Sie ENTER (nach Ausfüllen eines Eingabefeldes wie Variable drücken Sie ENTER zweimal), um eine leere **Variable** im Daten/Matrix-Editor zu erstellen und anzuzeigen.

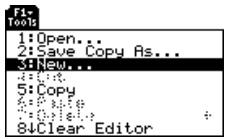
Die aktuelle Variable verwenden

Sie können den Daten/Matrix-Editor jederzeit verlassen, um zu einer anderen Anwendung zu gehen. Um zu der Variablen zurückzukehren, die angezeigt wurde, als Sie den Daten/Matrix-Editor verlassen haben, rufen Sie den Daten/Matrix-Editor erneut auf und wählen **1:Current**.

Eine neue Variable im Daten/Matrix-Editor erstellen

Im Daten/Matrix-Editor:

1. Drücken Sie F1, und wählen Sie **3:New**.
2. Geben Sie den Typ, das Verzeichnis und den Variablennamen an. Geben Sie bei einer Matrix auch die Zeilen- und Spaltenanzahl an.



Eine andere Variable öffnen

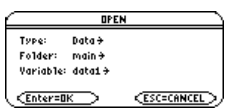
Sie können jederzeit eine andere Variable öffnen.

1. Drücken Sie vom Daten/Matrix-Editor aus F1, und wählen Sie **1:Open**.

   – oder –

   Drücken Sie in einer beliebigen Anwendung APPS, und wählen Sie **2:Open**.

2. Wählen Sie den Typ, das Verzeichnis und die zu öffnende Variable.
3. Drücken Sie ENTER.



Hinweis: Standardmäßig zeigt **Variable** die erste in der alphabetischen Liste vorhandene Variable an. Sind keine Variablen vorhanden, ist das Feld leer.

Hinweis zum Löschen einer Variablen

Da sämtliche Variablen des Daten/Matrix-Editors automatisch gespeichert werden, können sich relativ viele Variablen ansammeln, und es wird u.U. viel Speicherplatz belegt.

Zum Löschen einer Variablen verwenden Sie den VAR-LINK -Bildschirm ([2nd] [VAR-LINK]). Näheres zu VAR-LINK finden Sie in *Speicher-und Variablen-Verwaltung*.

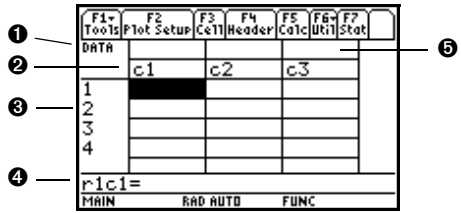
Zellenwerte eingeben und ansehen

Wenn Sie eine neue Variable erstellen, ist der Daten/Marix-Editor anfangs leer (für Listen- oder Datenvariablen) oder mit Nullen ausgefüllt (für eine Matrix). Wenn Sie eine vorhandene Variable öffnen, werden deren Werte angezeigt. Sie können dann entweder zusätzliche Werte eingeben oder die vorhandenen bearbeiten.

Der Bildschirm des Daten/Matrix-Editors

Im folgenden ist ein leerer Daten/Matrix-Editor dargestellt. Beim ersten Öffnen des Bildschirms markiert der Cursor die Zelle in Zeile 1, Spalte 1.

❶ Variablentyp
❷ Kopfzeile
❸ Zeilennummer
❹ Zeilen- und Spaltennummer der markierten Zelle
❺ Spaltentitel-Zellen zur Eingabe eines Titels für jede Spalte



Hinweis: Verwenden Sie die Titelzelle über jeder Spalte zur Bezeichnung der in der Spalte befindlichen Daten.

Wenn Werte eingegeben werden, zeigt die Eingabezeile den vollständigen Wert der markierten Zelle an.

Einen Wert eingeben oder bearbeiten

Sie können einen beliebigen Term in eine Zelle eingeben (Zahl, Variable, Funktion, Zeichenfolge etc.).

1. Bewegen Sie den Cursor zu der gewünschten Zelle, um diese für die Eingabe bzw. die Bearbeitung zu markieren.
2. Drücken Sie [ENTER] oder [F3], um den Cursor in die Eingabezeile zu setzen.
3. Geben Sie einen neuen Wert ein, oder bearbeiten Sie den bereits vorhandenen.
4. Drücken Sie [ENTER], um den Wert in die markierte Zelle zu übertragen.

Wenn Sie [ENTER] drücken, rückt der Cursor automatisch zur nächsten Zelle, und Sie können mit der Eingabe oder Bearbeitung von Werten fortfahren. Die Bewegungsrichtung des Cursors hängt von der Variablenart ab.

Hinweis: Sie können einen neuen Wert eingeben, ohne zuvor [ENTER] oder [F3] zu drücken. Zur Bearbeitung einer vorhandenen Variablen müssen Sie aber [ENTER] oder [F3] verwenden.

Variablentyp	**Nach Betätigung von [ENTER], bewegt sich der Cursor:**
List oder data	Abwärts, zur Zelle in der nächsten Zeile.
Matrix	Nach rechts, zur Zelle in der nächsten Spalte. Von der letzten Zelle einer Zeile rückt der Cursor automatisch zur erste Zelle der nächsten Zeile weiter. Dadurch können Sie Werte in Zeile 1, Zeile 2 etc. eingeben.

Im Editor scrollen

Bewegung des Cursors:	**Drücken Sie:**
Um je eine Zelle	⊖, ⊕, ⊳, oder ⊲
Um je eine Seite	[2nd] und dann ⊖, ⊕, ⊳, oder ⊲
Zu Zeile 1 in der aktuellen Spalte bzw. zur letzten Zeile des Bildschirms, die Daten enthält. Steht der Cursor in oder hinter dieser letzten Zeile, wird er mit [♦] ⊖ auf Zeile 999 bewegt.	[♦] ⊕ oder [♦] ⊖

Bewegung des Cursors:	Drücken Sie:
Zu Spalte 1 bzw. zur letzten Spalte, die Daten enthält. Befindet sich der Cursor in oder hinter dieser letzten Spalte, wird er mit ◆ ⓑ zu Spalte 99 bewegt.	◆ ⓐ oder ◆ ⓑ

Hinweis: Für die Übertragung eines Wertes aus der Eingabezeile können Sie auch ⓓ oder ⓒ verwenden.

Wenn Sie abwärts/aufwärts scrollen, bleibt die Kopfzeile am oberen Bildschirmrand stehen, so dass die Spaltennummern stets sichtbar sind. Wenn Sie nach rechts/links scrollen, bleiben die Zeilennummern am linken Bildschirmrand stehen, so dass sie stets sichtbar sind.

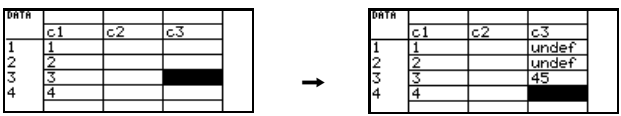
So werden Zeilen und Spalten automatisch gefüllt

Wenn Sie in eine Zelle einen Wert eingeben, wandert der Cursor zur nächsten Zelle. Sie können den Cursor jedoch zu jeder beliebigen Zelle bewegen, um dort einen Wert einzugeben. Wenn Sie zwischen einzelnen Zellen Zwischenräume lassen, werden diese vom Gerät automatisch behandelt.

- Bei Listenvariablen bleibt eine Lückenzelle solange unbestimmt (Eintrag undef), bis Sie einen Wert für diese Zelle eingeben.

Hinweis: Wenn Sie mehr als eine Spalte mit Elementen in eine Listenvariable eingeben, wird diese automatisch in eine Datenvariable umgewandelt.

- Bei einer Datenvariablen werden Lücken in einer Spalte genau wie in einer Liste behandelt. Bei Lücken zwischen zwei Spalten entsteht eine leere Spalte.



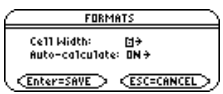
- Matrizenvariablen werden automatisch weitere Zeilen und/oder Spalten hinzugefügt, wenn Sie bei der Eingabe eines Wertes in eine Zelle die aktuellen Grenzen überschreiten. Die übrigen Zellen in den neuen Zeilen und/oder Spalten werden mit Nullen aufgefüllt.

Hinweis: Bei der Erstellung einer Matrix wird zwar deren Größe angegeben, es können aber problemlos Zeilen und/oder Spalten hinzugefügt werden.

Die Zellenbreite ändern

Die Zellenbreite bestimmt, wie viele Zeichen in einer Zelle angezeigt werden. So ändern Sie die Zellenbreite im Daten/Matrix-Editor:

1. Aktivieren Sie das Dialogfeld **FORMATS** wie folgt: Drücken Sie:
   F1 **9**
   – oder –
   ♦ |
   ♦ **F**



Die Zellenbreite stellt die maximale Anzahl der in einer Zelle anzeigbaren Zeichen dar. Alle Zellen sind gleich breit.

   Hinweis: Möchten Sie eine Zahl vollständig sehen, können Sie die Zelle jederzeit markieren; die Zahl erscheint dann in der Eingabezeile.

2. Markieren Sie die aktuelle **Cell Width**-Einstellung, und drücken Sie ▶ oder ◀, um das Ziffernmenü (**3** bis **12**) zu öffnen.
3. Bewegen Sie den Cursor zur gewünschten Zahl, um diese zu markieren, und drücken Sie ENTER. (Einstellige Zahlen können Sie auch über die Tastatur eingeben; drücken Sie dann ENTER).
4. Zum Schließen des Dialogfeldes drücken Sie ENTER.

Den Spalteninhalt löschen

Durch diesen Vorgang wird der Inhalt einer Spalte gelöscht. Die Spalte wird nicht entfernt.

Löschen des Inhalts:	Vorgehensweise:
Einer Spalte	1. Setzen Sie den Cursor auf eine beliebige Zelle der Spalte. 2. Drücken Sie: [2nd] [F6] [F6] und wählen Sie **5:Clear Column**. (Diese Option steht für Matrizen nicht zur Verfügung.)
Aller Spalten	Drücken Sie [F1], und wählen Sie **8:Clear Editor**. Wenn Sie aufgefordert werden, den Vorgang zu bestätigen, drücken Sie [ENTER] (oder [ESC] zum Annullieren).

Hinweis: Bei Listen- oder Datenvariablen sind gelöschte Spalten leer. Bei Matrizen ist eine gelöschte Spalte mit Nullen ausgefüllt.

Eine Kopfzeile mit einem Term definieren

Bei einer Listenvariablen oder einer Spalte einer Datenvariablen können Sie in den Kopfzeile eine Funktion eingeben, die automatisch eine Liste mit Elementen erzeugt. Eine Spalte einer Datenvariable kann außerdem durch eine andere Spalte definiert werden.

Eine Kopfzeiledefinition eingeben

Im Daten/Matrix-Editor:

1. Setzen Sie den Cursor auf eine beliebige Zelle in der Spalte, und drücken Sie F4.

   – oder –

   Setzen Sie den Cursor auf die Kopfzelle (**c1, c2** etc.), und drücken Sie ENTER.

   Hinweis:

   - ENTER ist dann nicht erforderlich, wenn Sie eine neue Definition eingeben oder die bereits vorhandene ersetzen möchten. Wenn Sie die vorhandene Definition nur ändern möchten, müssen Sie jedoch ENTER drücken.
   - Wenn Sie eine vorhandene Definition ansehen möchten, drücken Sie F4, oder setzen Sie den Cursor auf die Kopfzelle; die Definition wird in der Eingabezeile angezeigt.

2. Geben Sie den neuen Term ein, der die vorhandene Definition ersetzt.

   Wenn Sie in Schritt 1 F4 oder ENTER verwendet haben, steht der Cursor in der Eingabezeile und hat die Definition (falls vorhanden) markiert. Sie können auch:

   - CLEAR drücken, um den markierten Term zu löschen. Geben Sie dann den neuen Term ein.

     – oder –

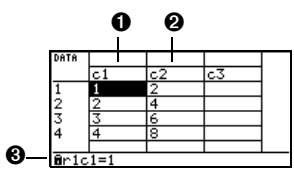
- ◀ oder ▶ drücken, um die Markierung aufzuheben. Bearbeiten Sie dann den alten Term.

Hinweis: Änderungen können Sie rückgängig machen, indem Sie ESC vor ENTER drücken.

Sie können einen Term verwenden, der:	**Beispiel:**
Eine Zahlenreihe erzeugt.	c1=seq(x^2,x,1,5) c1={1,2,3,4,5}
Sich auf eine andere Spalte bezieht.	c2=2∗c1 c4=c1∗c2–sin(c3)

Hinweis: Die Funktion seq wird im Modul *Technische Referenz* beschrieben. Beziehen Sie sich auf eine leere Spalte, wird eine Fehlermeldung angezeigt (es sei denn **Auto-calculate = OFF.** .

3. Drücken Sie ENTER, ▼ oder ▲, um die Definition zu speichern und die Spalten zu aktualisieren.



❶ c1=seq(x,x,1,7); ❷ c2=2∗c1; ❸ Eine gesperrte Zelle (🔒) kann nicht direkt geändert werden, da sie durch die Kopfzeile definiert ist.

Hinweis: Bei Datenvariablen werden Definitionen der Kopfzeile gespeichert, sobald Sie den Daten/Matrix-Editor verlassen. Bei Listenvariablen wird die Definition nicht gespeichert (nur die sich ergebenden Zellenwerte).

Eine Definition der Kopfzeile löschen

1. Setzen Sie den Cursor auf eine beliebige Zelle der Spalte, und drücken Sie F4.

   – oder –

   Setzen Sie den Cursor auf die Kopfzelle (**c1, c2** etc.), und drücken Sie ENTER.
2. Drücken Sie CLEAR, um den markierten Term zu löschen.
3. Drücken Sie ENTER, ⏷ oder ⏶.

Eine vorhandene Liste als Spalte verwenden

Wenn Sie eine oder mehrere Listen haben, können Sie diese auch als Spalten in einer Datenvariablen verwenden.

Im:	Vorgehensweise:
Data/Matrix Editor	Verwenden Sie in der gewünschten Spalte F4, um die Kopfzeile zu definieren. Beziehen Sie sich auf die vorhandene Listenvariable. Beispiel: c1=list1
Hauptbildschirm oder einem Programm	Verwenden Sie gemäß der Beschreibung im Modul *Technische Referenz* den Befehl **NewData**. Beispiel: **NewData** *datavar, list1* [*, list2*] [*, list3*] ... ❶ ❷

Im:	Vorgehensweise:
❶ Datenvariable. Liegt diese Datenvariable bereits vor, wird sie auf Grundlage der angegebenen Liste neu definiert. ❷ Vorhandene Listenvariablen, die in Spalten der Datenvariablen kopiert werden sollen.	

Hinweis: Ist Ihr Gerät mit dem Zusatz CBL 2™ oder CBR™, ausgestattet, verwenden Sie dieses Verfahren für Ihre erfaßten Listen. Vorhandene Listenvariablen können Sie mit 2nd [VAR-LINK] anzeigen.

Eine Matrix mit einer Liste füllen

Zum Füllen einer Matrix mit einer Liste können Sie nicht den Daten/ Matrix-Editor verwenden. Statt dessen können Sie vom Hauptbildschirm oder von einem Programm aus den Befehl **list▶mat** verwenden. Näheres finden Sie im Modul *Technische Referenz*.

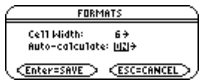
Die Auto-Calculate-Funktion

Der Daten/Matrix-Editor verfügt über eine Auto-calculate-Funktion für Listen- und Datenvariablen. Grundeinstellung ist Auto-calculate = ON. Deshalb werden bei Änderungen, die eine Definition der Kopfzeile betreffen (oder eine Spalte, auf die sich in einer Kopfdefinition bezogen wird), sämtliche Spalten automatisch neu berechnet. Beispiel:

- Wenn Sie eine Ddefinition der Kopfzeileändern, wird die neue Definition automatisch angewendet.
- Wenn der Kopf von Spalte 2 als c2=2∗c1 definiert ist, wirkt sich jede Änderung in Spalte 1 automatisch auf Spalte 2 aus.

So schalten Sie Auto-calculate im Daten/Matrix-Editor ein und aus:

1. Drücken Sie:
   F1 **9**
   – oder –
   ♦ |
   ♦ **F**



2. Stellen Sie **Auto-Calculate** auf **OFF** oder **ON**.
3. Drücken Sie ENTER, um das Dialogfenster zu schließen.

Wenn **Auto-calculate = OFF**, und Sie nehmen wie oben beschrieben Änderungen vor, dann werden die Definitionen der Kopfzeile solange nicht neu berechnet, bis Sie **Auto-calculate = ON** einstellen.

Hinweis: Stellen Sie **Auto-calculate = OFF**, um mehrere Änderungen zu machen, ohne dass jedes Mal eine Neuberechnung stattfindet. Sie können eine Definition, wie bspw. c1=c2+c3, eingeben, bevor die Spalten 2 und 3 eingegeben werden, etwaige Fehler in einer Definition übergehen, bevor diese Fehler behoben werden können.

Die Funktionen Shift und CumSum in einer Kopfzeile verwenden

Beim Definieren einer Kopfzeile können Sie wie unten beschrieben die Funktionen shift und **cumSum** verwenden. Diese Beschreibungen weichen leicht vom Modul *Technische Referenz* ab. In diesem Abschnitt wird die Verwendung dieser Funktionen im Daten/Matrix-Editor dargestellt. Im Modul *Technische Referenz* finden Sie eine allgemeinere Erklärung für den Hauptbildschirm oder ein Programm.

Die Funktion Shift verwenden

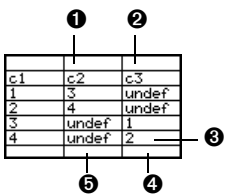
Mit der Funktion **shift** können Sie eine Spalte kopieren und um eine bestimmte Elementanzahl auf- oder abwärts verschieben. Verwenden Sie F4, um mit folgender Syntax einen Spaltenkopf zu definieren:

shift (*column* [,*integer*]**)**

❶ ❷

❶ Spalte für die Verschiebung.
❷ Elementenanzahl, um die verschoben werden soll (positiv = aufwärts; negativ = abwärts). Grundeinstellung ist -1.

Beispiel für eine Auf- und Abwärtsverschiebung um zwei Elemente:



❶ c2=shift(c1,2)
❷ c3=shift(c1, -2)
❸ Verschobene Spalten haben dieselbe Länge wie die Ausgangsspalte (c1).
❹ Die letzten beiden Elemente von c1 werden nach unten, über den Rand hinaus verschoben, undefinierte Elemente nach oben.
❺ Die ersten beiden Elemente von c1 werden nach oben, über den Rand hinaus verschoben, undefinierte Elemente nach unten.

Hinweis: Geben Sie “shift” über die Tastatur ein, oder wählen Sie “shift” aus die CATALOG.

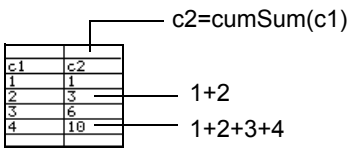
Die Funktion CumSum verwenden

Die Funktion **cumSum** liefert eine Summe der Elemente in einer Spalte. Verwenden Sie F4, um eine Kopfzeile mit folgender Syntax zu definieren:

cumSum (*Spalte***)**

└ Ausgangsspalte für die Summenbildung

Beispiel:



Hinweis: Geben Sie "**cumSum**" über die Tastatur ein, wählen Sie es aus die CATALOG, oder drücken Sie 2nd [MATH], und wählen Sie es aus dem List-Untermenü.

Spalten sortieren

Nach der Eingabe von Informationen in eine Daten-, Listen- oder Matrizenvariable können Sie eine bestimmte Spalte problemlos in numerischer oder alphabetischer Reihenfolge sortieren. Sie können auf Grundlage einer "Schlüssel"-Spalte auch alle Spalten als ein Gesamtobjekt sortieren.

Eine einzelne Spalte sortieren

Im Daten/Matrix-Editor:

1. Setzen Sie den Cursor auf eine beliebige Zelle der Spalte.
2. Drücken Sie:
   [2nd] [F6]
   [F6]
   und wählen Sie **3:Sort Column**.

   Zahlen werden aufsteigend sortiert.

   Zeichenstrings werden in alphabetischer Reihenfolge sortiert.

C1		C1
fred	→	75
sally		82
chris	→	98
jane		chris
75	→	fred
98		jane
82		sally

Alle Spalten auf Grundlage einer “Schlüssel”-Spalte sortieren

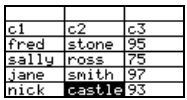
Es sei eine Datenbank-Struktur gegeben, bei welcher die Informationen in jeder Spalte innerhalb einer Zeile zusammmenhängen (wie beispielsweise der Vor- und Zuname eines Studenten und dessen Prüfungsergebnisse). In diesem Fall würde das Sortieren nur einer Spalte den Zusammenhang zwischen den verschiedenen Spalten zerstören.

Im Daten/Matrix-Editor:

1. Setzen Sie den Cursor auf eine beliebige Zelle der “Schlüssel”-Spalte.

   Setzen Sie hier den Cursor auf die zweite Spalte (**c2**), um eine Sortierung nach Nachnamen vorzunehmen.

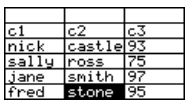
   Hinweis: Bei einer Listenvariablen kommt dies dem Sortieren einer einzelnen Spalte gleich.



c1	c2	c3
fred	stone	95
sally	ross	75
jane	smith	97
nick	castle	93

2. Drücken Sie:
   [2nd] [F6]
   [F6]
   und wählen Sie **4:Sort Col**, adjust all.

   Hinweis: Dieser Menüpunkt ist nicht verfügbar, wenn eine gesperrte Spalte vorhanden ist.



c1	c2	c3
nick	castle	93
sally	ross	75
jane	smith	97
fred	stone	95

Bei Anwendung dieses Verfahrens auf eine Datenvariable:

- Müssen alle Spalten gleich lang sein.
- Darf keine der Spalten gesperrt sein (durch eine Funktion in der Kopfzeile definiert sein). Befindet sich der Cursor in einer gesperrten Spalte, so wird am Anfang der Eingabezeile angezeigt.

Eine Kopie einer Listen-, Daten- oder Matrizenvariablen speichern

Sie können eine Kopie einer Listen-, Daten- oder Matrizenvariablen speichern. Sie können auch eine Liste in eine Datenvariable kopieren oder eine Spalte aus einer Datenvariablen wählen und diese in eine Liste kopieren.

Gültige Kopietypen

Diese Arten können kopiert werden:	In folgende Arten:
Listen	Listen oder Daten
Daten	Daten
Datenspalte	Listen
Matrizen	Matrizen

Hinweis: Eine Liste wird automatisch in eine Datenvariable umgeformt, wenn Sie mehr als eine Spalte mit Informationen eingeben.

Vorgehensweise

Im Daten/Matrix-Editor:

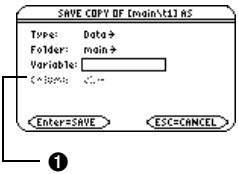
1. Zeigen Sie die zu kopierende Variable an.

2. Drücken Sie F1, und wählen Sie **2:Save Copy As**.

3. Im Dialogfeld:

   - Wählen Sie den **Type** und das Verzeichnis für die Kopie: Type und **Folder**.
   - Geben Sie für die Kopie einen Variablennamen ein.
   - Wählen Sie, wenn verfügbar, die Spalte, aus welcher kopiert werden soll.

   Hinweis: Geben Sie den Namen einer bereits bestehenden Variablen ein, wird deren Inhalt ersetzt.

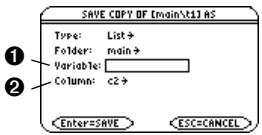


❶ Diese Zeile wird unscharf eingeblendet, es sei denn, Sie kopieren eine Datenspalte in eine Liste. Die Spalteninformation wird für andere Kopietypen nicht verwendet.

4. Drücken Sie ENTER (nach dem Ausfüllen eines Eingabefeldes wie Variable müssen Sie ENTER zweimal drücken).

Eine Datenspalte in eine Liste kopieren

Während eine Datenvariable über mehrere Spalten verfügen kann, besteht eine Listenvariable aus nur einer Spalte. Deshalb müssen Sie beim Kopieren aus einer Datenvariablen in eine Liste die zu kopierende Spalte wählen.



❶ Listenvariable, in welche kopiert werden soll.

❷ Datenspalte, die in die Liste kopiert wird. Hier wird standardmäßig die Spalte eingeblendet, in welcher sich der Cursor befindet.

Statistik und Datenplots

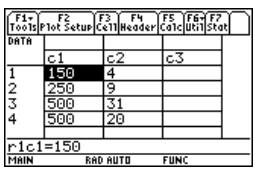
Schritte zur statistischen Analyse

Dieser Abschnitt stellt einen Überblick über die zur Durchführung einer statistischen Berechnung oder zur graphischen Darstellung eines Statistik-Plots benötigten Schritte dar. Eine genauere Beschreibung finden Sie auf den nachfolgenden Seiten.

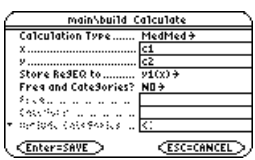
Statistische Daten berechnen und graphisch auswerten

1. Modus Graph (MODE) auf **FUNCTION** einstellen.
2. Statistische Daten im Daten/Matrix-Editor eingeben.

   Hinweis: Näheres zur Eingabe von Daten in den Daten/Matrix-Editor finden Sie in Daten/Matrix Editor.

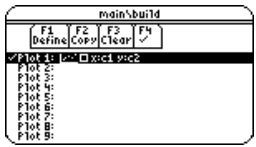


3. Statistische Berechnungen durchführen, um Statistikvariablen zu ermitteln oder Daten in ein Modell einzupassen (F5).



4. Statistik-Plots definieren und auswählen (F2 und dann F1).

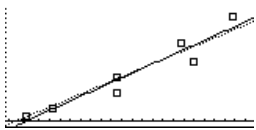
   Hinweis: Sie können Statistik-Plots und **y(x)**-Funktionen auch mit Hilfe des Y= Editors definieren und wählen.



5. Das Ansichtfenster definieren (◆ [WINDOW]).

6. Ändern Sie bei Bedarf das Graphenformat.

   F1 **9**
   — oder —
   ◆ |
   ◆ **F**



7. Die ausgewählten Statistik-Plots und Funktionen graphisch darstellen (◆ [GRAPH]).

   Hinweis: Optimieren Sie das Ansichtsfenster für Statistik-Plots mit ZoomData. F2 **Zoom** ist im Y= Editor, Window-Editor und Graph-Bildschirm verfügbar.

Die Plots untersuchen

Im Graphik-Bildschirm können Sie:

- Die Koordinaten eines beliebigen Pixels mit Hilfe des freibeweglichen Cursors anzeigen sowie die Koordinaten eines geplotteten Punkts anzeigen, indem Sie einen Plot tracen.
- Mit Hilfe des Menüleisten-Menüs F2 **Zoom** einen Abschnitt des Graphs vergrößern oder verkleinern.
- Mit Hilfe des Menüleisten-Menüs F5 **Math** eine beliebige graphisch darzustellende Funktion (jedoch keinen Plot) analysieren.

Eine statistische Berechnung durchführen

Verwenden Sie vom Daten/Matrix-Editor aus das MenüleistenMenü F5 **Calc**, um statistische Berechnungen durchzuführen. Sie können Statistiken mit einer oder mit zwei Variablen analysieren oder auch verschiedenartige Regressionsanalysen durchführen.

Das Dialogfeld Calculate

Eine Datenvariable muß geöffnet sein. Der Daten/Matrix-Editor führt mit Listen- oder Matrizenvariablen keine statistischen Berechnungen durch.

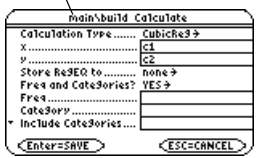
Ausgangspunkt ist der Daten/Matrix-Editor:

1. Drücken Sie F5, um das Dialogfeld **Calculate** zu öffnen.

   In diesem Beispiel sind alle Punkte aktiv. Auf Ihrem Rechner sind die Menüpunkte nur dann aktiv, wenn sie für die aktuellen Einstellungen von **Calculation Type**, **Use Freq and Categories** gültig sind.

   Hinweis: Ein für die aktuellen Einstellungen ungültiger Menüpunkt wird unscharf angezeigt. Der Cursor kann nicht auf einen unscharf angezeigten Menüpunkt gesetzt werden.

Pfadname der Datenvariablen



2. Nehmen Sie für die aktiven Menüpunkte die geeigneten Einstellungen vor.

Menüpunkt	Beschreibung
Calculation Type	Wählen Sie den Berechnungstyp.
x	Geben Sie die Nummer der im Daten/Matrix-Editor für x-Werte verwendeten Spalte (**C1**, **C2** etc.) ein. x ist die unabhängige Variable.
y	Geben Sie die Nummer der für y-Werte verwendeten Spalte ein. y ist die abhängige Variable. Dies ist für alle **Calculation Types** außer **OneVar** erforderlich.
Store RegEQ to	Wenn es sich bei Calculation Type um eine Regressionsanalyse handelt, können Sie einen Funktionsnamen wählen (**y1(x), y2(x)**, etc.). Dadurch können Sie die Regressionsgleichung speichern und im Y= Editor anzeigen.
Use Freq and Categories?	Wählen Sie **NO** oder **YES**. Beachten Sie, daß die Freq, Category und Include Categories nur bei der Einstellung Use Freq and Categories? = YES aktiv sind.
Freq	Geben Sie die Nummer der Spalte ein, die für jeden Datenpunkt eine Häufigkeit enthält. Wenn Sie keine Spaltennummer eingeben, wird für alle Datenpunkte dieselbe Gewichtung angenommen (1).
Category	Geben Sie die Nummer der Spalte ein, die eine Klasse für jeden Datenpunkt enthält.

Menüpunkt	Beschreibung
Include Categories	Wenn Sie eine Klassen-Spalte bei Category angeben, können Sie die Berechnung mit Hilfe dieses Menüpunkts auf bestimmte Klassenwerte beschränken. Legen Sie beispielsweise {1,4} fest, werden in der Berechnung nur Datenpunkte mit einem Klassenwert von 1 oder 4 verwendet.

Hinweis: Möchten Sie für x, y, Freq oder Category eine vorhandene Listenvariable verwenden, geben Sie an Stelle der Spaltennummer den Listennamen ein. Ein Beispiel für die Verwendung von Freq, Category undInclude Categories ist verfügbar.

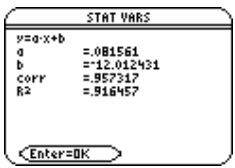
3. Drücken Sie [ENTER] (nach dem Ausfüllen eines Dialogfeldes drücken Sie [ENTER] zweimal).

   Die Ergebnisse werden auf dem **STAT VARS**-Bildschirm angezeigt. Das Format hängt von Berechnungstyp unter **Calculation Type** ab. Beispiel:

Bei Calculation Type = OneVar

Wird ▼ anstelle von = gezeigt, können Sie durch Scrollen zusätzliche Ergebnisse anzeigen.

Bei Calculation Type = LinReg



Hinweis: Bei statistischen Berechnungen werden undefinierte Datenpunkte (**undef**) ignoriert.

4. Zum Schließen des **STAT VARS**-Bildschirms drücken Sie [ENTER].

Den STAT VARS-Bildschirm erneut anzeigen

Mit dem Menü Stat des Daten/Matrix-Editors können Sie frühere Ergebnisse erneut anzeigen lassen (bis sie aus dem Speicher gelöscht werden).

[2nd] [F7]

[F7]

Frühere Ergebnisse werden gelöscht, wenn Sie:

- Die Datenpunkte bearbeiten oder den Calculation Type ändern.
- Eine andere Datenvariable öffnen oder dieselbe Datenvariable erneut öffnen (wenn sich die Berechnung auf eine Spalte in einer Datenvariablen bezogen hat). Ergebnisse werden auch dann gelöscht, wenn Sie den Daten/Matrix-Editor verlassen und mit einer Datenvariablen erneut öffnen.
- Das aktuelle Verzeichnis ändern (wenn sich die Berechnung auf eine Listenvariable im vorigen Verzeichnis bezogen hat).

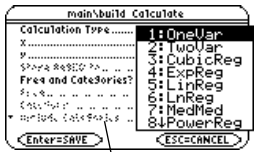
Arten der statistischen Berechnung

Wie bereits im vorigen Abschnitt beschrieben, können Sie mit dem Dialogfeld Calculate die gewünschte Art der statistischen Berechnung angeben. In vorliegendem Abschnitt finden Sie genauere Angaben zu den verschiedenen Berechnungstypen.

Den Berechnungstyp wählen

Markieren Sie im Dialogfeld Calculate (F5) die aktuelle Einstellung für Calculation Type, und drücken Sie ⊙.

Sie können dann aus einem Menü einen Typ auswählen.



Unscharf angezeigte Menüpunkte sind für den aktuellen Berechnungstyp nicht gültig.

Berechnungstyp	Beschreibung
OneVar	Statistiken mit einer Variablen —
TwoVar	Statistiken mit zwei Variablen — Berechnet die Statistikvariablen.
CubicReg	Kubische Regression — Paßt die Daten an das Polynom dritter Ordnung $\mathbf{y=ax^3+bx^2+cx+d}$ an. Es müssen mindestens vier Datenpunkte vorliegen. • Für vier Punkte ist die angepaßte Funktion ein Polynom. • Bei fünf oder mehr Punkten handelt es sich um ein Regressionspolynom.

Berechnungstyp	Beschreibung
ExpReg	Exponentialregression — Paßt die Daten an die Modellgleichung **y=ab**^{x} an (bei der a der y-Achsenabschnitt ist) und verwendet dafür die Methode der kleinsten Quadrate, sowie umgeformte **x**- und **ln(y)**-Werte.
LinReg	Lineare Regression — Paßt die Daten an das Modell **y=ax+b** an (wobei a die Steigung und b der y-Achsenabschnitt ist) und verwendet dafür die Methode der kleinsten Quadrate.
LnReg	Logarithmische Regression — Paßt die Daten an die Modellgleichung **y=a+b ln(x)** an und verwendet dafür die Methode der kleinsten Quadrate, sowie umgeformte **ln(x)**- und **y**-Werte.
Logistic	Logistische Regression — Paßt die Daten an das Modell **y=a/(1+b∗e^(c∗x))+d** an und aktualisiert alle Systemstatistik-Variablen.
MedMed	Median-Median — Paßt die Daten an das Modell **y=ax+b** an (wobei a die Steigung und b der y-Achsabschnitt ist) und verwendet dafür die Median-Median-Linie, die zum Verfahrensansatz der Ausgleichsgeraden gehört. Summenpunkte **medx1**, **medy1**, **medx2**, **medy2**, **medx3**, und **medy3** werden berechnet und in Variablen gespeichert, aber nicht auf dem STAT VARS-Bildschirm angezeigt.
PowerReg	Potenzregression — Paßt die Daten an die Modellgleichung **y=ax**^{b} an und verwendet dazu die Methode der kleinsten Quadrate sowie umgeformte **ln(x)-** und **ln(y)-**Werte.

Berechnungstyp	Beschreibung
QuadReg	Quadratische Regression — Paßt die Daten an das Polynom zweiter Ordnung **$y=ax^2+bx+c$** an. Es müssen mindestens drei Datenpunkte vorliegen. • Für drei Punkte ist die angepaßte Funktion ein Polynom. • Bei vier oder mehr Punkten handelt es sich um ein Regressionspolynom.
QuartReg	Regression vierter Ordnung — Paßt die Daten an ein Polynom vierter Ordnung **$y=ax^4+bx^3+cx^2+dx+e$** an. Es müssen mindestens fünf Datenpunkte vorliegen. • Für fünf Punkte ist die angepaßte Funktion ein Polynom. • Bei sechs oder mehr Punkten handelt es sich um ein Regressionspolynom.
SinReg	Sinusregression — Berechnet die Sinusregression und aktualisiert alle Systemstatistik-Variablen. Die Ausgabe erfolgt unabhängig von der Winkelmodus-Einstellung stets im Bogenmaß.

Hinweis: Bei **TwoVar**- und allen Regressionsrechnungen müssen die Spalten, die Sie für x und y (und ggf. für Freq oder Category) angeben, gleich lang sein.

Vom Hauptbildschirm oder von einem Programm aus

Verwenden Sie den Befehl für den gewünschten Berechnungstyp. Die Befehle haben denselben Namen wie die Berechnungstypen unter Calculation Types. Näheres zu den einzelnen Befehlen finden Sie im Modul *Technische Referenz*.

Wichtig: Mit diesen Befehlen werden zwar statistische Berechnungen durchgeführt, die Ergebnisse aber nicht automatisch angezeigt. Die Ergebnisse können Sie mit dem Befehl **ShowStat** anzeigen.

Statistikvariablen

Die Ergebnisse statistischer Berechnungen werden in Variablen gespeichert. Auf diese Variablen können Sie zugreifen, indem Sie den Variablennamen eingeben oder gemäß der Beschreibung in *Speicher-und Variablen-Verwaltung* den VAR-LINK-Bildschirm verwenden. Wenn Sie die Daten bearbeiten oder den Berechnungstyp ändern, wird der Inhalt sämtlicher Statistikvariablen gelöscht. Weitere Bedingungen, die zum Löschen des Variableninhalts führen, sind aufgeführt.

Berechnete Variablen

Statistikvariablen werden als Systemvariablen gespeichert. Beachten Sie aber, daß **regCoe**f und **regeq** als Listen- respektive Funktionsvariable behandelt werden.

	Eine Var	Zwei Var	Regressionen
Mittel von x-Werten	$\bar{x}$	$\bar{x}$	
Summe von x-Werten	Σx	Σx	
Summe von x^2-Werten	Σx^2	Σx^2	
Stichproben-Standardabweichung von x	Sx	Sx	
Grundgesamtheits-Standardabweichung von x †	σx	σx	

	Eine Var	Zwei Var	Regressionen
Anzahl der Datenpunkte	nStat	nStat	
Mittel von y-Werten		$\bar{y}$	
Summe von y-Werten		Σy	
Summe von y^2-Werten		Σy^2	
Stichproben-Standardabweichung von y		Sy	
Grundgesamtheits-Standardabweichung von y †		σy	
Summe von x∗y-Werten		Σxy	
Minimum von x-Werten	minX	minX	
Maximum von x-Werten	maxX	maxX	
Minimum von y-Werten		minY	
Maximum von y-Werten		maxY	
erstes Quartil	q1		
Median	medStat		
drittes Quartil	q3		
Regressionsgleichung			regeq
Regressionskoeffizienten (a, b, c, d, e)			regCoef
Korrelationskoeffizient ††			corr
Bestimmungskoeffizient ††			R^2

	Eine Var	Zwei Var	Regressionen
Summenpunkte (nur Med-Med) †			medx1, medy1, medx2, medy2, medx3, medy3

† Diese Variablen werden zwar berechnet, aber nicht auf dem STAT VARS-Bildschirm angezeigt.

†† **corr** ist nur für eine lineare Regression definiert; $\mathbf{R}^2$ ist für alle polynomischen Regressionen definiert.

Hinweis:

- Wenn **regeq** 4x + 7 ist, dann ist **regCoef** {4 7}. Verwenden Sie einen Index, um auf den "*a*"-Koeffizienten (erstes Element der Liste) zuzugreifen, d.h. **regCoef[1]**.

- Das erste Quartil ist der Median der Punkte zwischen **minX** und **medStat**, und das dritte Quartil ist der Median der Punkte zwischen **medStat** und **maxX**.

Einen Statistik-Plot definieren

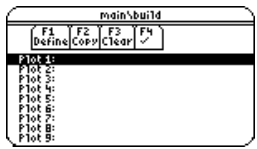
Im Daten/Matrix-Editor können Sie die eingegebenen Daten verwenden, um verschiedenartige Statistik-Plots zu definieren. Sie können bis zu neun Plots gleichzeitig definieren.

Vorgehensweise

Ausgangspunkt ist der Daten/Matrix-Editor:

1. Drücken Sie F2, um den **Plot Setup**-Bildschirm zu öffnen.

   Zu Beginn ist keiner der Plots definiert.

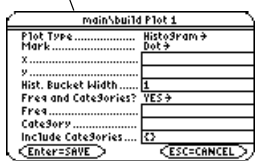


2. Markieren Sie mit dem Cursor den zu definierenden Plot.
3. Drücken Sie F1, um den Plot zu definieren.

   In diesem Beispiel sind alle Menüpunkte aktiv. Auf Ihrem Rechner sind die Menüpunkte nur dann aktiv, wenn sie für die aktuellen Einstellungen von **Plot Type** und **Use Freq and Categories?** gültig sind.

   Hinweis: Ein für die aktuellen Einstellungen ungültiger Menüpunkt wird unscharf angezeigt. Der Cursor kann nicht auf einen unscharf angezeigten Menüpunkt gesetzt werden.

Pfadname der Datenvariablen



4. Nehmen Sie für die aktiven Menüpunkte die geeigneten Einstellungen vor.

Menüpunkt	Beschreibung
Plot Type	Wählen Sie die Plotart.

Menüpunkt	Beschreibung
Mark	Wählen Sie das Symbol zum Plotten der Datenpunkte: Box (□), Cross (x), Plus (+), Square (■), oder Dot (•).
x	Geben Sie die Nummer der im Daten/Matrix-Editor für die x -Werte verwendeten Spalte ein (**C1**, **C2**, etc.) x ist die unabhängige Variable.
y	Geben Sie die Nummer der für y -Werte verwendeten Spalte ein. y ist die abhängige Variable. Nur bei Plot Type = Scatter oder xyline aktiv.
Hist. Bucket Width	Bestimmt die Breite jedes einzelnen Stabs in einem Histogram.
Use Freq and Categories?	Wählen Sie **NO** oder **YES**. Beachten Sie, dass Freq, Category, und Include Categories nur bei der Einstellung Use Freq and Categories? = YES. aktiv sind. (Freq ist nur bei Plot Type = Box Plot oder Histogram aktiv.)
Freq	Geben Sie die Nummer der Spalte ein, die eine Häufigkeit für jeden Datenpunkt enthält. Wenn Sie keine Spaltennummer eingeben, wird für alle Datenpunkte dieselbe Gewichtung angenommen (1).
Category	Geben Sie die Nummer der Spalte ein, die einen Klassenwert für jeden Datenpunkt enthält.

Menüpunkt	Beschreibung
Include Categories	Wenn Sie eine Klassen-Spalte unter Category angeben, können Sie die Berechnung mit Hilfe dieses Menüpunkts auf bestimmte Klassenwerte beschränken. Geben Sie beispielsweise {1,4} an, werden im Plot nur Datenpunkte mit einem Klassenwert von 1 oder 4 verwendet.

Hinweis:

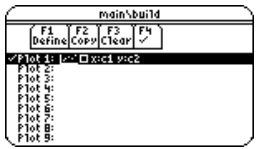
- Bei den mit Spaltennummern definierten Plots wird stets die letzte Datenvariable im Daten/Matrix-Editor verwendet, selbst, wenn diese Variable nicht zur Erstellung der Definition gedient hat.
- Möchten Sie für x, y, Freq oder Category eine vorhandene Listenvariable verwenden, geben Sie anstelle der Spaltennummer den Listennamen ein.
- Ein Beispiel für die Verwendung von Freq, Category und Include Categories ist verfügbar.

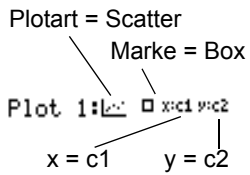
5. Drücken Sie ENTER (nach dem Ausfüllen eines Dialogfeldes drücken Sie ENTER zweimal).

Der **Plot Setup**-Bildschirm wird wieder angezeigt.

Der soeben definierte Plot wird automatisch für die graphische Darstellung gewählt.

Beachten Sie die Kurzschreibweise für diesen Plot.





Hinweis: Bei Statistik-Plots werden undefinierte Datenpunkte (**undef**) ignoriert.

Einen Plot wählen oder die Auswahl aufheben

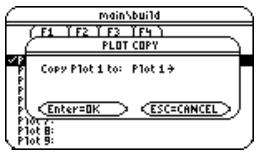
Markieren Sie im Plot Setup den Plot, und drücken Sie F4, um ihn ein- oder auszuschalten. Ein gewählter Statistik-Plot bleibt auch ausgewählt, wenn Sie:

- Den Graphikmodus ändern. (Statistik-Plots werden im 3D-Modus nicht graphisch dargestellt.)
- Einen Graph-Befehl ausführen.
- Eine andere Variable im Daten/Matrix-Editor öffnen.

Eine Plot-Definition kopieren

Ausgangspunkt ist **Plot Setup**:

1. Markieren Sie den Plot, und drücken Sie F2.
2. Drücken Sie ⊙, und wählen Sie die Nummer des Plots, in den kopiert werden soll.
3. Drücken Sie ENTER.



Hinweis: War der Ausgangsplot ausgewählt (✓), dann ist auch dessen Kopie ausgewählt.

Eine Plot-Definition löschen

Markieren Sie im Plot Setup den Plot, und drücken Sie F3 Möchten Sie einen vorhandenen Plot neudefinieren, müssen Sie diesen nicht unbedingt erst löschen; Sie können an der vorhandenen Definition Änderungen vornehmen. Soll ein Plot nicht graphisch dargestellt werden, können Sie die Auswahl aufheben.

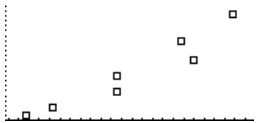
Arten von Statistik-Plots

Bei der Definition eines Plots, wie sie im letzten Abschnitt beschrieben wurde, können Sie im Plot Setup-Bildschirm die Plotart wählen. In diesem Abschnitt werden die verfügbaren Plotarten näher beschrieben.

Scatter

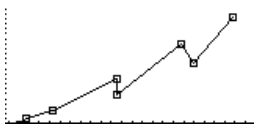
Datenpunkte aus x und y werden als Koordinatenpaare geplottet. Deshalb müssen die Spalten oder Listen, die Sie für x und y angeben, gleich lang sein.

- Geplottete Punkte werden durch das von Ihnen bei Mark gewählte Symbol dargestellt.
- Wenn nötig, können Sie für x und y dieselbe Spalte oder Liste angeben.



Xyline

Die ist ein Streuplot, in dem die Datenpunkte in der Reihenfolge, in welcher sie in x und y auftreten, geplottet und miteinander verbunden sind.



Sie können vor dem Plotten alle Spalten im Daten/Matrix-Editor sortieren.

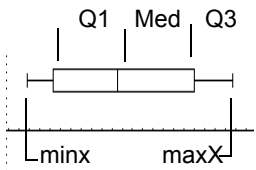
[2nd] [F6] **3** or [2nd] [F6] **4**

[F6] **3** or [F6] **4**

Boxplot

Hier werden aus einer Variablen bestehende Daten unter Berücksichtigung der Minimum- und Maximum-Datenpunkte (**minX** und **maxX**) der Menge geplottet.

- Eine Box wird durch ihr erstes Quartil (**Q1**), den Median (**Med**) und das dritte Quartil (**Q3**) definiert.
- Die Bereiche **minX** bis **Q1** und **Q3** bis **maxX** werden als Linie dargestellt.



- Wenn Sie mehrere Boxplots auswählen, werden sie in der Reihenfolge ihrer Plotnummern übereinander geplottet.
- Um statistische Daten darzustellen, verwenden Sie **NewPlot** als modifizierten Boxplot.
- Beim Definieren eines Plots im Daten/Matrix-Editor wählen Sie Mod Box Plot als Plot-Typ.

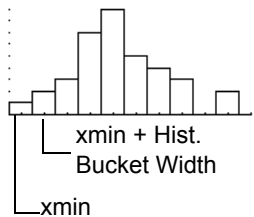
Bei einem veränderten Boxplot werden die Punkte außerhalb des Intervalls [**Q1**−**X, Q3+X**], wobei **1.5 (Q3**−**Q1)** definiert ist, ausgeschlossen. Diese Punkte, auch Ausreißer genannt, werden gesondert außerhalb der Boxplot-Struktur in dem von Ihnen gewählten Stil geplottet.

Histogram

Mit Histogram werden Ein-Variablendaten als Histogramm geplottet. Die x-Achse wird in gleichbreite Balken oder Stäbe unterteilt. Die Höhe eines Stabs (dessen y-Wert) gibt an, wie viele Datenpunkte in den Bereich des jeweiligen Stabs fallen.

- Beim Definieren des Plots können Sie unter **Hist. Bucket Width** die Stabbreite festlegen (Vorgabe ist 1).
- Ein Datenpunkt am Rand eines Stabs wird zu dem Stab zu seiner Rechten gezählt.

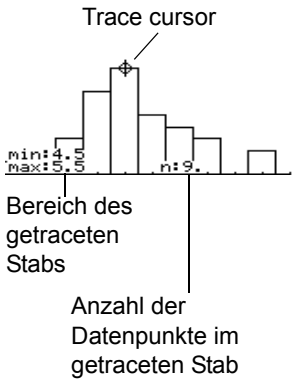
$$\text{Stabanzahl} = \frac{xmax - xmin}{\text{Hist. Bucket Width}}$$



- Mit **ZoomData** (F2 **9** im Graphikbildschirm, Y= Editor, oder Window-Editor) werden zwar **xmin** und **xmax** angepaßt, so daß alle Datenpunkte eingeschlossen sind, die y-Achse wird aber nicht angepaßt.

- Verwenden Sie ◆ [WINDOW], um **ymin auf 0** und **ymax** auf die Anzahl der im höchsten Stab erwarteten Datenpunkte einzustellen.

- Wenn Sie ein Histogramm tracen (F3), werden auf dem Bildschirm Angaben zu dem getraceten Stab angezeigt.



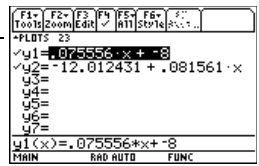
Den Y= Editor für Statistik-Plots verwenden

In den vorigen Abschnitten wurde beschrieben, wie Statistik-Plots im Daten/Matrix-Editor definiert und ausgewählt werden. Sie können Statistik-Plots auch im Y= Editor definieren und auswählen.

Die Liste der Statistik-Plots anzeigen

Drücken Sie ♦ [Y=], um den Y= Editor anzuzeigen. Zu Beginn befinden sich die neun Statistik-Plots “über” dem oberen Bildschirmrand, oberhalb der **y(x)**-Funktionen. Die PLOTS-Anzeige gibt einige Grundinformationen.

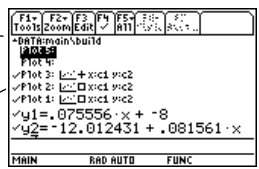
PLOTS 23 heißt z.B., daß die Plots 2 & 3 gewählt sind.



Möchten Sie die Liste der Statistik-Plots einsehen, verwenden Sie ⊙, um über die **y(x)**-Funktionen zu scrollen.

Wenn ein Plot markiert ist, zeigt dies die Datenvariable, die für die Plots verwendet wird.

Wenn ein Plot definiert ist, zeigt dies dieselbe Kurzschreibweise wie der Plot Setup-Bildschirm an.



Im Y= Editor können Sie fast alle Operationen mit Statistik-Plots durchführen, die auch an allen anderen **y(x)**-Funktionen durchgeführt werden können.

Hinweis: Bei den mit Spaltennummern definierten Plots wird stets die letzte Datenvariable im Daten/ Matrix-Editor verwendet, selbst, wenn diese Variable nicht zur Erstellung der Definition gedient hat.

Sie möchten:	Vorgehensweise:
Eine Plotdefinition ändern	Markieren Sie den Plot, und drücken Sie F3. Der gleiche Definitionsbildschirm wie im Daten/Matrix-Editor wird angezeigt.
Einen Plot auswählen oder die Auswahl aufheben	Markieren Sie den Plot, und drücken Sie F4.
Alle Plots und/ oder Funktionen ausschalten	Drücken Sie F5, und wählen Sie den entsprechenden Menüpunkt. Sie können über dieses Menü auch alle Funktionen einschalten.

Hinweis: Mit 2nd F6; F6 kann die Darstellung eines Plots nicht festgelegt werden. In der Plot-Definition kann jedoch die für den Plot zu verwendende Markierung gewählt werden.

Plots und Y=-Funktionen graphisch darstellen

Sie können Statistik-Plots und **y(x)**-Funktionen gleichzeitig auswählen und graphisch darstellen.

Einen definierten Statistik-Plot graphisch darstellen und tracen

Nach Eingabe der Datenpunkte und Definieren der Statistik-Plots können Sie die ausgewählten Plots mit Hilfe des Y= Editors auf gleiche Weise wie Funktionen graphisch darstellen (siehe Grundlagen der graphischen Darstellung von Funktionen).

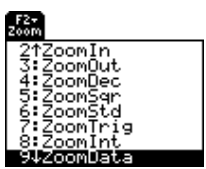
Das Ansichtfenster definieren

Statistik-Plots werden im aktuellen Graphikbildschirm angezeigt, und es werden die im Window-Editor definierten Fenstervariablen verwendet.

Verwenden Sie ♦ [WINDOW], um den Window-Editor einzublenden. Sie können entweder:

- Passende Werte eingeben.
  — oder —
- **9:ZoomData** aus dem Menü F2 **Zoom** wählen. (Sie können zwar jeden Zoom verwenden, **ZoomData** ist aber speziell für Statistik-Plots optimiert).

ZoomData stellt das Ansichtfenster so ein, daß alle statistischen Datenpunkte angezeigt werden.

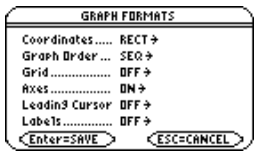


Bei Histogrammen und Boxplots werden nur **xmin** und **xmax** angepaßt. Ist das obere Ende eines Histogramms nicht sichtbar, können Sie den Wert für **ymax** ermitteln, indem Sie das Histogramm tracen.

Hinweis: F2 **Zoom** ist im Y= Editor, im Window-Editor und im Graphikbildschirm verfügbar.

Ändern des Diagrammformats

Drücken Sie folgende Tasten:
F1 **9**
— oder —
◆ |
◆ **F**
im Bildschirm Y= Editor, Window Editor oder Graph.



Passen Sie danach die Einstellungen Ihren Anforderungen an.

Einen Statistik-Plot tracen

Drücken Sie im Graphikbildschirm F3, um einen Plot zu tracen. Die Bewegung des Trace-Cursors hängt von der Art des Plots, Plot Type, ab.

Plotart	Beschreibung
Scatter oder xyline	Das Tracen beginnt am ersten Datenpunkt.
Box plot	Das Tracen beginnt am Median. Drücken Sie ◀, um den Plot nach **Q1** und **minX** zu verfolgen. Drücken Sie ▶, um den Plot nach **Q3** und **maxX** zu verfolgen.

Plotart	Beschreibung
Histogram	Der Cursor bewegt sich von der Mitte der Oberseite jedes Stabs aus und beginnt mit dem ersten Stab links.

Hinweis: Wenn Sie einen angezeigten Statistik-Plot über den rechten oder linken Bildschirmrand hinaus verfolgen, führt der Graphikbildschirm keinen automatischen Schwenk aus. Sie können aber stets ENTER drücken, um den Bildschirm mit dem Trace-Cursor zu zentrieren.

Wenn Sie ▲ oder ▼ drücken, um auf einen anderen Plot oder eine andere **y(x)**-Funktion überzugehen, wird beim Tracen auf den aktuellen oder den Startpunkt dieses Plots übergegangen (nicht auf das nächste Pixel).

Häufigkeiten und Klassen verwenden

Sie können die Analyse von Datenpunkten beeinflussen, indem Sie Häufigkeitswerte und/oder Klassenwerte verwenden. Häufigkeitswerte dienen zum Gewichten bestimmter Datenpunkte. Mit Klassenwerten können Sie eine Untermenge (Klasse) der Datenpunkte analysieren.

Beispiel für eine Häufigkeitsspalte

Bei einer Datenvariablen können Sie zur Angabe eines Häufigkeitswerts (bzw. Gewichtung) für die Datenpunkte jeder Zeile eine beliebige Spalte im Daten/Matrix-Editor verwenden. Ein Häufigkeitswert muß eine ganze Zahl ≥ 0 sein, wenn Calculation Type = OneVar oder MedMed oder wenn Plot Type = Box Plot. Bei anderen statistischen Berechnungen oder Plots kann der Häufigkeitswert eine beliebige Zahl ≥ 0 sein.

Beispiel: Sie geben die Prüfungsergebnisse eines Studenten ein, und es gilt:

- Die Prüfung zur Mitte des Semesters erhält gegenüber anderen Tests das doppelte Gewicht.
- Die Abschlußprüfung zählt dreimal so viel.

Im Daten/Matrix-Editor können Sie die Prüfungsergebnisse und die Häufigkeitswerte in zwei Spalten eingeben.

Prüfungs-ergebnisse	**Häufigkeits-werte**
c1	c2
85	1
97	1
92	2
89	1
91	1
95	3

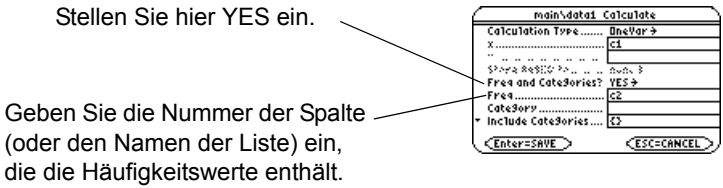
Diese gewichteten Ergebnisse entsprechen der rechts abgebildeten Einzelspalte mit Ergebnissen.

c1
85
97
92 ❶
92 ❶
89
91
95 ❷
95 ❷
95 ❷

❶ Besetzungszahl 2
❷ Besetzungszahl 3

Hinweis: Der Häufigkeitswert 0 schließt den Datenpunkt aus der Analyse aus.

Möchten sie Häufigkeitswerte verwenden, geben Sie bei der Durchführung einer statistischen Berechnung oder beim Definieren eines Statistik-Plots die Häufigkeitsspalte an. Beispiel:



Hinweis: Sie können Häufigkeitswerte auch einer Listenvariablen anstatt einer Spalte entnehmen.

Beispiel für eine Klassenspalte

Bei einer Datenvariablen können Sie zur Angabe einer Klasse (oder eines Untermengenwerts) für die Datenpunkte jeder Zeile eine beliebige Spalte verwenden. Ein Klassenwert kann eine beliebige Zahl sein.

Sie möchten die Testergebnisse einer Menge von Schülern aus der 10. und der 11. Klasse eingeben. Sie möchten sowohl die Ergebnisse der gesamten Schülermenge als auch Klassen, wie Mädchen der 10. Klasse, Jungen der 10. Klasse, Mädchen und Jungen der 10. Klasse usw., analysieren.

Legen Sie zunächst fest, welche Klassenwerte Sie verwenden möchten.

Klassenwert	Steht für:
1	Mädchen 10. Klasse

Klassenwert	Steht für:
2	Junge 10. Klasse
3	Mädchen 11. Klasse
4	Junge 11. Klasse

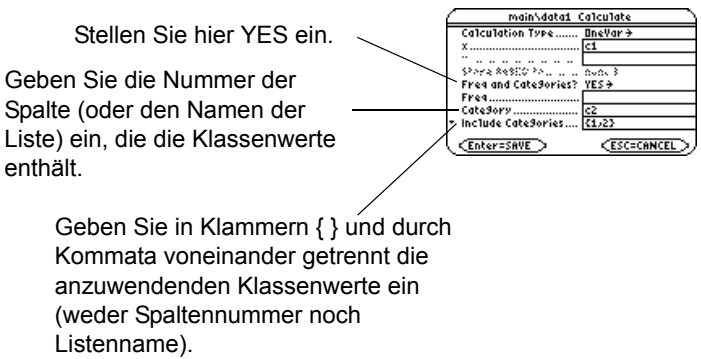
Hinweis: Sie benötigen keinen Klassenwert für die gesamte Schülermenge. Es sind auch keine Klassenwerte für alle Schüler der 10. oder der 11. Klasse erforderlich, da es sich bei ihnen um Kombinationen anderer Klassen handelt.

Im Daten/Matrix-Editor können Sie die Ergebnisse und die Klassenwerte in zwei Spalten eingeben.

Testergebnisse	Klassenwerte
c1	c2
85	1
97	3
92	2
88	3
90	2
95	1
79	4
68	2
92	4
84	3

Testergebnisse	Klassenwerte
c1	c2
82	1

Möchten sie Klassenwerte verwenden, geben Sie bei der Durchführung einer statistischen Berechnung oder bei der Definition eines Statistik-Plots die Klassenspalte und die Klassenwerte an.



Geben Sie in Klammern { } und durch Kommata voneinander getrennt die anzuwendenden Klassenwerte ein (weder Spaltennummer noch Listenname).

Hinweis: Sie können Häufigkeitswerte auch einer Listenvariablen anstatt einer Spalte entnehmen.

Zur Analyse von:	Beziehen Sie diese Klassen ein:
Mädchen 10. Klasse	{1}

Zur Analyse von:	Beziehen Sie diese Klassen ein:
Jungen 10. Klasse	{2}
Mäd. u. Jung. 10. Klasse	{1,2}
Mädchen 11. Klasse	{3}
Jungen 11. Klasse	{4}
Mäd. u. Jung. 11. Klasse	{3,4}
alle Mäd. (10. u. 11. Kl.)	{1,3}
alle Jung. (10. u. 11. Kl.)	{2,4}

Hinweis: Zur Analyse der gesamten Schülermenge füllen Sie das Klassen-Eingabefeld nicht aus. Klassenwerte werden ignoriert.

Wenn Sie über ein CBL 2™ oder CBR™ verfügen

Die Systeme Calculator-Based Laboratory™ (CBL 2) und Calculator-Based Ranger™ (CBR) sind separat erhältliche Zusatzoptionen, die zur Sammlung von Daten aus praktischen Experimenten dienen. Die CBL 2- und CBR-Programme für den TI-89 Titanium / Voyage™ 200: sind auf der TI-Website unter folgender Adresse erhältlich education.ti.com.

So werden CBL 2™-Daten gespeichert

Daten, die Sie mit dem CBL 2/CBL erfassen, werden zunächst im CBL 2 selbst gespeichert. Sie müssen die Daten dann mit dem Befehl **Get**, der im Modul *Technische Referenz* beschrieben wird, laden (d.h. sie an den TI-89 Titanium / Voyage™ 200 übertragen).

Sie können zwar jeden Satz geladener Daten in verschiedenen Variablentypen speichern (list, real, matrix, pic), für die Durchführung von statistischen Berechnungen bieten sich aber Listenvariablen an.

Wenn Sie die gesammelten Daten zum TI-89 Titanium / Voyage™ 200 übertragen, können Sie die zu verwendenden Listenvariablennamen angeben. Mit dem CBL2 ist es beispielsweise möglich, über einen bestimmten Zeitraum Temperaturdaten zu sammeln. Wenn Sie die Daten übertragen, können Sie zum Beispiel:

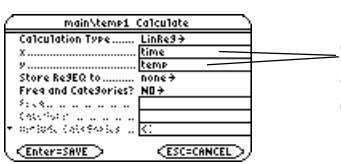
- Temperaturdaten in einer Listenvariablen namens temp speichern.
- Zeitdaten in einer Listenvariablen namens time speichern.

Nachdem Sie die CBL 2-Information im TI-89 Titanium / Voyage™ 200 gespeichert haben, gibt es zwei Möglichkeiten, die CBL 2-Listenvariablen zu verwenden.

Hinweis: Anleitungen zur Bedienung des CBL 2 und zum Übertragen von Daten an den TI-89 Titanium / Voyage™ 200 finden Sie im Handbuch der CBL 2-Einheit.

Bezugnahme auf CBL 2™-Listen

Bei der Durchführung einer statistischen Berechnung oder der Definition eines Plots, können Sie CBL 2-Listenvariablen explizit angeben. Beispiel:

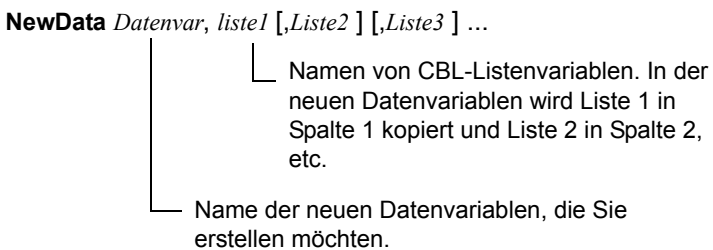


Geben Sie anstelle einer Spaltennummer den Namen der CBL2-/CBL-Listenvariablen ein.

Eine Datenvariable mit CBL 2™-Listen erzeugen

Sie können eine neue Datenvariable erstellen, die aus CBL 2-Listenvariablen besteht.

- Verwenden Sie im Hauptbildschirm oder von einem Programm aus den Befehl **NewData**.



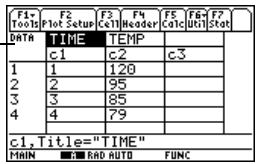
Beispiel:

NewData *temp1*, *time*, *temp*

erstellt eine Datenvariable namens temp1, in welcher die Zeit in Spalte 1 und die Temperatur in Spalte 2 dargestellt ist.

- Erstellen Sie im Daten/Matrix-Editor eine neue, leere Datenvariable mit einem beliebigen Namen. Definieren sie für jede CBL 2-Liste, die Sie mit einschließen möchten, eine Kopfzeile als den Namen dieser Liste.

Definieren Sie z.B. Spalte 1 als Time, Spalte 2 als Temp.



Hinweis: Zum Definieren oder Löschen einer Kopfzeile verwenden Sie F4. Näheres finden Sie im Modul *Daten/Matrix Editor*.

Nun sind die Spalten mit den CBL-Listen verbunden. Werden die Listen verändert, erfolgt eine automatische Aktualisierung der Spalten. Wenn die Listen aber gelöscht werden, gehen die Daten verloren.

Sollen die Datenvariablen unabhängig von den CBL 2-Listen vorliegen, können Sie den Kopf jeder Spalte löschen. Die Daten in der Spalte bleiben erhalten, aber es besteht keine Verbindung mehr zwischen Spalte und CBL 2-Liste.

CBR™

Außerdem können Sie mit dem Calculator-Based Ranger™ (CBR) die mathematischen und wissenschaftlichen Zusammenhänge zwischen Entfernung, Geschwindigkeit, Beschleunigung und Zeit für die bei Ihrer Untersuchung gesammelten Daten ermitteln.

Programmieren

Ein vorhandenes Programm ausführen

Nachdem Sie ein Programm erstellt haben (wie in den verbleibenden Abschnitten dieses Einheits erläutert), können Sie es vom Hauptbildschirm aus ausführen. Die vom Programm erzeugte Ausgabe wird auf dem Programm-I/O-Bildschirm, in einem Dialogfeld oder auf dem Graphikbildschirm angezeigt.

Ein Programm ausführen

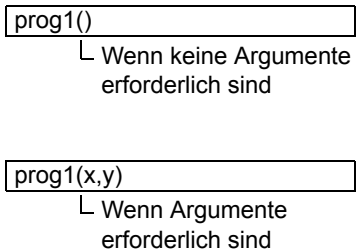
Ausgangspunkt ist der Hauptbildschirm:

1. Geben Sie den Namen des Programms ein.

2. Sie müssen immer ein Paar runder Klammern hinter dem Programmnamen eingeben.

   Für manche Programme ist die Übergabe eines Arguments notwendig.

   Hinweis: Mit [2nd] [VAR-LINK] können Sie sich die vorhandenen **PRGM**-Variablen anzeigen lassen. Markieren Sie eine Variable, und drücken Sie [ENTER], um ihren Namen in die Eingabezeile zu kopieren.

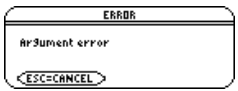


3. Drücken Sie [ENTER].

Hinweis: Die Argumente sind Ausgangswerte für ein Programm.

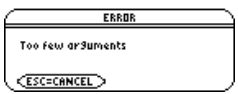
Wenn Sie ein Programm ausführen, nimmt der TI-89 Titanium / Voyage™ 200 Graphischer Rechner automatisch eine Überprüfung auf Fehler vor. Beispielsweise wird folgende Fehlermeldung angezeigt, wenn Sie:

- Kein Klammernpaar ( ) hinter dem Programmnamen eingeben.



Diese Fehlermeldung wird angezeigt, wenn Sie:

- Zu wenige Argumente eingeben.



Bei Auftreten eines Fehlers können Sie die Programmausführung abbrechen, indem Sie [ESC] drücken. Sie können dann die Fehler beheben und das Programm erneut starten.

Hinweis: Der TI-89 Titanium / Voyage™ 200 überprüft das Programm auch auf Laufzeitfehler.

Einen Programmstopp erzwingen

Wenn die Programmausführung läuft, wird in der Statuszeile `BUSY` angezeigt.

Drücken Sie [ON], um einen Programmstopp zu erzwingen. Eine entsprechende Meldung wird angezeigt.

- Möchten Sie nun das Programm im Programmeditor anzeigen lassen, drücken Sie [ENTER]. Der Cursor steht bei dem Befehl, an dem der Programmstopp erfolgte.
- Möchten Sie die Programmausführung abbrechen, drücken Sie [ESC].

Wo wird die Programmausgabe angezeigt?

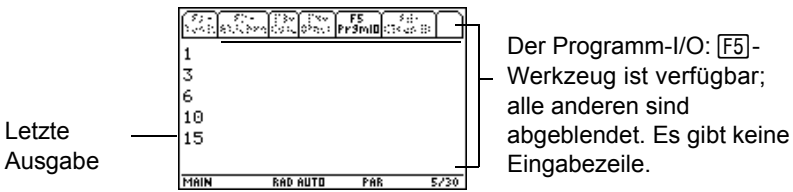
Je nach den Befehlen im Programm, leitet der TI-89 Titanium / Voyage™ 200 die Ausgabe automatisch auf den entsprechenden Bildschirm.

- Die meisten Ein-/Ausgabebefehle benutzen den Programm-I/O-Bildschirm. (Eingabebefehle sind Befehle, die den Benutzer auffordern, eine bestimmte Eingabe vorzunehmen).
- Befehle, die sich auf Graphen beziehen, benutzen in der Regel den Graphikbildschirm.

Nach Programmende zeigt der TI-89 Titanium / Voyage™ 200 den letzten zuvor angezeigten Bildschirm an.

Der Programm-I/O-Bildschirm

Im Programm-I/O-Bildschirm erfolgt die neue Ausgabe unterhalb der vorherigen (die vom selben Programm oder von einem anderen stammen kann). Ist die Bildschirmseite gefüllt, wird der Bildschirminhalt bei der nächsten Ausgabe nach oben gescrollt.



Hinweis: Nehmen Sie den Befehl **ClrIO** in Ihr Programm auf, um Ausgaben zu löschen, die von einem vorher ausgeführten Programm stammen. Sie können **ClrIO** auch vom Hauptbildschirm ausführen.

Wenn bei Programmende der Programm-I/O-Bildschirm angezeigt wird, kann man irrtümlich annehmen, es handele sich um den Hauptbildschirm, da sich beide Anzeigen ähneln. Der Programm-I/O-Bildschirm wird jedoch nur für die Programmausgabe bzw. für Eingabeaufforderungen benutzt. Sie können in diesem Bildschirm keine normalen Berechnungen ausführen.

Hinweis: Wenn nach Ausführen eines Programms keine Hauptbildschirm-Rechenvorgänge möglich sind, befinden Sie sich wahrscheinlich noch im Programm-I/O.

Den Programm-I/O-Bildschirm verlassen

Ausgangspunkt ist der Programm-I/O-Bildschirm:

- Drücken Sie F5 wechseln Sie vom Hauptbildschirm zum Programm-I/O-Bildschirm und zurück.
  – oder –
- Drücken Sie ESC, 2nd [QUIT], oder
  HOME
  ◆ [CALC HOME]
  um den Hauptbildschirm anzuzeigen.

  – oder –
- Wechseln Sie zu einem anderen Anwendungsbildschirm (mit APPS, ◆ [Y=] etc.).

Eine Programmeditor-Sitzung starten

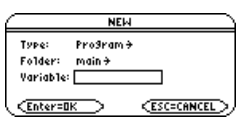
Beim Aufruf des Programmeditors können Sie das aktuelle Programm bzw. die aktuelle Funktion fortsetzen (d. h. die zuletzt angezeigte Programmeditor-Sitzung), ein vorhandenes Programm bzw. eine vorhandene Funktion öffnen oder ein neues Programm bzw. eine neue Funktion beginnen.

Ein neues Programm oder eine neue Funktion beginnen

1. Drücken Sie APPS, und wählen Sie dann **Program Editor**.

2. Wählen Sie **3:New**.

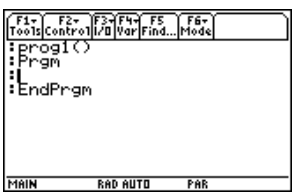
3. Machen Sie die erforderlichen Angaben für das neue Programm bzw. die neue Funktion.



Feld	Beschreibung:
Type	Wählen Sie, ob Sie ein neues Programm oder eine neue Funktion erstellen möchten.
Folder	Wählen Sie das Verzeichnis, in dem das Programm bzw. die Funktion gespeichert werden sollen. Nähere Erläuterungen zu Verzeichnissen finden Sie in *Startbildschirm-Modul des Rechners*.
Variable	Geben Sie einen Variablennamen für das Programm/die Funktion ein. Wenn Sie eine bereits existierende Variable angeben, erhalten Sie nach Drücken von ENTER eine Fehlermeldung. Wenn Sie die Fehlermeldung durch Drücken von ESC oder ENTER bestätigen, wird erneut das Dialogfeld **NEW** angezeigt.

4. Drücken Sie ENTER, um eine leere “Schablone” aufzurufen (zum Abschluß der Eingabe in ein Eingabefeld, z. B. **Variable**, müssen Sie ENTER zweimal drücken).

Dies ist eine Programmvorlage. Die Funktionsvorlage sieht ähnlich aus.



Sie können nun, wie in den folgenden Abschnitten erläutert, mit dem Programmeditor arbeiten.

Hinweis: Das Programm/die Funktion wird bei der Eingabe automatisch gespeichert. Deshalb brauchen Sie das Programm/die Funktion nicht manuell zu speichern,

bevor Sie den Programmeditor verlassen, ein neues Programm beginnen oder ein vorhandenes öffnen.

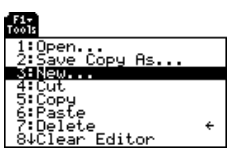
Das aktuelle Programm fortsetzen

Sie können den Programmeditor jederzeit verlassen und zu einer anderen Anwendung wechseln. Um zu dem vor Verlassen des Programmeditors angezeigten Programm bzw. der angezeigten Funktion zurückzukehren, drücken Sie APPS, und wählen Sie **1:Current**.

Ein neues Programm im Programmeditor beginnen

Um das aktuelle Programm bzw. die aktuelle Funktion zu verlassen und ein neues Programm/eine neue Funktion zu beginnen:

1. Drücken Sie F1, und wählen Sie **3:New**.
2. Geben Sie den Typ, ein Verzeichnis und eine Variable für das neue Programm/die neue Funktion an.
3. Drücken Sie zweimal ENTER.



Ein früheres Programm öffnen

Sie können jederzeit ein früher erstelltes Programm/eine früher erstellte Funktion öffnen.

1. Innerhalb des **Program Editor**: Drücken Sie F1, und wählen Sie **1:Open**.

   – oder –

   Aus einer anderen Anwendung heraus: Drücken Sie APPS, und wählen Sie **2:Open**.

2. Wählen Sie den gewünschten Typ, das Verzeichnis und die Variable.
3. Drücken Sie ENTER.

Hinweis: Standardmäßig zeigt Variable das erste Programm/die erste Funktion in der alphabetischen Reihenfolge an.

Ein Programm kopieren

Sie können ein Programm/eine Funktion kopieren, die Kopie ändern und so das Original unverändert beibehalten.

1. Lassen Sie sich das zu kopierende Programm/die Funktion anzeigen.
2. Drücken Sie F1, und wählen Sie **2:Save Copy As**.
3. Geben Sie ein Verzeichnis und eine Variable für die Kopie an.
4. Drücken Sie zweimal ENTER.

Hinweis zum Löschen eines Programms

Da alle Programmeditor-Sitzungen automatisch gespeichert werden, können sich relativ viele Programme ansammeln und es wird u.U. viel Speicherplatz belegt.

Sie löschen ein, Programm/eine Funktion mit Hilfe des Bildschirms VAR-LINK ([2nd] [VAR-LINK]). Nähere Erläuterungen zu VAR-LINK finden Sie in *Speicherund Variablen-Verwaltung*.

Überblick über die Programmeingabe

Ein Programm ist eine Folge von Befehlen, die nacheinander abgearbeitet werden (wobei manche den Programmablauf ändern können). Generell gilt, dass alles, was auf dem Hauptbildschirm ausgeführt werden kann, auch in ein Programm aufgenommen werden kann. Die Ausführung eines Programms endet am Programmende oder wenn es einen **Stop** Befehl erreicht.

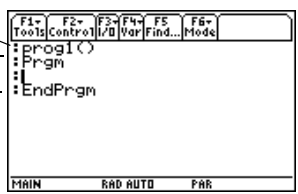
Programmzeilen eingeben und bearbeiten

Beginnen Sie in einer leeren Programmschablone mit der Eingabe der Programmzeilen.

Programmname, den Sie festlegen, wenn Sie ein neues beginnen.

Die Programmbefehle geben Sie zwischen **Prgm** und **EndPrgm** ein.

Jede Programmzeile beginnt mit einem Doppelpunkt.



Hinweis: Verwenden Sie das Cursorfeld, um den Cursor an die gewünschte Position (für Eingabe oder Ändern) zu bringen. Gehen Sie mit [♦] ⊖ oder [♦] ⊕ zum Anfang bzw. Ende eines Programms.

Die Eingabe und die Bearbeitung der Programmbefehle im Programmeditor nehmen Sie auf gleiche Weise vor wie das Eingeben und Bearbeiten von Text im Texteditor. Siehe “Text eingeben und bearbeiten” in *Texteditor*.

Drücken Sie zum Abschluß jeder Programmzeile [ENTER]. Dadurch wird eine neue leere Zeile eingefügt, in der Sie die Eingabe fortsetzen können. Eine Programmzeile kann länger sein als eine Bildschirmzeile. Ist dies der Fall, wird sie automatisch in der nächsten Bildschirmzeile fortgesetzt.

Hinweis: Ein Befehl wird nicht beim Eingeben ausgeführt. Er wird erst ausgeführt, wenn Sie das Programm ausführen.

Mehrere Befehle in eine einzige Zeile eingeben

Sie können mehrere Befehle in eine einzige Zeile eingeben, sofern Sie sie jeweils durch einen Doppelpunkt trennen (drücken Sie [2nd] [:]).

Kommentare eingeben

Mit Hilfe des Kommentarsymbols (◔) können Sie Anmerkungen in ein Programm aufnehmen. Bei der Programmausführung werden alle Zeichen rechts des ◔ gnoriert.

```
  :prog1()
  :Prgm
❶ :◔Displays sum of 1 thru n
  :Request "Enter an integer",n
❷ :expr(n)→n:◔Convert to numeric expression
  :------
```

❶ Beschreibung des Programms.
❷ Beschreibung von **expr**.

Hinweis: Verwenden Sie Kommentare, um erläuterende Informationen zum Programmcode aufzuzeichnen.

So geben Sie ein Kommentarsymbol ein:

- Drücken Sie

  – oder –
- Drücken Sie F2, und wählen Sie **9**:◉.

Den Programmablauf steuern

Wenn Sie ein Programm ausführen, werden die Befehle grundsätzlich sequentiell abgearbeitet. Es stehen jedoch auch Befehle zur Verfügung, die den Programmablauf verändern. Zum Beispiel:

- Mit Strukturen wie **If...EndIf** wird eine Bedingung überprüft und anhand des Ergebnisses der Überprüfung entschieden, welcher Teil des Programms ausgeführt wird.
- Schleifen wie **For...EndFor** dienen zum Wiederholen einer Befehlsgruppe.

Einrückungen benutzen

Komplexere Programme mit **If...EndIf** Strukturen und **For...EndFor** Schleifen können Sie leichter lesbar und verständlicher gestalten, indem Sie Einrückungen verwenden.

```
:If x>5 Then
:  Disp "x is > 5"
:Else
:  Disp "x is < or = 5"
:EndIf
```

Ergebnisse von Berechnungen anzeigen

In einem Programm werden Ergebnisse von Berechnungen nur angezeigt, wenn Sie einen Ausgabebefehl benutzen. Dies ist ein wichtiger Unterschied zu Berechnungen, die Sie auf dem Hauptbildschirm vornehmen.

Das Ergebnis der nebenstehenden Berechnungen wird in einem Programm nicht angezeigt (da kein Ausgabebefehl vorhanden ist). Auf dem Hauptbildschirm würden die Ergebnisse jedoch angezeigt.

```
:12*6
:cos(π/4)
:solve(x^2-x-2=0,x)
```

Mit Ausgabebefehlen wie **Disp** zeigen Sie Ergebnisse in einem Programm an.

```
:Disp 12*6
:Disp cos(π/4)
:Disp solve(x^2-x-
2=0,x)
```

Das Anzeigen eines Berechnungsergebnisses bewirkt jedoch keine Speicherung des Ergebnisses. Um auf ein Ergebnis später erneut zuzugreifen, legen Sie es in einer Variablen ab.

```
:cos(π/4)→maximum
:Disp maximum
```

Hinweis: Eine Zusammenstellung der verfügbaren Ausgabebefehle ist verfügbar.

Werte an ein Programm übergeben

Sie können Werte wie folgt an ein Programm übergeben:

- Indem Sie vor Ausführen des Programms die benötigten Werte in Variablen speichern (mit STO▸). Das Programm kann dann auf diese Variablen zugreifen.

- Indem Sie Werte direkt in den Programmtext aufnehmen.

```
:Disp 12*6
:cos(π/4)→maximum
```

- Indem Sie Eingabebefehle benutzen, die den Benutzer bei Ausführung des Programms zur Eingabe der benötigten Werte auffordern.

```
:Input "Enter a
value",i
:Request "Enter an
integer",n
```

- Indem Sie einen oder mehrere Werte bei Programmstart an das Programm übergeben.

```
prog1(3,5)
```

Hinweis: Eine Zusammenstellung der verfügbaren Eingabebefehle ist verfügbar.

Beispiel für die Übergabe von Werten an ein Programm

Das folgende Programm zeichnet einen Kreis im Graphikbildschirm und anschließend eine horizontale Gerade entlang des obersten Punkts des Kreises. Drei Werte müssen an das Programm übergeben werden: die x- und die y-Koordinate das Kreismittelpunkts sowie der Radius r.

- Wenn sie das Programm im Programmeditor verfassen:

  Geben Sie im Klammerpaar ( ) neben dem Programmnamen die Variablen an, die zur Aufnahme der übergebenen Werte vorgesehen sind.

  Beachten Sie, dass das Programm auch Befehle für das Einrichten des Graphikbildschirms enthält.

```
:circ(x,y,r) ❶
.Prgm
:FnOff
:ZoomStd
:ZoomSqr
:Circle x,y,r
:LineHorz y+r
:EndPrgm
```

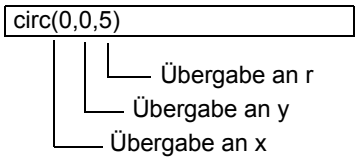
❶ Nur **circ( )** wird in der leeren Schablone angezeigt. Die Parameternamen müssen Sie eintragen.

Hinweis: Sie können hier nicht circle als Programmnamen benutzen, da dies zu einem Konflikt mit dem gleichnamigen Befehlsnamen führt.

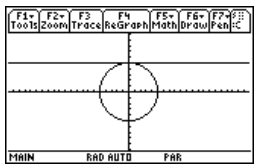
Vor dem Zeichnen des Kreises deaktiviert das Programm alle ausgewählten Y= Editor-Funktionen, zeigt ein Standard-Ansichtfenster an und richtet dies verhältnisgerecht ein.

- Um das Programm vom Hauptbildschirm auszuführen:

  Muß der Benutzer die entsprechenden Werte im Klammerpaar ( ) angeben.



  Die Argumente werden in dieser Reihenfolge an das Programm übergeben.



Hinweis: Dieses Beispiel geht davon aus, dass der Benutzer Werte eingibt, die im mit **ZoomStd** und **ZoomSqr** eingerichteten Fenster angezeigt werden können.

Überblick über die Funktionseingabe

Eine Funktion, die Sie im Programmeditor eingeben, ähnelt sehr stark den Funktionen und Anweisungen, die Sie normalerweise auf dem Hauptbildschirm benutzen.

Wozu dienen benutzerdefinierte Funktionen?

Funktionen (wie auch Programme) sind ideal für wiederkehrende Berechnungen oder Aufgaben. Sie brauchen eine Funktion nur einmal zu verfassen. Danach können Sie sie so oft wie nötig wiederverwenden. Funktionen besitzen jedoch einige Vorzüge gegenüber Programmen.

- Sie können Funktionen erstellen, die die integrierten Funktionen des TI-89 Titanium / Voyage™ 200 Graphischer Rechner erweitern. Die neuen Funktionen können Sie dann auf gleiche Weise wie andere benutzen.
- Funktionen geben Werte zurück, die graphisch dargestellt oder in eine Tabelle übernommen werden können. Programme können dies nicht.
- Sie können eine Funktion innerhalb eines Terms benutzen (nicht jedoch ein Programm). Zum Beispiel: **3∗func1(3)** ist gültig, aber nicht **3∗prog1(3)**.
- Da Sie einer Funktion Argumente übergeben können, können Sie allgemeingültige Funktionen verfassen, die nicht an spezifische Variablennamen gebunden sind.

Hinweis: Sie können eine Funktion auch im Hauptbildschirm erstellen, aber der Programmeditor ist für komplexe, mehrzeilige Funktionen besser geeignet.

Unterschiede zwischen Funktionen und Programmen

In diesem Handbuch wird der Begriff Befehl vielfach übergreifend für Anweisungen und Funktionen benutzt. Beim Verfassen einer Funktion müssen Sie jedoch zwischen Anweisungen und Funktionen unterscheiden.

Eine benutzerdefinierte Funktion:

- Kann nur die folgenden Anweisungen benutzen. Alle anderen sind nicht zulässig.

Cycle	**Define**	**Exit**
For...EndFor	**Goto**	**If...EndIf** (alle Formen)
Lbl	**Local**	**Loop...EndLoop**
Return	**While...EndWhile**	→ (STO▸ Taste)

- Kann alle integrierten Funktionen des TI-89 Titanium / Voyage™ 200 außer den folgenden benutzen:

  setFold **setGraph** **setMode**
  setTable **switch**

- Kann jede Variable nutzen, jedoch einen Wert nur in eine lokale Variable ablegen.
  - Die Argumente, mit denen Werte an eine Funktion übergeben werden, werden automatisch als lokale Variablen behandelt. Wenn Sie Werte in weitere Variablen ablegen möchten, müssen Sie sie innerhalb der Funktion als lokale Variablen deklarieren.
- Kann kein Programm als Unterroutine aufrufen, kann aber eine andere benutzerdefinierte Funktion aufrufen.
- Kann kein Programm definieren.
- Kann keine globale Funktion definieren, kann jedoch eine lokale Funktion definieren.

Hinweis: Nähere Erläuterungen zu lokalen Variablen sind verfügbar.

Eine Funktion eingeben

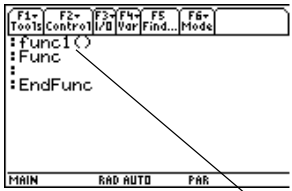
Wenn sie im Programmeditor eine neue Funktion erstellen, zeigt der TI-89 Titanium / Voyage™ 200 eine leere “Schablone” an.

Hinweis: Verwenden Sie das Cursorfeld, um den Cursor an die gewünschte Position (für Eingabe oder Ändern) zu bringen.

Der Funktionsname, den Sie beim Beginnen einer neuen Funktion festlegen.

Die Befehle geben Sie zwischen **Func** und **EndFunc** ein.

Alle Zeilen einer Funktion beginnen mit einem Doppelpunkt.



Vergessen Sie nicht, hier die ggf. erforderlichen Argumente anzugeben. Verwenden Sie in der Definition nur Argumentnamen, die nie zum Aufrufen der Funktion benutzt werden.

Ist für die Funktion eine Eingabe erforderlich, müssen Werte an die Funktion übergeben werden. (Eine benutzerdefinierte Funktion kann nur lokale Variablen speichern und kann keine Anweisungen verwenden, die den Benutzer zu einer Eingabe auffordern).

Einen Wert aus einer Funktion zurückgeben

Es gibt zwei Möglichkeiten, einen Wert aus einer Funktion zurückzugeben:

- Lassen Sie den Rückgabewert in der letzten Funktionszeile (vor **EndFunc**), berechnen

```
:cube(x)
:Func
:x^3
:EndFunc
```

- Verwenden Sie **Return**. Der Nutzen dieses Befehls liegt darin, daß Sie eine Funktion an einem beliebigen Punkt vor dem Funktionsende verlassen und einen Rückgabewert übergeben können.

```
:cube(x)
:Func
:If x<0
:  Return 0
:x^3
:EndFunc
```

Hinweis: In diesem Beispiel erfolgt die Berechnung nur, wenn **x**≥**0**; andernfalls wird `0` zurückgegeben.

Das Argument x wird automatisch als lokale Variable behandelt. Wäre hier jedoch eine andere Variable erforderlich, müsste diese mit dem Befehl Local als lokale Variable deklariert werden.

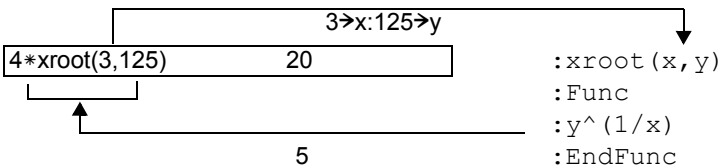
Am Ende einer Funktion ist ein implizites **Return** vorhanden. Wenn die letzte Zeile kein Term ist, tritt ein Fehler auf.

Eine Beispielfunktion

Die folgende Funktion liefert die x-te Wurzel von y ($\sqrt[x]{y}$). Zwei Werte müssen an die Funktion übergeben werden: x und y.

Hinweis: Da x und y in der Funktion lokal sind, besitzen eventuell bereits existierende Variable x oder y keine Auswirkungen auf sie.

Funktion bei Aufruf vom Hauptbildschirm	Definition der Funktion im Programmeditor

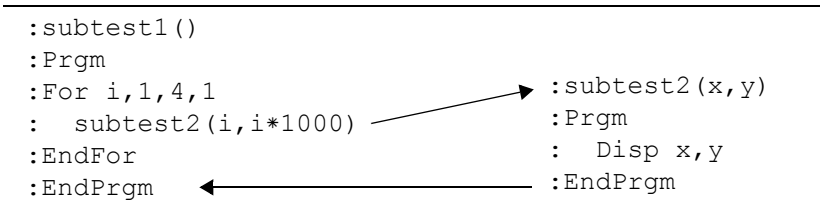


Ein Programm aus einem anderen heraus aufrufen

Ein Programm kann ein anderes Programm als Unterprogramm aufrufen. Das Unterprogramm kann extern (ein anderes Programm) oder intern (Teil des Hauptprogramms) sein. Unterprogramme sind sinnvoll, wenn eine Gruppe von Befehlen an mehreren Stellen im Programm wiederholt werden muß.

Ein anderes Programm aufrufen

Zum Aufrufen eines anderen Programms benutzen Sie die gleiche Syntax wie für das Ausführen eines Programms auf dem Hauptbildschirm.



Ein internes Unterprogramm aufrufen

Ein internes Unterprogramm definieren Sie mit dem Befehl **Define** in Verbindung mit **Prgm...EndPrgm**. Da ein Unterprogramm definiert werden muß, bevor es zum ersten Mal aufgerufen werden kann, sollten Unterprogramme am Anfang des Hauptprogramms definiert werden.

Ein internes Unterprogramm wird auf gleiche Weise aufgerufen und ausgeführt wie ein eigenständiges Programm.

```
  :subtest1()
  :Prgm
❶ :local subtest2
❷ :Define subtest2(x,y)=Prgm
⋮ :  Disp x,y
❷ :EndPrgm
  :Beginning of main program
  :For i,1,4,1
❸ :  subtest2(i,I*1000)
  :EndFor
  :EndPrgm
```

❶ Deklariert das Unterprogramm als lokale Variable.
❷ Definiert das Unterprogramm.
❸ Ruft das Unterprogramm auf.

Hinweis: Benutzen Sie das Menü F4 **Var** zur Eingabe der Befehle **Define** und **Prgm...EndPrgm**.

Hinweise zur Benutzung von Unterprogrammen

Am Ende eines Unterprogramms wird die Ausführung an das aufrufende Programm zurückgegeben. Um ein Unterprogramm zu einem beliebigen Zeitpunkt vorher zu verlassen, können Sie den Befehl **Return** benutzen.

Ein Unterprogramm kann nicht auf lokale Variablen zugreifen, die im aufrufenden Programm deklariert sind. Gleichermaßen kann das aufrufende Programm nicht auf lokale Variablen zugreifen, die in einem Unterprogramm deklariert sind.

Der Befehl **Lbl** ist lokal für das Programm, in dem er sich befindet. Daher kann mit dem Befehl **Goto** im aufrufenden Programm nicht zu einem Label in einem Unterprogramm (und umgekehrt) verzweigt werden.

Variablen in einem Programm benutzen

Programme verwenden Variablen grundsätzlich auf gleiche Weise wie Sie dies auf dem Hauptbildschirm tun. Allerdings bestimmt der “Geltungsbereich” einer Variablen, wie sie gespeichert und wie auf sie zugegriffen wird.

Geltungsbereich von Variablen

Geltungsbereich	Beschreibung
Systemvariablen (globale Variablen)	Variablen mit reservierten Namen, die automatisch erzeugt werden, um Daten über den Status des TI-89 Titanium / Voyage™ 200 Graphischer Rechner zu speichern. So sind beispielsweise die Window-Variablen (**xmin**, **xmax**, **ymin**, **ymax** etc.) global von jedem Verzeichnis aus verfügbar. • Sie können stets unter bloßer Angabe des Variablennamens auf diese Variablen zugreifen, unabhängig davon, welches das aktuelle Verzeichnis ist. • Ein Programm kann keine Systemvariablen erzeugen. Es kann deren Werte jedoch benutzen und (in den meisten Fällen) neue Werte darin ablegen.

Geltungsbereich	Beschreibung
Verzeichnis-variablen	Variablen, die in einem bestimmten Verzeichnis gespeichert sind. • Wenn Sie zum Speichern eines Werts nur den Variablennamen angeben, wird er im aktuellen Verzeichnis abgelegt. Zum Beispiel: 5→start • Wenn Sie nur einen Variablennamen angeben, muß die Variable sich im aktuellen Verzeichnis befinden. Andernfalls kann sie nicht gefunden werden (selbst, wenn sie in einem anderen Verzeichnis vorhanden ist). • Zum Speichern in einer Variablen, die sich in einem anderen Verzeichnis befindet, bzw. zum Abrufen einer solchen Variablen müssen Sie den Pfad angeben. Zum Beispiel: 5→class\start (class = Verzeichnisname; start = Variablenname) Hat ein Programm Verzeichnisvariablen erzeugt, bleiben diese auch nach Programmende erhalten, d. h. sie belegen weiterhin Speicherplatz.

Geltungsbereich	Beschreibung
Lokale Variablen	Temporäre Variablen, die nur während der Ausführung eines Programms existieren. Bei Programmbeendigung werden lokale Variablen automatisch gelöscht. • Sie erzeugen eine lokale Variable in einem Programm, indem Sie sie mit dem Befehl **Local** deklarieren. • Eine lokale Variable ist auch dann eindeutig, wenn eine Verzeichnisvariable gleichen Namens existiert. • Lokale Variablen sind ideal, um Werte temporär zu speichern, die Sie nicht dauerhaft ablegen möchten.

Hinweis: Eine Graph-Funktion kann nicht auf lokale Variablen eines Programms zugreifen. Zum Beispiel:

```
Local a
5→a
Graph a*cos(x)
```

Dies kann zur Anzeige eines Fehlers oder zu einem unerwarteten Resultat führen (wenn die Variable a bereits im aktuellen Verzeichnis vorhanden ist).

"Circular Definition"-Fehler

Bei der Auswertung einer benutzerdefinierten Funktion oder der Ausführung eines Programms können Sie ein Argument angeben, das dieselbe Variable enthält, welche bereits zum Definieren der Funktion oder Erstellen des Programms verwendet wurde. Die oben beschriebenen Änderungen im Umgang mit benutzerdefinierten Funktionen

und Programmen schließen das Auftreten eines Circular definition-Fehlers jedoch nicht gänzlich aus. Beispiel:

```
❶ x+1→x

  – oder –

  For i,i,10,1
❶ Disp i
  EndFor
```

❶ Führt zu einem `Circular definition` -Fehler, wenn x oder i keinen Wert besitzen. Der Fehler tritt hingegen nicht auf, wenn x oder i bereits ein Wert zugeteilt wurde.

Befehle für Variablen

Befehl	Beschreibung
STO▸ Taste	Speichert einen Wert in eine Variable ab. Wie auf dem Hauptbildschirm bewirkt das Drücken von STO▸ das Symbol →.
Archive	Verschiebt die angegebenen Variablen aus dem RAM-Speicher in den Benutzerarchiv-Speicher.
BldData	Dient zum Erstellen einer Datenvariablen auf Grundlage der in den Y=Editor, den Window-Editor etc. eingegebenen Graph-Daten.
CopyVar	Kopiert den Inhalt einer Variablen.
Define	Definiert eine Programm-/Unterroutinenvariable oder eine Funktionsvariable in einem Programm.

Befehl	Beschreibung
DelFold	Löscht ein Verzeichnis. Zuvor müssen sämtliche Variablen in diesem Verzeichnis gelöscht werden.
DelVar	Löscht eine Variable.
getFold	Gibt den Namen des aktuellen Verzeichnisse zurück.
getType	Gibt eine Zeichenkette zurück, die den Datentyp einer Variablen anzeigt. (EXPR, LIST, etc.)
Local	Deklariert eine oder mehrere Variablen als lokale Variable.
Lock	Sperrt eine Variable, so dass sie nicht versehentlich geändert oder gelöscht werden kann. Erst nachdem die Sperre aufgehoben ist, kann sie geändert bzw. gelöscht werden.
MoveVar	Verschiebt eine Variable aus einem Verzeichnis in ein anderes.
NewData	Erzeugt eine Datenvariable, deren Spalten aus einer Folge angegebener Listen bestehen.
NewFold	Erzeugt ein neues Verzeichnis.
NewPic	Erzeugt eine Bild-Variable auf der Basis einer Matrix.
Rename	Benennt eine Variable um.
Unarchiv	Verschiebt bestimmte Variablen aus dem Speicher des Benutzerdatenarchivs in den RAM-Speicher.
Unlock	Hebt die Sperre einer Variablen auf.

Hinweis: Die Befehle **Define**, **DelVar** und **Local** finden Sie im Menü F4 **Var** des Programmeditors.

Lokale Variablen in Programmen verwenden

Eine lokale Variable ist eine temporäre Variable, die nur während der Auswertung einer benutzerdefinierten Funktion oder der Ausführung eines benutzerdefinierten Programms existiert.

Beispiel für lokalen Variablen

Der folgende Programmausschnitt zeigt eine **For...EndFor** Schleife (Einzelheiten dazu folgen weiter hinten in dieser Einheit). Die Variable i ist der Schleifenzähler. In den meisten Fällen wird diese Variable nur für die Dauer der Programmausführung benutzt.

```
❶ :Local I
  :For i,0,5,1
  :  Disp I
  :EndFor
  :Disp i
```

❶ Deklariert die Variable i als “lokal”.

Hinweis: Benutzen Sie nach Möglichkeit den Typ “lokale” Variable für alle Variablen, die nur innerhalb eines Programms benötigt werden und nach Programmende nicht gespeichert bleiben müssen.

Wenn Sie die Variable i als lokal deklarieren, wird sie bei Programmende automatisch gelöscht und belegt nicht unnötig Speicherplatz.

Wodurch wird der Undefined Variable-Fehler verursacht?

Die Fehlermeldung **Undefined** variable wird angezeigt, wenn Sie eine benutzerdefinierte Funktion auswerten oder ein benutzerdefiniertes Programm ausführen, das sich auf nicht initialisierte lokale Variablen (Variablen, welchen kein Wert zugeteilt wurde) bezieht.

Bei diesem Beispiel handelt es sich eher um eine Funktion mit mehreren Anweisungen als um ein Programm. Hier werden zwar Zeilenumbrüche gezeigt, Sie würden den Text aber wie z.B.: **Define fact(n)=Func:Local…** als eine ununterbrochene Zeile eingeben. Die Auslassung (...) zeigt an, dass der Text der Eingabezeile über den Bildschirm hinausgeht.

Beispiel:

```
  Define fact(n)=Func:
❶ Local m:
  While n>1:
    n*m→m: n-1→n:
  EndWhile:
  Return m:
  EndFunc
```

❶ Der lokalen Variablen m wurde kein Anfangswert zugeteilt.

In obigem Beispiel existiert die lokale Variable m unabhängig von einer etwaigen außerhalb der Funktion vorliegenden Variablen *m*.

Lokale Variablen müssen initialisiert werden

Allen lokalen Variablen ist ein Anfangswert zuzuteilen, bevor diese aufgerufen werden können.

```
  Define fact(n)=Func:
❶ Local m: 1→m:
  While n>1:
    n*m→m: n-1→n:
  EndWhile:
  Return m:
  EndFunc
```

❶ 1 wird als Anfangswert für m gespeichert.

Der TI-89 Titanium / Voyage™ 200 Graphischer Rechner kann symbolische Berechnungen nicht mit lokalen Variablen durchführen.

Durchführung symbolischer Berechnungen

Soll eine Funktion oder ein Programm eine symbolische Berechnung durchführen, so müssen Sie anstelle einer lokalen eine globale Variable verwenden. Hierbei ist allerdings zu beachten, dass die globale Variable nicht bereits außerhalb des Programms vorhanden sein darf. Folgende Methoden können Ihnen bei diesem Vorgang behilflich sein.

- Stellen Sie einen Bezug zum Namen einer globalen Variablen (in der Regel zwei oder mehr Zeichen) her, die wahrscheinlich nicht außerhalb der Funktion bzw. des Programms existiert.

- Verwenden Sie in der Funktion bzw. im Programm **DelVar**, um die globale Variable, falls vorhanden, vor deren Aufruf zu löschen. (**DelVar** löscht weder gesperrte noch archivierte Variablen.)

Zeichenkettenoperationen

Zeichenketten (Strings) dienen zum Speichern und Anzeigen von Text. Sie können eine Zeichenkette direkt eingeben oder in eine Variable ablegen.

So werden Zeichenketten benutzt

Eine Zeichenkette ist eine Folge von Zeichen, die in “Anführungszeichen” eingeschlossen sind. Beim Programmieren können Zeichenketten benutzt werden, um Informationen anzuzeigen oder den Benutzer zu einer Aktion aufzufordern. Zum Beispiel:

```
Disp "The result is",answer
```

– oder –

```
Input "Enter the angle in degrees",ang1
```

– oder –

```
"Enter the angle in degrees"→str1
Input str1,ang1
```

Manche Eingabebefehle (wie etwa **InputStr**) legen eine Benutzereingabe automatisch als Zeichenkette ab, wodurch das Eingeben von Anführungszeichen nicht erforderlich ist.

Eine Zeichenkette kann nicht mathematisch ausgewertet werden, und zwar auch, wenn es sich anscheinend um einen numerischen Term handelt. Beispiel: Die Zeichenkette "`61`" stellt die Zeichen "`6`" und "`1`" dar und nicht die Zahl `61`.

Sie können zwar keine Zeichenkette wie "`61`" oder "`2x+4`" in einer Berechnung benutzen, aber Sie können eine Zeichenkette mit dem Befehl **expr** in einen numerischen Term umwandeln.

Befehle für Zeichenketten

Hinweis: Im Modul *Technische Referenz* finden Sie die Syntax für alle Befehle und Funktionen des TI-89 Titanium / Voyage™ 200 PLT.

Befehl	Beschreibung
#	Wandelt eine Zeichenkette in einen Variablennamen um. Dies wird als Umleitung bezeichnet.
&	Fügt zwei Zeichenketten zu einer einzigen zusammen (hängt sie aneinander an).
char	Liefert das Zeichen, das dem angegebenen Zeichencode entspricht. Ist das Gegenstück zum Befehl **ord**.
dim	Liefert die Anzahl der Zeichen in einer Zeichenkette.

Befehl	Beschreibung
expr	Wandelt einen Zeichenkettenausdruck in einen Term um und führt diesen aus. Ist das Gegenstück zum Befehl **string**. **Wichtig:** Manche Befehle, die der Benutzereingabe dienen, legen den eingegebenen Wert als Zeichenkette ab. Bevor Sie eine mathematische Operation auf diesen Wert anwenden können, müssen Sie ihn in einen numerischen Term umwandeln.
format	Liefert einen Term entsprechend der Formatschablone (fixed, scientific, engineering etc.) als Zeichenfolge.
inString	Durchsucht eine Zeichenkette, um festzustellen, ob sie den angegebenen Teilstring enthält. Wenn ja, gibt **inString** die Position des Zeichens zurück, an dem der Teilstring zum ersten Mal in der Zeichenkette beginnt.
left	Liefert eine bestimmte Anzahl von Zeichen ab dem Anfang der Zeichenkette (links).
mid	Liefert eine bestimmte Anzahl von Zeichen ab einer beliebigen Position in einer Zeichenkette.
ord	Liefert den Zeichencode des ersten Zeichens in einer Zeichenkette. Ist das Gegenstück zum Befehl **char**.
right	Liefert eine bestimmte Anzahl von Zeichen ab dem Ende der Zeichenkette (rechts).
rotate	Führt eine Rotation der Zeichen innerhalb einer Zeichenfolge aus. Der Standard ist ‑1 (ein Zeichen nach rechts rotieren).

Befehl	Beschreibung
shift	Verschiebt die Zeichen in einer Zeichenfolge und ersetzt sie durch ein Leerzeichen. Der Standard ist -1 (um ein Zeichen nach rechts verschieben und durch ein Leerzeichen ersetzen). Beispiel: shift("abcde",2)⇒"cde " oder shift("abcde")⇒" abcd"
string	Wandelt einen numerischen Term in eine Zeichenkette um. Ist das Gegenstück zum Befehl **expr**.

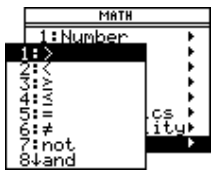
Bedingungsprüfungen

Bedingungsprüfungen dienen dazu, in Programmen eine Entscheidung über den Fortgang der Ausführung zu treffen. So kann ein Programm z. B. entscheiden, eine Aktion auszuführen, wenn eine Bedingung wahr ist und eine andere, wenn die Bedingung falsch ist. Bedingungsprüfungen werden in Kontrollstrukturen wie **If...EndIf** und in Schleifen wie **While...EndWhile** benutzt (weiter hinten in dieser Einheit erläutert).

Einen Testoperator eingeben

- Geben Sie den Operator unmittelbar mit der Tastatur ein.
  – oder –

- Drücken Sie 2nd [MATH], und wählen Sie **8:Test**. Wählen Sie dann aus dem Menü den Operator.
  – oder –



- Zeigen Sie die integrierten Funktionen mit folgenden Tasten an:
  [CATALOG]
  2nd [CATALOG]
  Die Testoperatoren werden im unteren Bereich des Menüs F2 Built-in angezeigt.

Relationale Bedingungsprüfungen

Mit relationalen Operatoren können Sie eine Bedingungsprüfung definieren, in der zwei Werte verglichen werden. Die Werte können Zahlen, Terme, Listen oder Matrizen sein, aber sie müssen vom gleichen Typ und von gleicher Dimension sein.

Operator	Wahr, wenn:	Beispiel
>	Größer als	a>8
<	Kleiner als	a<0
≥	Größer oder gleich	a+b≥100
≤	Kleiner oder gleich	a+6≤b+1

Operator	Wahr, wenn:	Beispiel
=	Gleich	list1=list2
≠	Ungleich	mat1≠mat2

Hinweis: Mit der Tastatur können Sie eingeben:
>= for ≥
<= for ≤
/= for ≠
(Für das Zeichen / drücken Si ÷.)

Boolesche Bedingungsprüfungen

Mit Booleschen Operatoren können Sie die Ergebnisse zweier getrennter Prüfungen miteinander kombinieren.

Operator	Wahr, wenn:	Beispiel
and	Beide Prüfungen wahr sind	a>0 and a≤10
or	Mindestens eine Prüfung wahr ist	a≤0 or b+c>10
xor	Eine Prüfung wahr und die andere falsch ist.	a+6<b+1 xor c<d

Die Funktion “Nicht”

Die Funktion “Nicht” (**not**) ändert das Ergebnis einer Prüfung von wahr in falsch und umgekehrt. Zum Beispiel:

not x>2	istwahr, wenn	x≤2
	falsch, wenn	x>2

Hinweis: Wenn Sie **not** auf dem Hauptbildschirm benutzen, wird es im Protokoll-Bereich als **~** angezeigt. Beispiel: **not x>2** wird angezeigt als **~(x>2)**.

If, Lbl und Goto zur Programmablaufsteuerung benutzen

In einer **If...EndIf** Struktur wird eine Bedingungsprüfung benutzt, um zu entscheiden, ob einer oder mehrere Befehle ausgeführt werden. Auch die Befehle **Lbl** (Label) und **Goto** können in einem Programm benutzt werden, um an eine andere Stelle zu verzweigen (zu springen).

Das Menü F2 Control

Die Eingabe von **If...EndIf** Strukturen nehmen Sie mit dem Menü F2 **Control** des Programmeditors vor.

Der Befehl **If** ist unmittelbar im F2-Menü enthalten.

Sie können ein Untermenü mit weiteren **If** -Strukturen aufrufen, indem Sie **2:If...Then** wählen.

Wenn sie eine Struktur wie **If...Then...EndIf** auswählen, wird eine Schablone an der Cursorposition eingefügt.

```
:If | Then ❶
:EndIf
```

❶ Der Cursor wird so plaziert, dass Sie eine Bedingungsprüfung eingeben können.

If-Befehl

Soll nur ein Befehl ausgeführt werden, wenn eine Bedingungsprüfung wahr ergibt, benutzen Sie die allgemeine Form:

```
  :If x>5
❶ :   Disp "x is greater than 5"
❷ :Disp x
```

❶ Wird nur ausgeführt, wenn x>5; andernfalls übersprungen.
❷ Wert von x wird stets angezeigt.

In diesem Beispiel müssen Sie x einen Wert zuweisen (in x speichern), bevor der Befehl **If** ausgeführt wird.

Hinweis: Verwenden Sie Einrückungen, um ein Programm leichter lesbar und verständlicher zu machen.

If...Then...EndIf Strukturen

Um mehrere Befehle auszuführen, wenn eine Bedingungsprüfung wahr ergibt, benutzen Sie die Struktur:

```
  :If x>5 Then
❶ :  Disp "x is greater than 5"
❶ :  2*x→x
❷ :EndIf
  :Disp x
```

❶ Wird nur ausgeführt, wenn x>5.
❷ Zeigt Wert von:
- 2x wenn x>5
- x wenn x≤5

Hinweis: EndIf markiert das Ende des **Then**-Blocks, der ausgeführt wird, wenn die Bedingung wahr ist.

If...Then...Else... EndIf Strukturen

Verwenden Sie folgende Struktur, um eine Gruppe von Befehlen auszuführen, wenn eine Bedingungsprüfung wahr ergibt und eine andere Gruppe, wenn sie falsch ergibt:

```
  :If x>5 Then
❶ :  Disp "x is greater than 5"
❶ :  2*x→x
  :Else
❷ :  Disp "x is less than or
❷    equal to 5"
  :  5*x→x
  :EndIf
❸ :Disp x
```

❶ Wird nur ausgeführt, wenn x>5.
❷ Wird nur ausgeführt, wenn x≤5.
❸ Zeigt Wert an:
- 2x wenn x>5
- 5x wenn x≤5

If...Then...ElseIf... EndIf Strukturen

Mit einer komplexeren Form des **If**-Befehls können Sie eine Folge von Bedingungen überprüfen. Nehmen Sie an, Ihr Programm fordert zur Eingabe einer von vier Optionen auf. Benutzen Sie zum Abprüfen der einzelnen Optionen (**If Choice=1**, **If Choice=2**, etc.), eine **If...Then...ElseIf...EndIf** Struktur.

Weitere Informationen und Beispiele finden Sie im Modul *Technische Referenz*.

Die Befehle Lbl und Goto

Außerdem können Sie den Ablauf eines Programms mit den Befehlen **Lbl** (label) und **Goto** steuern.

Den Befehl **Lbl** benutzen Sie, um einer bestimmten Stelle des Programms eine Marke (einen Namen) zuzuweisen.

Lbl *labelName*

└ Der Name, den Sie dieser Stelle des Programms zuweisen (die Regeln für die Namensgebung sind die gleichen wie für Variablennamen).

Sie können dann von einer beliebigen Stelle des Programms mit dem Befehl **Goto** an die Stelle verzweigen, die Sie mit dieser Marke versehen haben.

Goto *labelName*

└ Gibt an, zu welchem **Lbl** Befehl verzweigt wird

Da der Befehl **Goto** ein unbedingter Befehl ist (er verzweigt stets zur angegebenen Marke), wird er häufig in Verbindung mit **If** benutzt, um die Verzweigung mit einer Bedingungsprüfung zu verbinden. Zum Beispiel:

```
  :If x>5
❶ :  Goto GT5
❷ :Disp x
  :--------
  :--------
  :Lbl GT5
  :Disp "The number was > 5”
```

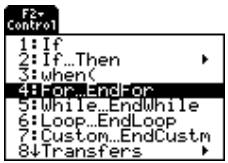
❶ Wenn x>5, wird direkt zur Marke GT5 verzweigt.
❷ Dieses Beispielprogramm muß Befehle enthalten, die die Ausführung von Lbl GT5 verhindern (z. B. **Stop**), wenn x<= 5.

Schleifen für die Wiederholung von Befehlen verwenden

Sie verwenden Schleifen, um eine Gruppe gleicher Befehle mehrmals hintereinander auszuführen. Es stehen Ihnen mehrere Schleifentypen zur Verfügung. Jeder Schleifentyp bietet Ihnen auf der Grundlage einer Bedingungsprüfung eine andere Möglichkeit zum Beenden (Verlassen) der Schleife.

Das Menü F2 Control

Zur Eingabe der meisten Schleifenbefehle verwenden Sie das F2 **Control** Menü des Programmeditors.



Wenn Sie die Schleife auswählen, werden der Schleifenbefehl und sein **End** Befehl an der Cursorposition eingefügt.

```
:For | ❶
:EndFor
```

❶ Sind für die Schleife Argumente erforderlich, wird der Cursor hinter dem einleitenden Schleifenbefehl angezeigt.

Sie können dann mit der Eingabe der Befehle beginnen, die in der Schleife ausgeführt werden.

Hinweis: Ein Schleifenbefehl markiert den Anfang der Schleife. Der entsprechende **End** Befehl markiert das Schleifenende.

For...EndFor Schleifen

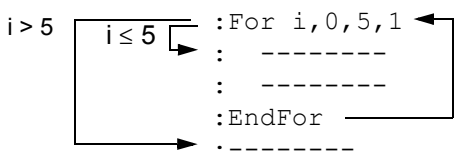
Eine **For...EndFor** Schleife benutzt einen Zähler, um die Anzahl der Schleifenwiederholungen zu steuern. Die Syntax des Befehls **For** ist:

Hinweis: Der Endwert kann kleiner sein als der Anfangswert, dann muß jedoch ein negatives Inkrement benutzt werden.

For(*variable*, *anfangswert*, *endwert* [, *inkrement*]**)**
❶ ❷ ❸ ❹

❶ Variable, die als Zähler dient
❷ Wert des Zählers zu Beginn des ersten Durchlaufs
❸ Die Schleife wird beendet, wenn *variable* diesen Wert überschreitet
❹ Wird nach jedem Durchlauf zum Zähler addiert. “Inkrement” ist optional; wenn kein Inkrement angegeben wird, wird 1 verwendet.

Bei Ausführung von **For** wird der Wert der variablen mit dem endwert verglichen. Ist variable nicht größer (bzw. bei negativem Inkrement kleiner) als endwert, wird die Schleife ausgeführt; andernfalls wird die Programmausführung mit dem nach **EndFor** folgenden Befehl fortgesetzt.



Hinweis: Der Befehl **For** erhöht die Zählervariable automatisch, so dass das Programm die Schleife nach einer bestimmten Anzahl von Wiederholungen beenden kann.

Nach dem Durchlauf der Schleife (**EndFor**), wird die Programmsteuerung wieder an den Befehl **For** übergeben, variable erhöht und mit endwert verglichen.

Zum Beispiel:

```
   :For i,0,5,1
❶  :  Disp I
   :EndFor
❷  :Disp i
```

❶ Zeigt 0, 1, 2, 3, 4 und 5 an.
❷ Zeigt 6 an. Wenn *variable* auf 6 erhöht ist, wird die Schleife nicht mehr ausgeführt.

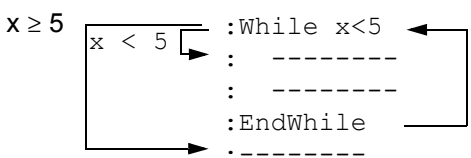
Hinweis: Sie können die Zählervariable als lokal deklarieren, wenn ihr Wert nach Programmende nicht mehr benötigt wird.

While...EndWhile Schleifen

Eine **While...EndWhile** Schleife wiederholt einen Block von Befehlen so lange, wie eine Bedingung wahr ist. Die Syntax des Befehls **While** ist:

While *bedingung*

Bei Ausführung von **While** wird die Bedingung ausgewertet. Ist bedingung wahr, wird die Schleife ausgeführt; andernfalls wird die Programmausführung mit dem nach **EndWhile** folgenden Befehl fortgesetzt.



Hinweis: Der Befehl **While** nimmt keine automatische Änderung der Bedingung vor. Sie müssen Befehle in die Schleife aufnehmen, die das Beenden der Schleife ermöglichen.

Nach dem Durchlauf der Schleife (**EndWhile**), wird die Programmsteuerung wieder an den Befehl **While** übergeben, wo bedingung erneut überprüft wird.

Damit die Schleife mindestens einmal durchlaufen wird, muß die bedingung anfänglich wahr sein.

- Allen Variablen, die in bedingung benutzt werden, muß vor dem Befehl **While** einWert zugewiesen werden. (Sie können die Werte fest ins Programm einfügen oder den Benutzer zur Eingabe auffordern).
- Die Schleife muß Befehle enthalten, welche die Werte in der bedingung ändern, so dass diese schließlich falsch ergibt. Andernfalls bleibt bedingung stets “wahr”, und das Programm kann die Schleife nicht verlassen (Endlosschleife).

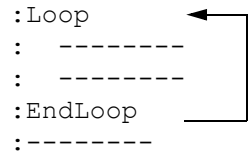
Zum Beispiel:

```
❶ :0→x
  :While x<5
❷ :  Disp x
❸ :  x+1→x
  :EndWhile
❹ :Disp x
```

❶ Initialisiert x.
❷ Zeigt 0, 1, 2, 3 und 4 an.
❸ Inkrementiert x.
❹ Zeigt 5 an. Wenn x den Wert erreicht 5, wird die Schleife nicht mehr ausgeführt.

Loop...EndLoop Schleifen

Ein **Loop...EndLoop** Befehl erzeugt eine Schleife, die im Prinzip endlos wiederholt wird, da der Befehl **Loop** keine Argumente besitzt.



Eine solche Schleife ist mit Befehlen zu versehen, die dem Programm das Beenden der Schleife erlauben. Häufig benutzt werden dafür: **If**, **Exit**, **Goto**, und **Lbl** (Marke). Zum Beispiel:

```
  :0→x
  :Loop
  :  Disp x
  :  x+1→x
❶ :  If x>5
  :    Exit
  :EndLoop
❷ :Disp x
```

❶ Ein **If** überprüft die Bedingung.
❷ Schleife wird beendet und Ausführung hier fortgesetzt, wenn x den Wert 6 erreicht.

Hinweis: Der Befehl **Exit** beendet die aktuelle Schleife.

In diesem Beispiel kann der Befehl **If** an einer beliebigen Stelle in der Schleife stehen.

Wenn der Befehl If hier steht:	Wird die Schleife:
Am Anfang der Schleife	Nur ausgeführt, wenn die Bedingung wahr ist.
Am Ende der Schleife	Mindestens einmal ausgeführt; danach nur dann erneut, wenn die Bedingung wahr ergibt.

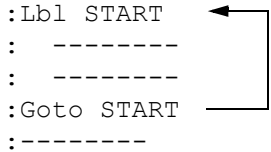
Der Befehl **If** könnte auch einen **Goto** Befehl benutzen, um zu einem Befehl **Lbl** (Marke) zu springen.

Ein sofortiger erneuter Schleifendurchlauf

Der Befehl **Cycle** übergibt die Ausführungssteuerung sofort an den einleitenden Schleifenbefehl und leitet einen neuen Durchlauf ein, bevor der aktuelle Durchlauf vollständig ausgeführt ist. Sie können diesen Befehl in folgenden Schleifentypen benutzen: **For...EndFor**, **While...EndWhile**, und **Loop...EndLoop**.

Schleifen aus Lbl und Goto

Zwar sind die Befehle **Lbl** (Marke) und **Goto** eigentlich keine Schleifenbefehle, aber sie können zum Aufbau einer Endlosschleife benutzt werden. Zum Beispiel:



Wie bei der Konstruktion mit **Loop...EndLoop** sind in die Schleife Befehle aufzunehmen, die das ordnungsgemäße Beenden ermöglichen.

Den TI-89 Titanium / Voyage™ 200 konfigurieren

Programme können Befehle enthalten, die die Konfiguration von TI-89 Titanium /Voyage™ 200 Graphischer Rechner ändern. Da Modusänderungen besonders nützlich sind, vereinfacht das Menüleistenmenü **Mode** des Programmeditors die Eingabe der korrekten Syntax für den Befehl **setMode**.

Konfigurationsbefehle

Befehl	Beschreibung
getConfg	Liefert eine Liste der Taschenrechnermerkmale.
getFold	Liefert den Namen des aktuellen Verzeichnisse.
getMode	Liefert die aktuelle Einstellung eines bestimmten Modus.
getUnits	Zeigt eine Liste der Standardeinheiten an.
setFold	Macht ein Verzeichnis zum aktuellen Verzeichnis.

Befehl	Beschreibung
setGraph	Aktiviert ein bestimmtes Graphikformat (**Coordinates**, **Graph Order**, etc.).
setMode	Aktiviert jeden Modus außer **Current Folder**.
setTable	Setzt den angegebenen Tabellen-Setupparameter (**tblStart**, Δ**tbl**, etc.).
setUnits	Setzt die Standardeinheiten für angezeigte Ergebnisse.
switch	Setzt in einem geteilten Bildschirm das aktive Fenster oder liefert die Nummer des aktiven Fensters.

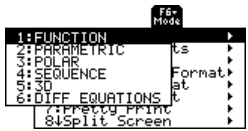
Hinweis: Die in den Funktionen **setMode( )**, **getMode( )**, **setGraph( )** und **setTable( )** verwendeten Parameter-/Modusstrings werden bei Verwendung in einem Programm nicht in andere Sprachen übersetzt (siehe Modul *Technische Referenz*).

Den Befehl SetMode eingeben

Im Programmeditor:

1. Bringen Sie den Cursor an die Stelle, an der Sie den Befehl **setMode** einfügen möchten.

2. Drücken Sie folgende Tasten:
   [2nd] [F6]
   [F6]
   Hierdurch wird eine Modusliste angezeigt.



   Hinweis: Im Modusmenü kann der Modus **Current Folder** nicht eingestellt werden. Verwenden Sie dazu den Befehl **setFold**.

3. Wählen Sie einen Modus, um ein Menü mit dessen gültigen Einstellungen aufzurufen.
4. Wählen Sie eine Einstellung.

Die korrekte Syntax wird in Ihr Programm eingefügt.

```
:setMode("Graph","FUNCTION")
```

Benutzereingaben abfragen und Programmausgabe einstellen

Zwar können Werte fest in ein Programm integriert (oder vorab in Variablen gespeichert) werden, ein Programm kann jedoch auch während der Ausführung den Benutzer zu Eingaben auffordern. Umgekehrt, kann ein Programm auch Informationen ausgeben (anzeigen), etwa Ergebnisse von Berechnungen.

Das Menü F3 I/O

Die meistbenutzten Ein-/Ausgabebefehle können Sie mit dem Menü F3 **I/O** des Programmeditors eingeben.

Sie können ein Untermenü mit weiteren Befehlen aufrufen, indem Sie **1:Dialog** wählen.

Eingabebefehle

Befehl	Beschreibung
getKey	Liefert den Code der nächsten gedrückten Taste.
Input	Fordert den Benutzer zur Eingabe eines Ausdrucks auf. Wie dieser behandelt wird, hängt von der Eingabeart ab. Beispiel: • Ein numerischer Ausdruck wird als Term behandelt. • Ein in “Anführungszeichen” eingeschlossener Ausdruck wird als Zeichenkette behandelt. **Input** kann auch den Graphikbildschirm aufrufen und so dem Benutzer die Möglichkeit geben, die Variablen xc und yc (rc und θc im Polar-Modus) durch Verschieben des Graphikcursors zu aktualisieren.

Befehl	Beschreibung
InputStr	Fordert den Benutzer zur Eingabe eines Ausdrucks auf. Der Ausdruck wird stets als Zeichenkette behandelt, braucht also nicht in "Anführungszeichen" gestellt zu werden.
PopUp	Zeigt ein Popup-Menü an, damit der Benutzer einen Menüpunkt auswählen kann.
Prompt	Fordert den Benutzer zur Eingabe einer Folge von Ausdrücken auf. Wie für **Input** richtet sich die Behandlung der einzelnen Ausdrücke nach der Art der Eingabe.
Request	Zeigt ein Dialogfeld an, das den Benutzer zur Eingabe eines Ausdrucks auffordert. **Request** behandelt den Ausdruck stets als Zeichenkette.

Hinweis: Zeichenketten können nicht in Berechnungen benutzt werden. Sie können eine Zeichenkette jedoch mit dem Befehl **expr** in einen numerischen Term umwandeln.

Ausgabebefehle

Befehl	Beschreibung
ClrIO	Löscht den Programm-I/O-Bildschirm.
Disp	Zeigt einen Term oder eine Zeichenkette auf dem Programm-I/O-Bildschirm an. **Disp** kann auch den Inhalt des aktuellen Programm-I/O-Bildschirms ohne zusätzliche Informationen ausgeben.
DispG	Zeigt den aktuellen Inhalt des Graphikbildschirms an.
DispHome	Zeigt den aktuellen Inhalt des Hauptbildschirms an

Befehl	Beschreibung
DispTbl	Zeigt den aktuellen Inhalt des Tabellen-Bildschirms an.
Output	Zeigt einen Term oder eine Zeichenkette beginnend an den angegebenen Koordinaten auf dem Programm-I/O-Bildschirm an.
Format	Legt das Anzeigeformat für numerische Informationen fest.
Pause	Hält die Programmausführung so lange an, bis der Benutzer ENTER drückt. Sie können während der Pause einen Term anzeigen. Dies gibt dem Benutzer Zeit, die Ausgabe zu lesen, und er kann das Programm dann weiter ausführen lassen.
Text	Zeigt ein Dialogfeld mit einem bestimmten Text (einer Zeichenkette) an.

Hinweis:

- In einem Programm bewirkt das Ausführen einer Berechnung noch keine Anzeige des Ergebnisses. Sie müssen dazu einen Ausgabebefehl verwenden.

- Nach einem **Disp** bzw. einem **Output** Befehl wird die Programmausführung unmittelbar fortgesetzt. Daher empfiehlt sich das Einfügen eines **Pause** Befehls.

Befehle für die grafische Benutzeroberfläche

Befehl	Beschreibung
Dialog... endDlog	Definiert einen Programmblock (bestehend aus Befehlen wie **Title**, **Request**, etc.), der ein Dialogfeld anzeigt.

Befehl	Beschreibung
Toolbar... **EndTbar**	Definiert einen Programmblock (bestehend aus Title, Item etc.), der die Menüleisten-Menüs ersetzt. Die so umdefinierte Menüleiste ist nur während des Programms und nur bis zur Auswahl eines Menüpunktes wirksam. Anschließend wird die ursprüngliche Menüleiste angezeigt.
CustmOn... **CustmOff**	Aktiviert oder entfernt eine benutzerdefinierte Menüleiste.
Custom... **EndCustm**	Definiert einen Programmblock, der bei Drücken von [2nd] [CATALOG] eine benutzerdefinierte Menüleiste anzeigt. Diese Menüleiste bleibt wirksam, bis der Benutzer erneut [2nd] [CATALOG] drückt oder die Anwendung wechselt.
DropDown	Zeigt ein Dropdown-Menü in einem Dialogfeld an.
Item	Zeigt einen Menüpunkt für eine umdefinierte Menüleiste an.
Request	Erzeugt ein Eingabefeld in einem Dialogfeld.
Text	Zeigt eine Zeichenkette in einem Dialogfeld an.
Title	Zeigt den Titel eines Dialogfelds oder einen Menüleisten-Menütitel innerhalb einer Menüleiste an.

Hinweis:

- Wenn Sie ein Programm ausführen, in dem eine “Custom”- Menüleiste erzeugt wird, ist diese auch nach Programmende weiterhin verfügbar.

- **Request** und **Text** sind eigenständige Befehle, die auch außerhalb eines Dialogfelds oder eines Menüleisten-Programmblocks benutzt werden können.

Benutzermenüs erstellen

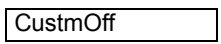
Mit der Benutzermenü-Funktion des TI-89 Titanium / Voyage™ 200 Graphischer Rechner können Sie eigene Menüleisten erstellen. In einem Benutzermenü können sich alle verfügbaren Funktionen, Anweisungen oder Zeichensätze befinden. Der TI-89 Titanium / Voyage™ 200 verfügt über ein vorgegebenes Benutzermenü, das Sie bearbeiten oder neu definieren können.

Das Benutzermenü ein- und ausschalten

Wenn Sie ein Benutzermenü erstellen, können Sie es entweder manuell ein- und ausschalten oder es von einem Programm automatisch ein- und ausschalten lassen.

Zum:	Drücken Sie:
Einschalten des Benutzermenüs	Im Hauptbildschirm oder einer anderen Anwendung: • Drücken Sie 2nd [CATALOG]. Im Hauptbildschirm oder einem Programm: • Führen Sie den Befehl **CustmOn** aus.

Zum:	Drücken Sie:
Ausschalten des Benutzermenüs	In allen Anwendungen: • Drücken Sie 2nd [CATALOG] erneut. – oder – • Gehen Sie zu einer anderen Anwendung. Bei Verwendung des vorgegebenen Benutzermenüs im Hauptbildschirm: 1. Wählen Sie das Menü **Tools** aus: 2nd [F7] F7 Select **3:CustmOff**. Dadurch wird **CustmOff** in die Eingabezeile eingefügt. 2. Drücken Sie ENTER. **CustmOff** können Sie auch in einem Programm verwenden.

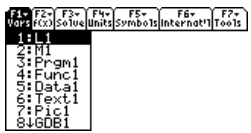


Hinweis: Wenn Sie das Benutzermenü einschalten, ersetzt es die normale Menüleiste. Sofern nicht ein anderes Benutzermenü erzeugt wurde, wird das vorgegebene angezeigt.

Benutzermenüs definieren

Arbeiten Sie mit folgender allgemeinen Struktur, um ein Benutzermenü zu erstellen.

```
:
Custom
: Title Titel des Menüs F1
:     Item Element 1
:     Item Element 2
:     …
: Title Titel des Menüs F2
:     …
: Title titleTitel des Menüs F2
:     …
 EndCustm
```



Hinweis: Wenn der Benutzer ein Menüelement wählt, wird der durch diesen Menübefehl definierte Text an der aktuellen Cursorposition eingefügt.

Beispiel:

```
:Custom

:Title "Vars"
:Item "L1":Item "M1":Item "Prgm1":Item "Func1":Item "Data1"
:Item "Text1":Item "Pic1":Item "GDB1":Item "Str1"

❶ :Title "f(x)"
❶ :Item "f(x)":Item "g(x)":Item "f(x,y)":Item "g(x,y)"
❶ :Item "f(x+h)":Item "Define f(x) ="
```

```
:Title "Solve"
:Item "Solve(":Item " and ":Item "{x,y}"
:Item "Solve( and ,{x,y})"

❷ :Title "Units"
❷ :Item "_m/_s^2":Item "_ft/_s^2":Item "_m":Item "_ft":Item "_l"
❷ :Item "_gal":Item "_\o\C":Item "_\o\F":Item "_kph":Item "_mph"

:Title "Symbols"
:Item "#":Item "\beta\":Item "?":Item "~":Item "&"

:Title "Internat'l"
:Item "\e`\":Item "\e'\":Item "\e^\":Item "\a`\"
:Item "\u`\":Item "\u^\":Item "\o^\":Item "\c,\":Item "\u..\"

:Title "Tools"
:Item "ClrHome":Item "NewProb":Item "CustmOff"

:EndCustm

:CustmOn
```

Hinweis: Das vorgegebene Benutzermenü Ihres Taschenrechners kann hiervon leicht abweichen.

Hinweis: Beachten Sie, dass "_\o\C" und "_\o\F" im Menü als °C und °F angezeigt werden. Siehe ebenso die internationalen akzentuierten Buchstaben.

Wenn Sie das vorgegebene Benutzermenü ändern möchten, verwenden Sie **3:Restore custom default** (wie unten beschrieben), um die Befehle für das vorgegebene Menü abzurufen. Kopieren Sie diese Befehle, erzeugen Sie mit dem Programmeditor ein neues Programm, und fügen Sie die Befehle in das leere Programm ein. Dann können Sie die Befehle nach Belieben verändern.

Hinweis: Dadurch werden alle Befehle in eine einzige Zeile eingefügt. Sie müssen sie nicht in separate Zeilen aufteilen.

Es kann nur jeweils ein Benutzermenü erstellt und verwendet werden. Wenn Sie mehrere benötigen, schreiben Sie für jedes Benutzermenü ein separates Programm. Führen Sie dann das Programm für das erforderliche Menü aus.

Das vorgegebene Benutzermenü wiederherstellen

Gehen Sie wie folgt vor, wenn Sie das vorgegebene Benutzermenü wiederherstellen möchten:

1. Wählen Sie im normalen Menü des Aussgangsbildschirms (nicht im Benutzermenü) **Clean Up**:
   [2nd] [F6]
   [F6]

2. Wählen Sie **3:Restore custom default**.

   Dadurch werden die zum Erstellen des Standardmenüs benötigten Befehle in die Eingabezeile kopiert.

3. Drücken Sie [ENTER], um die Befehle auszuführen und das vorgegebene Menü wiederherzustellen.

Das vorige Benutzermenü wird dadurch gelöscht. Wurde dieses Menü mit einem Programm erstellt, kann es später durch erneute Ausführung des Programms wieder erstellt werden.

Eine Tabelle oder eine Graphik erzeugen

Mit den Befehlen des vorliegenden Abschnitts können Sie eine Tabelle oder eine Graphik erzeugen, die mit einer oder mehreren Funktionen oder Gleichungen erstellt werden.

Befehle für Tabellen

Befehl	Beschreibung
DispTbl	Zeigt den aktuellen Inhalt des Tabellenbildschirms an.
setTable	Setzt die Tabellenparameter Graph <–> Table oder Independent. (Sie können die beiden anderen Tabellenparameter setzen, indem Sie die gewünschten Werte in die Systemvariablen **tblStart** und **Δtbl** abspeichern.)
Table	Erstellt auf der Basis von Termen oder Funktionen eine Tabelle und zeigt sie an.

Befehle für Graphiken

Befehl	Beschreibung
ClrGraph	Löscht Zeichnungen von Funktionen oder Termen, die mit dem Befehl **Graph** gezeichnet wurden.
Define	Erzeugt eine benutzerdefinierte Funktion.
DispG	Zeigt den aktuellen Inhalt des Graphikbildschirms an.
FnOff	Hebt die Auswahl aller (oder nur der angegebenen) Y= Funktionen auf.
FnOn	Wählt alle (oder nur die angegebenen) Y= Funktionen aus.
Graph	Stellt einen oder mehrere Terme graphisch dar, wobei der aktuelle Graphikmodus benutzt wird.
Input	Zeigt den Graphikbildschirm an, damit der Benutzer die Variablen xc und yc (rc und θc im Polar-Modus) durch Verschieben des Graphikcursors aktualisieren kann.
NewPlot	Erstellt eine neue Plot-Definition für Statistikdaten.
PlotsOff	Hebt die Auswahl aller (oder nur der angegebenen) Statistikplots auf.
PlotsOn	Wählt alle (oder nur die angegebenen) Statistikplots aus.
setGraph	Ändert Einstellungen für die verschiedenen Graphikformate (**Coordinates**, **Graph Order**, etc.).
setMode	Setzt den Graph-Modus und andere Modi.
Style	Setzt den Anzeigestil für eine Funktion.
Trace	Ermöglicht einem Programm das Tracen eines Graphen.

Befehl	Beschreibung
ZoomBox – bis – **ZoomTrig**	Führt alle **Zoom**-Operationen aus, die im Menü F2 des Y= Editor, des Window-Editors und des Graphikbildschirms zur Verfügung stehen.

Hinweis: Nähere Erläuterungen zur Verwendung von **setMode** sind verfügbar.

Befehle für Graphikbilder und Graphik-Einstellungen

Befehl	Beschreibung
AndPic	Zeigt den Graphikbildschirm an und überlagert ihn in einer **AND**-Operation mit einem gespeicherten Graphikbild.
CyclePic	Zeigt eine Folge gespeicherter Graphikbilder bewegt an.
NewPic	Erzeugt eine Graphikbildvariable auf der Grundlage einer Matrix.
RclGDB	Stellt alle Einstellungen wieder her, die in einer GDB-Variablen gespeichert sind.
RclPic	Zeigt den Graphikbildschirm an und überlagert ihn in einer **OR**-Operation mit einem gespeicherten Graphikbild.
RplcPic	Löscht den Graphikbildschirm und zeigt ein gespeichertes Graphikbild an.
StoGDB	Legt die aktuellen Graph-Einstellungen in einer GDB-Variablen ab.
StoPic	Kopiert den Graphikbildschirm (bzw. einen angegebenen rechteckigen Ausschnitt) in eine Graphikbildvariable.

Befehl	Beschreibung
XorPic	Zeigt den Graphikbildschirm an und überlagert ihn in einer **XOR**-Operation mit einem gespeicherten Graphikbild.

Hinweis: Weitere Erläuterungen zu Graphikbildern und Graphik-Einstellungen finden Sie in *Weitere Darstellungsarten*.

Im Graphikbildschirm zeichnen

Mit Befehlen dieses Abschnitts können sie ein Zeichenobjekt im Graphikbildschirm erstellen.

Pixel-Koordinaten und Punkt-Koordinaten

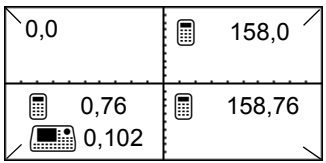
Beim Zeichnen eines Objekts benutzen Sie eines von zwei Koordinatensystemen zur Angabe eines Orts auf dem Bildschirm.

- **Pixel-Koordinaten** — Bezieht sich auf die Pixel, die den physischen Bildschirm bilden. Diese sind unabhängig vom Ansichtfenster, da der Bildschirm immer:

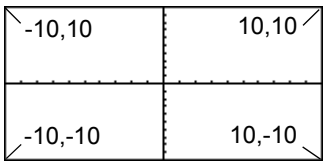
  159 (0 bis 158) Pixel breit, 77 (0 bis 76) Pixel hoch.

  239 (0 bis 238) Pixel breit, 103 (0 bis 102) Pixel hoch.

- **Punkt-Koordinaten** — Bezieht sich auf die Koordinaten für das aktuelle Ansichtfenster (gemäß Definition im Window-Editor).



Pixel-Koordinaten
(unabhängig vom Ansichtfenster)



Punkt-Koordinaten
(für Standard-Ansichtfenster)

Hinweis: Erläuterungen zu Pixel-Koordinaten in geteilten Bildschirmen finden Sie in *Geteilte Bildschirme*.

Viele Zeichenbefehle liegen in zwei Formen vor: einer für Pixel- und einer für Punkt-Koordinaten.

Hinweis: Pixel-Befehle beginnen mit **PxI**, z. B. **PxIChg**.

Gezeichnete Objekte löschen

Befehl	Beschreibung
ClrDraw	Löscht alle gezeichneten Objekte vom Graphikbildschirm.

Einen Punkt oder ein Pixel zeichnen

Befehl	Beschreibung
PtChg bzw. **PxlChg**	PxlChg kehrt ein Pixel an den angegebenen Koordinaten um (aktiviert bzw. deaktiviert es). **PtChg**, der entsprechende Befehl mit Punkt-Koordinaten, wirkt sich auf das Pixel aus, das dem Punkt am nächsten liegt. Ist das Pixel ausgeschaltet, wird es eingeschaltet. Ist das Pixel eingeschaltet, wird es ausgeschaltet.
PtOff bzw. **PxlOff**	Schaltet das Pixel an den angegebenen Koordinaten aus (löscht es). **PtOff**, der entsprechende Befehl mit Punkt-Koordinaten, wirkt sich auf das Pixel aus, das dem Punkt am nächsten liegt.
PtOn bzw. **PxlOn**	Schaltet das Pixel an den angegebenen Koordinaten ein (zeigt es an). **PtOn**, der entsprechende Befehl mit Punkt-Koordinaten, wirkt sich auf das Pixel aus, das dem Punkt am nächsten liegt.
PtTest bzw. **PxlTest**	Gibt wahr bzw. falsch zurück und dient dazu, zu ermitteln, ob ein Pixel ein (wahr) oder aus (falsch) ist.
PtText bzw. **PxlText**	Zeigt eine Zeichenkette an den angegebenen Koordinaten an.

Gerade und Kreise zeichnen

Befehl	Beschreibung
Circle bzw. **PxlCrcl**	Zeichnet, löscht oder invertiert einen Kreis mit angegebenem Mittelpunkt und Radius.

Befehl	Beschreibung
DrawSlp	Zeichnet eine Gerade mit angegebener Steigung durch einen angegebenen Punkt.
Line bzw. **PxlLine**	Zeichnet, löscht oder invertiert eine Gerade zwischen zwei Koordinatenpaaren.
LineHorz bzw. **PxlHorz**	Zeichnet, löscht oder invertiert eine horizontale Gerade durch eine angegebene Zeilenkoordinate.
LineTan	Zeichnet eine Tangente für einen angegebenen Term durch den angegebenen Punkt. (Es wird nur die Tangente gezeichnet, nicht der Term).
LineVert bzw. **PxlVert**	Zeichnet, löscht oder invertiert eine vertikale Gerade durch die angegebene Spaltenkoordinate.

Terme zeichnen

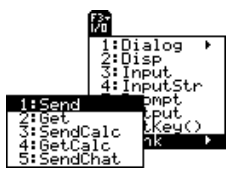
Befehl	Beschreibung
DrawFunc	Zeichnet einen angegebenen Term.
DrawInv	Zeichnet die Inverse eines angegebenen Terms.
DrawParm	Zeichnet eine Parameterdarstellung , wobei die angegebenen Terme als x- und y-Komponenten benutzt werden.
DrawPol	Zeichnet einen angegebenen Polar-Term.
DrwCtour	Zeichnet Konturen im 3D-Darstellungsmodus.
Shade	Zeichnet zwei Terme und schraffiert die Bereiche, in denen *term1* < *term2*.

Auf einen anderen TI-89 Titanium / Voyage™ 200, ein CBL 2 oder CBR zugreifen

Wenn Sie zwei TI-89 Titanium / Voyage™ 200 Graphischer Rechner miteinander verbinden (wie in *Geräte verbinden und aufrüsten* erläutert), können Programme beider Geräte Variablen miteinander austauschen. Wenn Sie einen TI-89 Titanium / Voyage™ 200 mit einem optional erhältlichen Calculator-Based Laboratory™ (CBL 2™) oder Calculator-Based Ranger™ (CBR™) verbinden, kann ein Programm vom TI-89 Titanium / Voyage™ 200 auf das CBL 2 oder CBR zugreifen.

Das Menü F3 I/O

Die Befehle in diesem Abschnitt geben Sie mit dem Menü F3 **I/O** des Programmeditors ein.



1. Drücken Sie F3, und wählen Sie **8:Link**.
2. Wählen Sie einen Befehl.

Auf einen anderen TI-89 Titanium / Voyage™ 200 zugreifen

Sind zwei TI-89 Titanium / Voyage™ 200 miteinander gekoppelt, fungiert das eine Gerät als Sender, das andere als Empfänger.

Befehl	Beschreibung
GetCalc	Im empfangenden Gerät ausgeführt. Richtet das Gerät für den Empfang einer Variablen über den Ein-/Ausgabeanschluß ein. • Nachdem das empfangende Gerät **GetCalc** ausgeführt hat, muß das sendende **SendCalc** ausführen. • Nachdem das sendende Gerät **SendCalc** ausgeführt hat, wird die Variable im empfangenden Gerät gespeichert (in der mit **GetCalc** angegebenen benannten Variablen).
SendCalc	Im sendenden Gerät ausgeführt. Sendet eine Variable über den I/O-Port an das empfangende Gerät. • Bevor das sendende Gerät **SendCalc** ausführt, muß das empfangende Gerät **GetCalc** ausführen.
SendChat	Wird auf dem sendenden Gerät als allgemeine Alternative zu **SendCalc** ausgeführt. Besonders hilfreich, wenn es sich beim empfangenden Gerät um einen TI-92 handelt (oder für ein allgemeines "Chat"-Programm, das sowohl die Verwendung eines TI-92, TI-92 Plus als auch eines Voyage™ 200 zuläßt).

Hinweis: Ein Musterprogramm, das das sendende und das empfangende Gerät so synchronisiert, daß **GetCalc** und **SendCalc** in der korrekten Reihenfolge ausgeführt werden, finden Sie in “Variablen mit einem Programm übertragen” in *Geräte verbinden und aufrüsten*.

Auf ein CBL 2 oder CBR zugreifen

Nähere Erläuterungen finden Sie in dem zum CBL 2 oder CBR gehörigen Handbuch.

Befehl	Beschreibung
Get	Ruft eine Variable aus dem angeschlossenen CBL 2 oder CBR ab und speichert sie im TI-89 Titanium / Voyage™ 200.
Send	Sendet eine Listenvariable vom TI-89 Titanium / Voyage™ 200 an das CBL 2 oder CBR.

Programme debuggen und Fehler abfangen

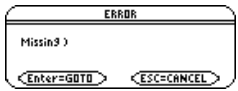
Im Rahmen der Programmerstellung können Sie mehrere Verfahren zur Fehlersuche und Fehlerbehebung anwenden. Sie können auch einen Fehlerbehandlungsbefehl in das Programm selbst aufnehmen.

Laufzeitfehler

Der erste Schritt beim Debuggen eines Programms ist, es auszuführen. Der TI-89 Titanium / Voyage™ 200 Graphischer Rechner überprüft bei Ausführung automatisch

jeden Befehl auf Syntaxfehler. Wird ein Fehler gefunden, wird eine entsprechende Meldung angezeigt.

- Sie können das Programm im Programmeditor anzeigen, indem Sie ENTER drücken. Der Cursor wird ungefähr an der Stelle angezeigt, an der der Fehler auftrat.



- Sie können die Programmausführung abbrechen und zum Hauptbildschirm zurückzukehren, indem Sie ESC drücken.

Wenn Ihr Programm dem Benutzer eine Auswahl aus mehreren Optionen anbietet, führen Sie das Programm unbedingt aus, und testen Sie jede Option.

Debugging-Verfahren

Laufzeitfehlermeldungen können Syntaxfehler erfassen und anzeigen, aber keine Fehler im logischen Aufbau des Programms. Die folgenden Verfahren können hier hilfreich sein.

- Verwenden Sie beim Testen keine lokalen Variablen: Auf diese Weise können Sie die Variablenwerte nach Programmende überprüfen. Deklarieren Sie die betreffenden Variablen als lokal, nachdem das Programm fehlerfrei ist.
- Fügen Sie vorübergehend **Disp** und **Pause** Befehle in das Programm ein, um die Werte wichtiger Variablen anzuzeigen.
    - **Disp** und **Pause** können nicht in einer benutzerdefinierten Funktion benutzt werden. Sie können die Funktion jedoch vorübergehend zu einem Programm machen, indem Sie **Func** und **EndFunc** in **Prgm** und **EndPrgm** ändern. Dann

können Sie **Disp** und **Pause** benutzen, um das Programm zu debuggen. Zum Abschluß entfernen Sie die **Disp** und **Pause** Befehle, und wandeln das Programm wieder in eine Funktion um.

- Um sicherzustellen, dass die Anzahl der Durchläufe einer Schleife korrekt ist, können Sie sich die Zählervariable oder die Werte der Bedingungsprüfung anzeigen lassen.
- Um festzustellen, ob ein Unterprogramm tatsächlich ausgeführt wird, können Sie am Anfang und am Ende der Routine eine Meldung anzeigen lassen, etwa `Entering subroutine` und `Exiting subroutine`.

Befehle für die Fehlerbehandlung

Befehl	Beschreibung
Try...EndTry	Definiert einen Programmblock, der dem Programm die Ausführung eines Befehls und ggf. das Abfangen eines Fehlers ermöglicht, der von diesem Befehl verursacht wurde.
ClrErr	Setzt den Fehlerstatus zurück und setzt die Fehlernummer in der Systemvariablen Errornum auf Null.
PassErr	Übergibt einen Fehler an die nächste Ebene des **Try...EndTry** Blocks.

Beispiel: Unterschiedliche Zugänge beim Programmieren

Das Programmierbeispiel im Modul *Mathematik-Schnellstart* stellt ein Programm dar, das den Benutzer auffordert, eine Ganzzahl einzugeben. Alle Ganzzahlen von 1 bis zur eingegebenen Ganzzahl werden addiert und das Ergebnis wird anzeigt.

Beispiel 1

In diesem Beispiel werden **InputStr** für die Eingabe, eine **While...EndWhile** Schleife für die Berechnung und **Text** für die Anzeige des Ergebnisses benutzt.

```
  :prog1()
  :Prgm
❶ :InputStr "Enter an integer",n
❷ :expr(n)→n
  :0→temp:1→I
❸ :While i≤n
:  :  temp+i→temp
:  :  i+1→I
❸ :EndWhile
❹ :Text "The answer is "&string(temp)
  :EndPrgm
```

❶ Eingabeaufforderung im Programm-I/O-Bildschirm.
❷ Wandelt die mit **InputStr** eingegebene Zeichenkette in einen Term um.
❸ Rechenschleife.
❹ Zeigt Ausgabe in einem Dialogfeld an.

Hinweis: Das Zeichen ≤ wird mit ♦ 0 (Null) erzeugt, das Zeichen **&** mit
TI-84 Plus: ♦ × (anfügen)
TI-84 Plus CE: 2nd **H**

Beispiel 2

In diesem Beispiel werden **Prompt** für die Eingabe, **Lbl** und **Goto** zum Erzeugen einer Schleife und **Disp** für die Anzeige des Ergebnisses benutzt.

```
  :prog2()
  :Prgm
❶ :Prompt n
  :0→temp:1→I
❷ :Lbl top
⋮ :  temp+i→temp
⋮ :  i+1→I
⋮ :  If i≤n
❷ :    Goto top
❸ :Disp temp
  :EndPrgm
```

❶ Eingabeaufforderung im Programm-I/O-Bildschirm.
❷ Rechenschleife.
❸ Zeigt Ausgabe im Programm-I/O-Bildschirm an.

Hinweis: Da **Prompt** `n` als numerischen Wert liefert, brauchen Sie `n` nicht mit **expr** umzuwandeln.

Beispiel 3

In diesem Beispiel wird **Dialog...EndDlog** zur Erzeugung von Dialogfeldern für die Ein- und Ausgabe benutzt. Das Ergebnis wird in einer **Loop...EndLoop** Schleife berechnet.

```
  :prog3()
  :Prgm
❶ :Dialog
⋮ :  Title "Enter an integer"
⋮ :  Request "Integer",n
❶ :EndDlog
❷ :expr(n)→n
  :0→temp:0→I
❸ :Loop
⋮ :  temp+i→temp
⋮ :  i+1→I
⋮ :  If i>n
⋮ :    Exit
❸ :EndLoop
❹ :Dialog
⋮ :  Title "The answer is"
⋮ :  Text string(temp)
❹ :EndDlog
  :EndPrgm
```

❶ Definiert ein Dialogfeld für die Eingabe.
❷ Wandelt die mit **Request** eingegebene Zeichenkette in einen Term um.
❸ Rechenschleife.
❹ Definiert ein Dialogfeld für die Ausgabe.

Beispiel 4

In diesem Beispiel werden die integrierten Funktionen des TI-89 Titanium / Voyage™ 200 zum Berechnen des Ergebnisses benutzt, wodurch keine Schleife notwendig ist.

```
  :prog4()
  :Prgm
❶ :Input "Enter an integer",n
❷ :sum(seq(i,i,1,n))→temp
❸ :Disp temp
  :EndPrgm
```

❶ Eingabeaufforderung im Programm-I/O-Bildschirm.
❷ Summenberechnung.
❸ Zeigt Ausgabe im Programm-I/O-Bildschirm an.

Hinweis: Da **Input** `n` als numerischen Wert liefert, brauchen Sie `n` nicht mit **expr** umzuwandeln.

Funktion	In diesem Beispiel benutzt für:
seq	Das Erzeugen der Folge der ganzen Zahlen von **1** bis **n**. **seq(***term*, *var*, *anf*, *ende* [,*schritt*]**)** ❶ ❷ ❸ ❸ ❹ ❶ Term zur Erzeugung der Folge ❷ Variable, die inkrementiert wird ❸ Anfangs- und Endwert von *var* ❹ Inkrement für *var* ; wenn keine Angabe, wird 1 benutzt

Funktion	In diesem Beispiel benutzt für:
sum	Bildet die Summe der ganzen Zahlen, die in der mit **seq** erzeugten Liste enthalten sind.

Assemblersprachen-Programme

Man kann Programme, die für den TI-89 Titanium / Voyage™ 200 Graphischer Rechner in Assembler geschrieben wurden, ausführen. Typischerweise laufen Assembler Programme viel schneller und bieten größere Sicherheit, als Programme die man mit Hilfe des vorhandenen Program Editors schreibt.

Wo Sie Assembler-Programme finden

Sowohl Assembler- als auch mit dem Editor erstellte Programme finden Sie auf der website von TI unter: education.ti.com.

Die dort verfügbaren Programme bieten zusätzliche Funktionen, die nicht im TI-89 Titanium / Voyage™ 200 enthalten sind. Auf der Texas Instruments Website finden Sie aktuellste Informationen.

Verwenden Sie, nachdem Sie ein Programm aus dem Internet auf Ihren Computer heruntergeladen haben, ein USB cable oder TI-GRAPH LINK™ Computer-Handheld-Verbindungskabel und die Software TI Connect, um das Programm an Ihren TI-89 Titanium / Voyage™ 200 zu senden.

Anweisungen zur Installation von Flash Apps finden Sie unter education.ti.com/guides.

Bemerkung zum TI-GRAPH LINK

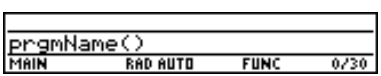
Wenn Sie ein TI-GRAPH LINK™ Computer-Handheld-Verbindungskabel und Software für den TI-89 oder TI-92 Plus besitzen, sollten Sie darauf achten, dass die Software TI-GRAPH LINK mit dem TI-89 Titanium oder Voyage™ 200 nicht kompatibel ist. Das Kabel funktioniert jedoch mit allen Geräten. Verwenden Sie auf Ihrem Computer die Software TI Connect.

Ein Computer-Handheld-Verbindungskabel oder ein Geräteverbindungskabel können Sie im TI Online Store unter education.ti.com/buy bestellen.

Ausführung eines Assemblersprachen-Programms

Nachdem Sie ein Assembler-Programm in Ihrem Gerät gespeichert haben, kann dies wie jedes andere Programm im Hauptbildschirm ausgeführt werden.

- Wenn für das Programm ein oder mehrere Argumente erforderlich sind, setzen Sie diese in ( ). Näheres zu den erforderlichen Argumenten finden Sie in der Dokumentation des Programms.



- Befindet sich das Programm nicht im aktuellen Verzeichnis, so muß dessen Pfadname angegeben werden.

Sie können ein Assembler-Programm als Unterprogramm von einem anderen Programm aus aufrufen, es löschen oder es wie jedes andere Programm verwenden.

Schnellzugriffstasten für die Ausführung eines Programms

Im Hauptbildschirm können Sie mit Schnellzugriffstasten bis zu neun benutzerdefinierte oder Assemblersprachen-Programme ausführen. Diese müssen allerdings gemäß folgender Tabelle benannt sein.

Drücken Sie im Hauptbildschirm:	Zur Ausführung des Programms namens (falls vorhanden):
♦ 1	kbdprgm1( )
⋮	⋮
♦ 9	kbdprgm9( )

Die Programme müssen im Verzeichnis `MAIN` gespeichert sein. Außerdem kann ein Programm, für welches ein Argument erforderlich ist, nicht mit Schnellzugriffstasten ausgeführt werden.

Wenn Sie Programme, deren Namen von den hier aufgeführten abweichen, mit Schnellzugriffstasten ausführen möchten, müssen Sie das vorhandene Programm entweder in **kbdprgm1( )** etc. kopieren oder umbenennen.

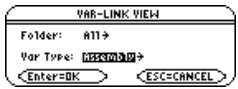
Ein Assembler-Programm kann nicht bearbeitet werden

Assembler-Programme lassen sich mit dem TI-89 Titanium / Voyage™ 200 nicht bearbeiten. Der eingebaute Programm-Editor öffnet keine Assembler-Programme.

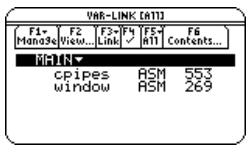
Anzeige einer Liste der Assembler-Programme

So werden die im Speicher befindlichen Assemblersprachen-Programme aufgelistet:

1. Rufen Sie den **VAR-LINK** Bildschirm auf (2nd [VAR-LINK]).
2. Drücken Sie F2 **View**.
3. Wählen Sie das entsprechende Verzeichnis (oder mit All alle Verzeichnisse), und stellen Sie **Var Type = Assembly** ein.



4. Drücken Sie ENTER, um die Liste der Assembler-Programme anzuzeigen.



Hinweis: Assembler-Programme gehören zum Datentyp ASM.

Informationen zum Schreiben eines Assembler-Programms

Es ist nicht Zweck dieses Handbuchs, die Informationen zu liefern, die ein unerfahrener Programmierer zum Schreiben eines Assembler-Programms benötigt. Wenn Sie allerdings Erfahrungen mit Assemblersprachen haben, konsultieren Sie bitte die Texas Instruments-Website (education.ti.com). Dort finden Sie genaue Angaben zum Zugreifen auf die Funktionen des TI-89 Titanium / Voyage™ 200.

Der TI-89 Titanium / Voyage™ 200 enthält auch einen **Exec**-Befehl zum Ausführen einer Zeichenfolge aus Motorola 68000 op-Codes. Diese Codes fungieren als eine Art Assembler-Programm. Weitere Angaben siehe TI-website.

Hinweis: Programme in Assembler können nur auf einem PC, nicht jedoch auf dem Taschenrechner geschrieben werden.

Achtung: Exec ermöglicht den Zugriff auf die gesamte Kapazität des Mikroprozessors. Fehler sind ohne weiteres möglich und führen zur Sperrung des Taschenrechners und zum Datenverlust. Es empfiehlt sich daher, vor der Arbeit mit dem Befehl **Exec** eine Sicherungskopie der Taschenrechnerdaten anzulegen.

Texteditor

Eine Texteditor-Sitzung starten

Wenn Sie den Texteditor starten, können Sie eine neue Sitzung beginnen, die aktuelle Sitzung fortsetzen (d. h. die zuletzt angezeigte Texteditor-Sitzung) oder eine frühere Sitzung öffnen.

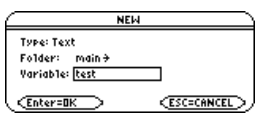
Eine neue Sitzung starten

1. Drücken Sie APPS, und wählen Sie **Text Editor**.

2. Wählen Sie **3:New**.

   Das Dialogfeld **NEW** wird angezeigt.

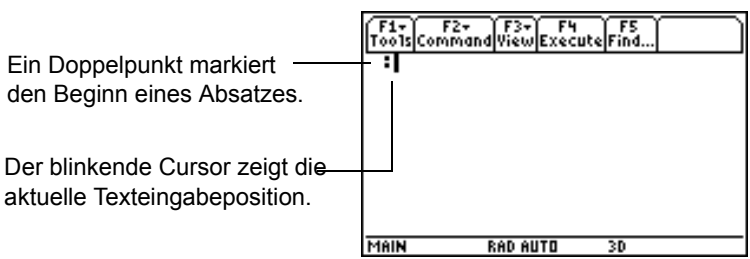
3. Geben Sie ein Verzeichnis und eine Textvariable zum Speichern der neuen Sitzung an.



Feld	Beschreibung
Type	Automatisch auf **Text** eingestellt; kann nicht geändert werden.

Feld	Beschreibung
Folder	Zeigt an, in welchem Verzeichnis die Textvariable gespeichert wird. Nähere Erläuterungen zu Verzeichnis finden Sie im Modul *Startbildschirm des Rechners*. Ein anderes Verzeichnis wählen Sie so: Drücken Sie ⊙, um das Menü der vorhandenen Verzeichnisse aufzurufen. Wählen Sie dort das gewünschte Verzeichnis.
Variable	Geben Sie einen Variablennamen ein. Wenn Sie eine bereits existierende Variable angeben, erhalten Sie nach Drücken von [ENTER] eine Fehlermeldung. Wenn Sie die Fehlermeldung durch Drücken von [ESC] oder [ENTER] bestätigen, wird erneut das Dialogfeld **NEW** angezeigt.

4. Drücken Sie [ENTER], um einen leeren Texteditor-Bildschirm aufzurufen (zum Abschluß der Eingabe in ein Eingabefeld, z. B. **Variable**, müssen Sie [ENTER] zweimal drücken).



Sie können nun wie in den folgenden Abschnitten erläutert mit dem Texteditor arbeiten.

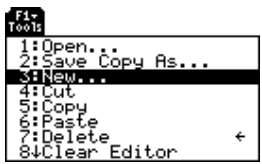
Hinweis: Die Sitzung wird bei der Eingabe automatisch gespeichert. Deshalb brauchen Sie die Sitzung nicht manuell zu speichern, bevor Sie den Texteditor beenden, eine neue Sitzung starten oder eine frühere öffnen.

Die aktuelle Sitzung fortsetzen

Sie können den Texteditor jederzeit beenden und zu einer anderen Anwendung wechseln. Um zu der vor Beenden des Texteditors angezeigten Sitzung zurückzukehren, drücken Sie APPS **8**, und wählen Sie **1:Current**.

Eine neue Sitzung innerhalb des Texteditors starten

Um die aktuelle Texteditor-Sitzung zu beenden und eine neue zu starten:



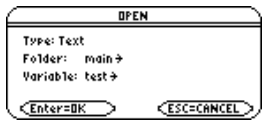
1. Drücken Sie F1, und wählen Sie **3:New**.
2. Geben Sie ein Verzeichnis und eine Textvariable für die neue Sitzung an.
3. Drücken Sie zweimal ENTER.

Eine frühere Sitzung öffnen

Sie können jederzeit eine frühere Texteditor-Sitzung öffnen.

1. Innerhalb des Texteditors: Drücken Sie F1, und wählen Sie **1:Open**.
   — oder —
   Aus einer beliebigen Anwendung heraus: Drücken Sie APPS **8**, und wählen Sie **2:Open**.

2. Wählen Sie das gewünschte Verzeichnis und die gewünschte Textvariable.
3. Drücken Sie ENTER.



Hinweis: Standardmäßig zeigt **Variable** die erste in der alphabetischen Liste vorhandene Textvariable an.

Einen Text kopieren

Sie können einen Text kopieren, die Kopie bearbeiten und so das Original unverändert beibehalten.

1. Lassen Sie sich den zu kopierenden Text anzeigen.
2. Drücken Sie F1, und wählen Sie **2:Save Copy As**.
3. Geben Sie ein Verzeichnis und eine Textvariable für die Kopie der Textes an.
4. Drücken Sie zweimal ENTER.

Hinweis zum Löschen von Texten

Da alle Texteditor-Sitzungen automatisch gespeichert werden, kann sich relativ viel Text ansammeln, und es wird u.U. viel Speicherplatz belegt.

Sie löschen einen Text, indem Sie den Bildschirm VAR-LINK (2nd [VAR-LINK]) zum Löschen der Textvariablen der Sitzung benutzen. Nähere Erläuterungen zu VAR-LINK finden Sie in *Speicher-und Variablen-Verwaltung*.

Text eingeben und bearbeiten

Nachdem Sie eine Texteditor-Sitzung gestartet haben, können Sie Text eingeben und bearbeiten. Grundsätzlich verwenden Sie dazu die gleichen Verfahren, die Sie bereits für Eingaben in die Eingabezeile des Hauptbildschirms kennengelernt haben.

Text eingeben

Wenn Sie eine neue Texteditor-Sitzung gestartet haben, wird ein leerer Bildschirm angezeigt. Wenn Sie eine frühere Sitzung öffnen oder zu der aktuellen Sitzung zurückkehren, wird der Text dieser Sitzung angezeigt.

Jeder Absatz beginnt mit einem Leerzeichen und einem Doppelpunkt.

Das führende Leerzeichen wird in Befehlsskripts und Laborberichten benutzt.

Blinkender Text-Cursor

Sie brauchen nicht an jedem Zeilenende ENTER zu drücken. Ist das Zeilenende erreicht, wird das nächste eingegebene Zeichen automatisch in die nächste Zeile gestellt. Drücken Sie ENTER nur, wenn Sie einen neuen Absatz beginnen möchten.

Wenn der Text den unteren Bildschirmrand erreicht, scrollen die oberen Zeilen über den oberen Rand aus dem Bildschirm.

Wenn Sie ein USB cable und die Software TI Connect™ mit dem TI-89 Titanium verwenden, bzw. ein TI-GRAPH LINK™ Rechner-Computer-Verbindungskabel und die Software TI Connect™ mit dem Voyage™ 200 Graphischer Rechner, dann können Sie mit Ihrer Computer-Tastatur eine Textdatei schreiben und diese anschließend an den TI-89 Titanium / Voyage™ 200 senden. Dies empfiehlt sich für die Eingabe umfangreicher Texte.

Informationen zum Bezug des benötigten Kabels oder von Updates der Software TI Connect™ finden Sie auf der TI Website unter education.ti.com, oder kontaktieren Sie Texas Instruments über TI-Cares™.

Hinweis:

- Verwenden Sie das Cursorfeld, um den Textcursor an die gewünschte Position (für Eingabe oder Ändern) zu bringen.

- Drücken Sie [2nd] ⬘ oder [2nd] ⬙, um den Text um je ein Bild aufwärts bzw. abwärts zu durchlaufen und [♦] ⬘ oder [♦] ⬙, wenn Sie den Cursor an den Anfang bzw. das Ende der Textsitzung bringen wollen.

Buchstaben eingeben

Aufgabe:	Auf dem TI-89:	Auf dem Voyage™ 200:
Einen einzelnen Kleinbuchstaben eingeben.	[alpha] und dann die Taste für den jeweiligen Buchstaben (die Statuszeile zeigt a an)	Buchstabentaste

Aufgabe:	Auf dem TI-89:	Auf dem Voyage™ 200:
Einen einzelnen Großbuchstaben eingeben.	[↑] und dann die Taste für den jeweiligen Buchstaben (die Statuszeile zeigt ⬟ an)	[↑] und danach die Buchstabentaste (in der Statuszeile ist ⬟ zu sehen)
Ein Leerzeichen eingeben.	[alpha] [␣] (Alpha-Funktion der Taste [(-)])	Leertaste
Die Kleinbuchstaben-Feststellfunktion einschalten.	[2nd] [a-lock] (Statuszeile zeigt a..z an)	(keine Tasten)
Die Großbuchstaben-Feststellfunktion einschalten.	[↑] [a-lock] (Statuszeile zeigt A..Z an)	[2nd] [CAPS]
Eine der beiden Feststellfunktionen ausschalten.	[alpha] (schaltet Großbuchstabentaste und Kleinbuchstabentaste aus)	[2nd] [CAPS] (schaltet Großbuchstabentaste aus)

Hinweis: Um auf dem TI-89, x, y, z oder t einzugeben, brauchen Sie weder [alpha] noch die Feststellfunktion. Für die Eingabe von X, Y, Z oder T müssen Sie jedoch [↑] oder die Großbuchstaben-Feststellfunktion verwenden. Beim Umschalten zwischen Anwendungen, wie z.B. zwischen Texteditor und Hauptbildschirm, ist die Feststellfunktion auf dem TI-89 stets ausgeschaltet.

Wenn eine der beiden Feststellfunktionen beim TI-89 eingeschaltet ist:

- Müssen Sie zum Eingeben eines Punktes, Kommas oder anderer Zeichen, welche die Primärfunktion einer Taste darstellen, die Feststellfunktion ausschalten.

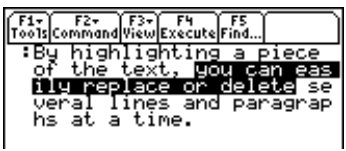
- Müssen Sie zum Eingeben eines Zweitfunktions-Zeichens wie z.B. 2nd [{] die Feststellfunktion nicht ausschalten. Nach der Eingabe dieses Zeichens bleibt die Feststellfunktion weiterhin aktiv.

Zeichen löschen

Zu löschendes Zeichen:	Drücken Sie:
Zeichen links des Cursors	← oder F1 **7**
Zeichen rechts des Cursors	♦ [DEL] (oder ♦ ←)
Alle Zeichen rechts des Cursors bis ans Absatzende	CLEAR
Den gesamten Text des Absatzes (unabhängig von der Position des Cursors)	CLEAR CLEAR

Hinweis: Wenn sich rechts des Cursors keine Zeichen befinden, löscht CLEAR den gesamten Absatz.

Markieren von Text

Zweck:	Vorgehensweise:
Text markieren	Bringen Sie den Cursor an den Anfang oder an das Ende des gewünschten Textstücks. Halten Sie ⇧ gedrückt, und drücken Sie: • ◀ oder ▶, um ein Zeichen links bzw. rechts des Cursors zu markieren. • ▼ oder ▲, um alle Zeichen bis zur Cursorposition in der vorhergehenden bzw. in der folgenden Zeile zu markieren. 

Hinweis: Sie heben die Markierung auf, ohne Text zu löschen oder zu ersetzen, indem Sie den Cursor bewegen.

Markierten Text ersetzen oder löschen

Zweck:	Vorgehensweise:
Markierten Text ersetzen	Geben Sie den neuen Text ein.
Markierten Text löschen	Drücken Sie ←.

Text ausschneiden, kopieren und einfügen

Mit Ausschneiden und Kopieren wird markierter Text in die Zwischenablage des TI-89 / Voyage™ 200 übertragen. Ausschneiden löscht dabei den Text an seinem Ursprungsort (und dient daher zum Verschieben von Text), während beim Kopieren der Text am Ursprungsort erhalten bleibt.

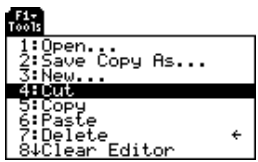
1. Markieren Sie den zu kopierenden bzw. zu verschiebenden Text.

2. Drücken Sie F1.

3. Wählen Sie den gewünschten Menüpunkt.

   - Zum Verschieben von Text: **4:Cut**.
     — oder —
   - Zum Kopieren von Text: **5:Copy**.

   Hinweis: Sie können die Schnelltasten:
   ◆ [CUT], ◆ [COPY], ◆ [PASTE]
   ◆ **X**, ◆ **C**, ◆ **V**
   zum Ausschneiden, Kopieren bzw. Einfügen verwenden, ohne das F1-Menü benutzen zu müssen.



4. Bewegen Sie den Textcursor an die Stelle, an der Sie den Text einfügen möchten.
5. Drücken Sie F1, und wählen Sie **6:Paste**.

Sie können dieses Verfahren zum Ausschneiden, Kopieren und Einfügen von Text verwenden:

- Innerhalb einer Text-Sitzung.

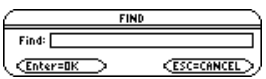
- Für den Textaustausch zwischen zwei Textsitzungen. Dazu schneiden Sie den Text in einer Sitzung aus bzw. kopieren ihn, öffnen anschließend eine andere Sitzung und fügen ihn dort ein.
- Aus einer Textsitzung in eine andere Anwendung. Sie können so z. B. Text in die Eingabezeile des Ausgangsbildschirms kopieren.

Text suchen

Gehen Sie innerhalb des Texteditors wie folgt vor:

1. Bringen Sie den Textcursor an eine beliebige Stelle vor dem zu suchenden Textstück. Die Suche beginnt immer an der aktuellen Cursorposition.

2. Drücken Sie F5.



3. Geben Sie den zu suchenden Text ein.

   Bei der Suche wird nicht nach Groß-/Kleinschreibung unterschieden. Beispiel: CASUS, casus und Casus ergeben das gleiche Suchresultat.

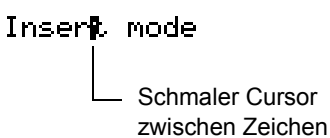
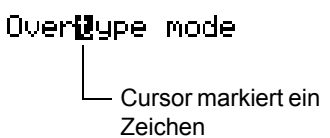
   Hinweis: Der zuletzt eingegebene Suchtext bleibt im Dialogfeld **FIND** erhalten. Sie können ihn überschreiben oder ändern.

4. Drücken Sie zweimal ENTER.

Wird der Suchtext:	Verhält sich der Cursor so:
Gefunden	Rückt an den Anfang des gefundenen Textteils.
Nicht gefunden	Bleibt an der ursprünglichen Stelle.

Zeichen überschreiben oder einfügen

Standardmäßig befindet sich der TI-89 / Voyage™ 200 im Einfügemodus. Drücken von 2nd [INS] schaltet zwischen dem Einfüge- und dem Überschreibmodus hin und her.

Befindet sich der TI-89 / Voyage™ 200 im:	Bewirkt die Eingabe des nächsten Zeichens:
	Das Zeichen wird am Cursor eingefügt.
	Das markierte Zeichen wird ersetzt.

Hinweis: An der Form des Cursors erkennen Sie, ob der Einfüge- oder der Überschreibmodus aktiv ist.

Den gesamten Text löschen

Um den gesamten Text der Sitzung zu löschen und wieder einen leeren Bildschirm zu erhalten, drücken Sie F1, und wählen Sie dann **8:Clear Editor**.

Sonderzeichen eingeben

Mit Hilfe des Menüs CHAR können Sie Sonderzeichen aus einer Liste auswählen. Einige gängige Zeichen können Sie auch mit der Tastatur eingeben. Sie können eine Tastaturbelegungsübersicht aufrufen, die die Zeichen und die zugeordneten Tasten anzeigt.

Das Menü CHAR benutzen

1. Drücken Sie 2nd [CHAR].
2. Wählen Sie die gewünschte Kategorie.

   In einem Untermenü sehen Sie die in der jeweiligen Kategorie verfügbaren Zeichen.
3. Wählen Sie ein Zeichen aus. Unter Umständen müssen Sie das Menü scrollen, um alle Zeichen zu sehen.

   Hinweis: Für akzentuierte Buchstaben wählen Sie International. Gebräuchliche internationale Zeichen sind außerdem im vorgegebenen Benutzermenü (2nd [CUSTOM]) verfügbar.

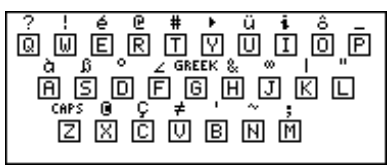
↓ zeigt an, dass das Menü weitere Einträge enthält.

Die Tastaturbelegung anzeigen

Die Tastaturbelegungsübersicht zeigt Tastenkombinationen für die Eingabe gewisser Sonderzeichen und griechischer Buchstaben. Außerdem werden Schnelltastenkombinationen für andere Taschenrechnerfunktionen angezeigt.

In der Tastaturbelegung werden nicht alle verfügbaren Tastenkombinationen angezeigt. Eine vollständige Liste ist innen auf der Vorder- und der Rückseite des Bucheinbands zu finden.

Um Tastenkombinationen zu nutzen, drücken Sie zuerst die Taste [2nd]. Einige wenige Sonderzeichen sind auf der Tastatur angegeben, die meisten aber nicht.

Auf dem TI-89 Titanium:	**Auf dem Voyage™ 200:**
Rufen Sie die Tastaturbelegung mit [♦] [EE] auf. Mit [ESC] wird die Tastaturbelegung ausgeblendet.	Rufen Sie die Tastaturbelegung mit [♦] [KEY] auf. Mit [ESC] wird die Tastaturbelegung ausgeblendet.
	
TI-89 Titanium Tastaturbelegung	Voyage™ 200 Tastaturbelegung
Um auf TI-89 Titanium Schnelltastenkombinationen zuzugreifen, drücken Sie zunächst [♦].	

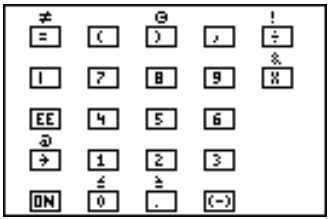
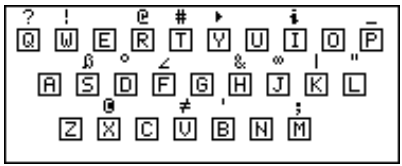
Tastenkombinationen der Tastaturbelegung des TI-89:	Tastenkombinationen der Tastaturbelegung des Voyage™ 200:
GREEK ([♦] [(]) — Hiermit erfolgt der Zugriff auf den griechischen Zeichensatz (eine Beschreibung erfolgt weiter hinten in diesem Absatz). SYSDATA ([♦] [,]) — Kopiert die aktuellen Graphik-Koordinaten in die Systemvariable sysdata. FMT ([♦] [|]) — ruft das Dialogfeld FORMATS auf. KBDPRGM1 – 9 ([♦] 1 - [♦] 9) — Haben Sie benutzerdefinierte oder Assemblersprachen-Programme mit kbdprgm1() bis kbdprgm9() benannt, dann führen diese Tastenkombinationen das entsprechende Programm aus.	GREEK ([2nd] **G**) — Hiermit erfolgt der Zugriff auf den griechischen Zeichensatz (eine Beschreibung erfolgt weiter hinten in diesem Absatz). CAPS ([2nd] [CAPS]) — Schaltet Großbuchstaben ein bzw. aus. Accent marks — (é, ü, ô, à, ç, und ~) werden dem nächsten eingegebenen Buchstaben zugeordnet (eine Beschreibung erfolgt weiter hinten in diesem Abschnitt).

Tastenkombinationen der Tastaturbelegung des TI-89:	Tastenkombinationen der Tastaturbelegung des Voyage™ 200:
OFF ([♦] [OFF]) — Ähnlich wie [2nd] [OFF], jedoch mit folgenden Unterschieden: • Sie können [♦] [OFF] drücken, wenn eine Fehlermeldung erscheint • Wird der TI-89 Titanium wieder eingeschaltet, erscheint wieder der Bildschirm, der beim Ausschalten aktiv war. HOMEDATA ([♦] [(-)]) — Kopiert die aktuellen Graphik-Koordinaten in den Protokoll-Bereich des Hauptbildschirms.	

Eingabe von Sonderzeichen über die Tastatur

Hinweis: Um das Auffinden der notwendigen Tasten zu erleichtern, sind in diesen Übersichtsdarstellungen für die Tastaturbelegung nur die Sonderzeichen angegeben.

Auf dem TI-89 Titanium:	Auf dem Voyage™ 200:
Drücken Sie [♦] und dann die Taste, der das gewünschte Symbol zugeordnet ist. Beispiel: [♦] [×] (anfügen) ergibt &.	Drücken Sie [2nd] und dann die Taste, der das gewünschte Symbol zugeordnet ist. Beispiel: [2nd] H ergibt &.

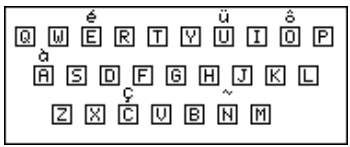
Auf dem TI-89 Titanium:	Auf dem Voyage™ 200:
	
Ob Alpha Lock ein- oder ausgeschaltet ist, besitzt keinen Einfluß auf die Eingabe dieser Sonderzeichen.	Ob Caps Lock ein- oder ausgeschaltet ist, besitzt keinen Einfluß auf die Eingabe dieser Sonderzeichen.

Hinweis: Als Erleichterung beim Auffinden der möglichen Tasten sind in diesen Tastaturbelegungen nur die Sondersymbole dargestellt.

Akzentzeichen mit der Voyage™ 200 Tastatur eingeben

Wenn Sie eine Akzentzeichentaste drücken, wird nicht sofort ein Buchstabe mit Akzent/Umlaut angezeigt, sondern der nächste Buchstabe, den Sie eingeben, wird mit dem Akzent/Umlaut versehen.

1. Drücken Sie 2nd und dann die Taste, der das Akzent-/Umlautzeichen zugeordnet ist.



Hinweis: Zur Vereinfachung sind hier nur die Akzent-/Umlautzeichen abgebildet.

2. Drücken Sie die Taste des Buchstabens, den Sie mit dem Akzent/Umlaut versehen möchten.
   - Sie können sowohl Klein- als auch Großbuchstaben mit einem Akzent/Umlaut versehen.
   - Sie können nur Zeichen mit einem Akzent/Umlaut versehen, für die dies zulässig ist.

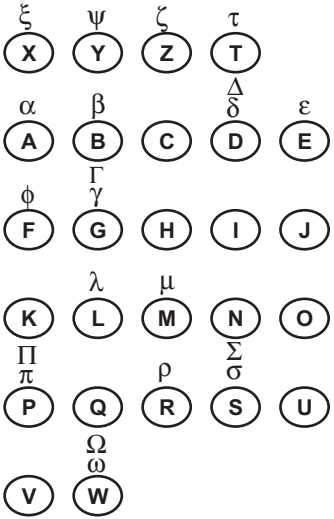
Umlaut/Akzent	Zulässig für folgende Buchstaben (Klein- und Großbuchstaben)	Beispiele
´	A, E, I, O, U, Y	é, É
¨	A, E, I, O, U, y (jedoch nicht Y)	ü, Ü
ˆ	A, E, I, O, U	ô, Ô
`	A, E, I, O, U	à, À
ç	C	ç, Ç
~	A, O, N	ñ, Ñ

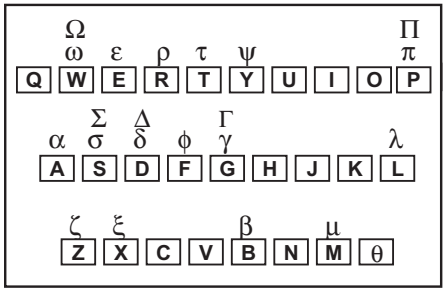
Griechische Buchstaben über die Tastatur eingeben

Drücken Sie die Tasten zum Aufrufen des griechischen Zeichensatzes. Wählen Sie dann den entsprechenden Buchstaben auf der Tastatur, um ein griechisches Zeichen einzugeben.

Auf dem TI-89:	Auf dem Voyage™ 200:
Drücken Sie ♦ (, um auf den griechischen Zeichensatz zuzugreifen.	Drücken Sie 2nd **G**, um auf den griechischen Zeichensatz zuzugreifen.

Auf dem TI-89:	Auf dem Voyage™ 200:





Hinweis: Wenn Sie eine Tastenkombination drücken, durch die kein griechischer Buchstabe aufgerufen wird, wird der normale Buchstabe der betreffenden Taste aufgerufen. Ihr Handheld zeigt keine Tastaturbelegungsübersicht für griechische Buchstaben an. Die abgebildetet Übersicht dient nur zu Referenzzwecken.

Mehrere Tasten bieten Zugang zu griechischen Groß- und Kleinbuchstaben.

Auf dem TI-89:	**Auf dem Voyage™ 200:**
Drücken Sie ♦ (, um auf den griechischen Zeichensatz zuzugreifen. Mit ♦ (alpha + Buchstabe erfolgt Zugriff auf griechische Kleinbuchstaben. Beispiel: ♦ (alpha [W] ergibt ω Mit ♦ (⬆ + Buchstabe erfolgt Zugriff auf griechische Großbuchstaben. Beispiel: ♦ (⬆ [W] ergibt Ω	Drücken Sie 2nd **G**, um auf den griechischen Zeichensatz zuzugreifen. Mit 2nd **G** + Buchstabe erfolgt Zugriff auf griechische Kleinbuchstaben. Beispiel: 2nd **G W** ergibt ω Mit 2nd **G** ⬆ + Buchstabe erfolgt Zugriff auf griechische Großbuchstaben. Beispiel: 2nd **G** ⬆ **W** ergibt Ω

Welche Tasten auf dem TI-89 zu drücken sind, hängt davon ab, ob die Feststelltaste aktiviert ist; beispielsweise:

Auf dem TI-89, wenn:	**Dann:**
Keine Feststellfunktion eingeschaltet ist.	♦ (**X** oder ♦ (alpha **X** ergibt ξ. (alpha ist für X, Y, Z, oder T nicht erforderlich.verwendet.) ♦ (alpha **W** ergibtt ω. ♦ (⬆ **W** ergibtt Ω (⬆ wird für Großbuchstaben verwendet.)
Die Kleinbuchstaben-Feststellfunktion (2nd [a-lock]) eingeschaltet ist.	♦ (**X** ergibt ξ. ♦ (**W** ergibt ω. ♦ (⬆ **W** ergibt Ω.

Auf dem TI-89, wenn:	Dann:
Die Großbuchstaben-Feststellfunktion ([↑] [a-lock]) eingeschaltet ist.	[♦] [(] **X** ergibt ξ. [♦] [(] **W** ergibt Ω. [♦] [(] [↑] **W** ergibt Ω.

Wichtig: Wenn Sie bei aktivierter Feststellfunktion [alpha] auf dem TI-89 drücken, um auf einen griechischen Buchstaben zuzugreifen, wird die Feststellfunktion ausgeschaltet.

Aufstellung aller Sonderzeichen

Eine Aufstellung aller Sonderzeichen finden Sie im Modul *Technische Referenz*.

Ein Befehlsskript eingeben und ausführen

Sie können mit dem Texteditor Befehlsskripte erstellen und diese jederzeit vom Hauptbildschirm aus ausführen. Auf diese Weise können Sie interaktive Beispielskripts erstellen, in denen Sie eine Folge von Befehlen vordefinieren, und diese einzeln ausführen.

Eine Befehlsmarke einfügen

Gehen Sie innerhalb des Texteditors wie folgt vor:

1. Bringen Sie den Cursor in die Zeile, die Sie als Befehlszeile ausweisen möchten.

2. Drücken Sie F2, um das Command-Menü zu öffnen.

3. Wählen Sie **1:Command**.

   Es wird nun ein“**C**” am Anfang der Textzeile angezeigt (links des Doppelpunkts).

   Hinweis: Es wird damit keine neue Zeile für einen Befehl eingefügt, sondern nur eine vorhandene Zeile als Befehlszeile markiert.

4. Geben Sie einen Befehl genau wie im Hauptbildschirm ein.

   Die Zeile darf nur den Befehl enthalten, keinen weiteren Text.

   Hinweis: Sie können eine Zeile vor oder nach Eingabe des Befehls als Befehlszeile markieren.

   Sie können in eine einzige Zeile mehrere Befehle eingeben, wenn Sie die Befehle jeweils durch einen Doppelpunkt trennen.

Eine Befehlsmarke löschen

Damit wird nur die Markierung “C” gelöscht, nicht jedoch der Befehlstext selbst.

1. Bringen Sie den Cursor an eine beliebige Stelle der markierten Zeile.

2. Drücken Sie F2, und wählen Sie **4:Clear command**.

Einen Befehl ausführen

Um einen Befehl ausführen zu können, muß die Zeile mit einem "**C**" markiert sein. Wenn Sie eine nicht mit einem "**C**" markierte Zeile auszuführen versuchen, wird dies ignoriert.

1. Bringen Sie den Cursor an eine beliebige Stelle der Befehlszeile.
2. Drücken Sie F4.

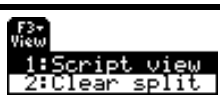
Der Befehl wird in die Eingabezeile des Hauptildschirms kopiert und ausgeführt. Während der Ausführung wird kurzzeitig der Hauptbildschirm angezeigt und anschließend wieder der Texteditor.

Nach der Ausführung rückt der Cursor in die nächste Skriptzeile vor, so daß Sie weitere Befehle ausführen können.

Hinweis: Um das Ergebnis auf dem Hauptbildschirm einzusehen, drücken Sie HOME; ◆ [CALC HOME] oder benutzen Sie eine geteilte Anzeige.

Den Texteditor-Bildschirm/Hauptbildschirm teilen

Durch Teilen des Bildschirms können Sie gleichzeitig Ihr Befehlsskript und das Ergebnis der Befehlsausführung einsehen.

Zweck:	Drücken Sie:	
Bildschirm teilen	F3, und wählen Sie **1:Script view**.	

Zweck:	Drücken Sie:
Rückkehr zur ganzseitigen Texteditor-Anzeige	F3, und wählen Sie **2:Clear split**.

Sie können einen geteilten Bildschirm auch manuell mit MODE einrichten. Mit F3 richten Sie eine geteilte Anzeige von Texteditor/ Hauptbildschirm jedoch auf viel einfachere Weise als mit MODE ein.

- Die aktive Anwendung wird durch eine fette Umrandung kenntlich gemacht. Standardmäßig ist der Texteditor die aktive Anwendung.
- Zum Umschalten zwischen Texteditor und Hauptbildschirm drücken Sie 2nd [⇔] (zweite Funktion von APPS).

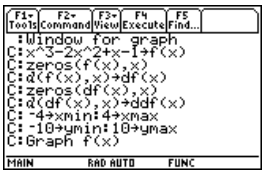
Ein Skript aus Hauptbildschirmeinträgen erstellen

Vom Hauptbildschirm aus können Sie alle im Protokoll-Bereich befindlichen Einträge in einer Textvariable abspeichern. Die Einträge werden automatisch im Skriptformat gespeichert, so daß Sie die Textvariable im Texteditor öffnen und die Einträge als Befehle ausführen können.

Nähere Erläuterungen hierzu finden Sie in “Eingaben in Hauptbildschirm als Texteditor-Skript speichern” in *Weitere Funktionen des Hauptbildschirms*.

Beispiel

1. Geben Sie das Skript ein. Drücken Sie F2, und wählen Sie **1:Command**, um die Befehlszeilenmarken zu setzen.



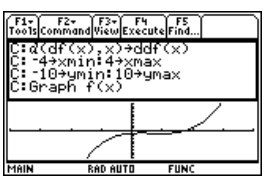
2. Drücken Sie F3, und wählen Sie **1:Script view**.

3. Setzen Sie den Cursor in die erste Befehlszeile. Drücken Sie anschließend F4, um den Befehl auszuführen.

   Hinweis: Manche Befehle benötigen längere Zeit zur Ausführung. Warten Sie, bis die Anzeige `Busy` verschwindet, bevor Sie erneut F4 drücken.

4. Drücken Sie wiederholt F4, um die einzelnen Befehle auszuführen. Führen Sie den letzten Befehl vor dem Befehl Graph aus, aber nicht Graph selbst.

5. Führen Sie den Befehl Graph aus.



   Hinweis: In diesem Beispiel zeigt der Befehl Graph den Graphik-Bildschirm statt des Hauptbildschirms an.

6. Drücken Sie F3, und wählen Sie **2:Clear split**, um zur ganzseitigen Anzeige des Texteditors zurückzukehren.

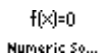
Gleichungslöser

Öffnen des Gleichungslösers und Eingabe einer Gleichung

Zeigen Sie zunächst den numerischer Gleichungslöser an, und geben Sie dann die zu lösende Gleichung ein.

Öffnen des numerischen Gleichungslösers

Drücken Sie APPS, und wählen Sie **Numeric Solver**.



Der Bildschirm des numerischen Gleichungslösers zeigt, falls vorhanden, die zuletzt eingegebene Gleichung an.

Eingabe einer Gleichung

Geben Sie Ihre Gleichung in die Zeile **eqn:** ein.

Sie können:	Zum Beispiel:
Eine Gleichung direkt eintippen.	**a=(m2−m1)/(m2+m1)∗g** **a+b=c+sin(d)**

Sie können:	Zum Beispiel:
Einen Bezug zu einer an anderer Stelle definierten Funktion oder Gleichung herstellen. **HInweis:** • Verwenden Sie keine Systemfunktionsnamen (wie **y1(x)** oder **r1(θ)**) als einfache Variablen (**y1** oder **r1**). • Vorsicht mit impliziter Multiplikation. **a(m2+m1)** wird z.B. als Funktionsreferenz behandelt, nicht als **a∗ (m2+m1)**.	**y1(x)** sei entweder im: • Y= Editor: **y1(x)=1.25x∗cos(x)** – oder – • im Hauptbildschirm definiert: **Define y1(x)=1.25x∗cos(x)** Sie würden dann folgendes in die numerische Auflösungsfunktion eingeben: **y1(x)=0** oder **y1(t)=0**, etc. Das Argument muß nicht mit dem zur Definition der Funktion oder Gleichung verwendeten übereinstimmen.
Geben Sie einen Term ohne das Zeichen = ein. **Hinweis:** Wenn Sie die Variablen definieren, können Sie entweder **exp** definieren oder danach auflösen.	**e+f−ln(g)** Wenn Sie ENTER drücken, wird der Term mit einer Systemvariablen namens exp gleichgesetzt: exp=e+f−ln(g)

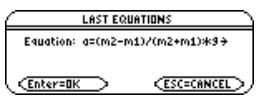
Sie können:	Zum Beispiel:
Rufen Sie eine zuvor eingegebene Gleichung auf, oder öffnen Sie eine gespeicherte Gleichung. **Hinweis:** Wenn Sie ENTER drücken, wird die aktuelle Gleichung automatisch in die Systemvariable **eqn** gespeichert.	Siehe entsprechenden Abschnitt in diesem Einheit.

Aufrufen zuvor eingegebener Gleichungen

Die zuletzt eingegebenen Gleichungen (bis zu 11 mit der Standardeinstellung) werden im Speicher aufbewahrt. So rufen Sie eine dieser Gleichungen auf:

1. Drücken Sie im Bildschirm des numerischen Gleichungslösers F5.

   Die zuletzt eingegebenen Gleichungen werden in einem Dialogfeld angezeigt.



2. Wählen Sie eine Gleichung.

   - Drücken Sie zum Wählen der angezeigten Gleichung ENTER.
   - Um eine andere Gleichung zu wählen, drücken Sie ▶. Eine Liste wird angezeigt, aus der Sie dann die gewünschte Gleichung wählen können.

   HInweis: Sie können bestimmen, wie viele Gleichungen gespeichert werden. Drücken Sie im Gleichungslöser F1, und wählen Sie **9:Format** (bzw. drücken Sie folgende Tasten: 🖩 ♦ I; ⌨ ♦ **F**). Wählen Sie dann eine Zahl von 1 bis 11.

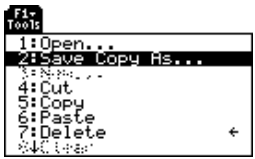
3. Drücken Sie ENTER.

Gleichungen werden nur einmal aufgelistet. Auch wenn Sie dieselbe Gleichung fünfmal eingeben, wird sie nur einmal aufgeführt.

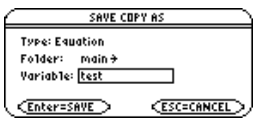
Speichern von Gleichungen zur späteren Verwendung

Da die Anzahl an Gleichungen, die Sie mit F5 **Eqns** aufrufen können, beschränkt ist, wird eine bestimmte Gleichung wahrscheinlich nicht unbegrenzt lang aufbewahrt.

Um die aktuelle Gleichung zur künftigen Verwendung zu sichern, speichern Sie sie in eine Variable.



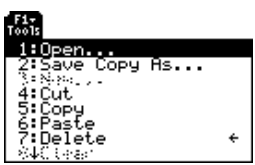
1. Drücken Sie im Bildschirm der numerischen Auflösungsfunktion F1, und wählen Sie **2:Save Copy As**.
2. Geben Sie ein Verzeichnis und einen Variablennamen für die Gleichung an.
3. Drücken Sie zweimal ENTER.



Hinweis: Eine Gleichungsvariable gehört, wie in den Bildschirmen MEMORY und VAR-LINK angezeigt, zum Datentyp EXPR.

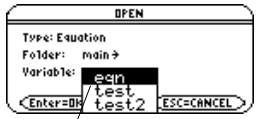
Öffnen einer gespeicherten Gleichung

So öffnen Sie eine zuvor gespeicherte Gleichungsvariable:



1. Drücken Sie im Bildschirm des numerischen Gleichungslösers F1, und wählen Sie **1:Open**.

2. Wählen Sie das betreffende Verzeichnis und die gewünschte Gleichungsvariable.
3. Drücken Sie ENTER.



Die Variable eqn enthält die aktuelle Gleichung; in der Liste wird sie stets alphabetisch aufgeführt.

Die bekannten Variablen definieren

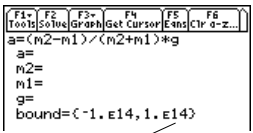
Geben Sie, nachdem Sie eine Gleichung in den numerischen Gleichungslöser eingetippt haben, die entsprechenden Werte für alle außer der unbekannten Variablen ein.

Definieren der Variablenliste

Nachdem Sie Ihre Gleichung in die Zeile eqn: eingegeben haben, drücken Sie ENTER oder ⊙.

Die Variablen werden in der Reihenfolge, in welcher sie in der Gleichung auftreten, auf dem Bildschirm aufgelistet. Wurde eine Variable bereits definiert, so wird ihr Wert angezeigt. Diese Variablenwerte können bearbeitet werden.

Hinweis: Der Wert einer gesperrten oder archivierten Variablen kann nicht bearbeitet werden.

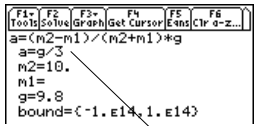


Die Lösung muß innerhalb der angegebenen Grenzen liegen. Diese können geändert werden.

Geben Sie für alle Variablen, außer für diejenige, nach der die Gleichung aufgelöst werden soll, eine Zahl oder einen Term ein.

Hinweise und häufige Fehler

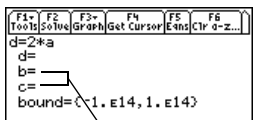
- Wenn Sie eine Variable:
  - Bezüglich einer anderen Variablen der Gleichung definieren, so muß diese zuerst definiert werden.
  - Bezüglich einer anderen Variablen definieren, die nicht in der Gleichung auftritt, dann muß diese Variable bereits einen Wert besitzen; sie kann nicht undefiniert bleiben.
  - Als Term definieren, so wird sie ausgewertet, sobald Sie den Cursor von der Zeile weg bewegen. Der Term muß als reelle Zahl auswertbar sein.



Da a bezüglich g definiert ist, muß g vor a definiert werden. Wenn Sie den Cursor in eine andere Zeile setzen, wird g/3 ausgewertet.

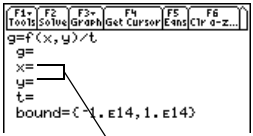
- Enthält die Gleichung eine bereits bezüglich anderer Variablen definierte Variable, so werden diese anderen Variablen aufgelistet.

  Hinweis: Wenn Sie einer Variablen im numerischen Gleichungslöser einen Wert zuteilen, ist diese Variable global definiert. Sie bleibt auch bestehen, wenn Sie den Gleichungslöser verlassen.



Wurde a zuvor als b+c→a definiert, dann werden b und c anstelle von a aufgelistet.

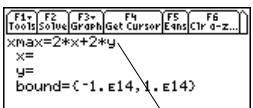
- Beziehen Sie sich auf eine zuvor definierte Funktion, so werden nicht die zum Definieren der Funktion, sondern alle als Argumente im Funktionsaufruf verwendeten Variablen aufgelistet.



Wurde f(a,b) zuvor als √(a^2+b^2) definiert und Ihre Gleichung enthält f(x,y), dann werden nicht a und b, sondern x und y aufgelistet.

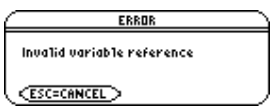
- Enthält die Gleichung eine Systemvariable (**xmin**, **xmax**, etc.), wird diese Variable nicht aufgelistet. Der Gleichungslöser verwendet den vorhandenen Wert der Systemvariablen.

  Hinweis: Sie können nach keiner anderen Systemvariablen als exp auflösen. Außerdem kann F3 nicht zur graphischen Darstellung verwendet werden, wenn die Gleichung eine Systemvariable enthält.



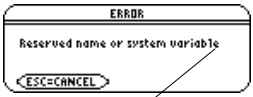
Im Standard-Ansichtfenster xmax=10.

- Obgleich Sie eine Systemvariable in der Gleichung verwenden können, tritt bei Verwendung von F3 zur graphischen Darstellung der Lösung ein Fehler auf.



- Beim Auftreten des nebenstehenden Fehlers, löschen Sie den eingegebenen Variablenwert. Bearbeiten Sie dann die Gleichung, so dass eine andere Variable verwendet wird.

  Hinweis: Dieser Fehler tritt auf, wenn Sie einen reservierten Namen falsch verwenden oder sich auf eine undefinierte Systemfunktion als einfache Variable ohne Klammern beziehen.



Beispiel: y1(x) ist undefiniert, und Sie verwenden y1.

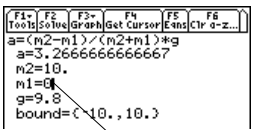
Bearbeiten der Gleichung

Drücken Sie im numerischen Gleichungslöser auf ⊙, bis sich der Cursor auf der Gleichung befindet. Der Bildschirm wird automatisch umgeschaltet und zeigt lediglich die Zeile **eqn:** an. Nehmen Sie die erforderlichen Änderungen vor, und drücken Sie dann ENTER oder ⊙, um zur Variablenliste zurückzukehren.

Angabe einer ersten Näherung bzw. von Grenzen (optional)

Um die Lösung schneller zu erhalten, oder um eine bestimmte Lösung zu finden (bei mehreren Lösungen), können Sie folgendermaßen vorgehen:

- Geben Sie eine erste Näherung für die unbekannte Variable ein. Die Näherung muß innerhalb der angegebenen Grenzen liegen.
- Geben Sie untere und obere Grenzen in der Nähe der Lösung ein.



Die erste Näherung muß innerhalb der Grenzen liegen.

Als Grenzen können Sie auch Variablen oder Terme eingeben, die sich nach passenden Werten auswerten lassen (**bound={lower,upper}**) oder aber eine gültige Listenvariable, die eine zweielementige Liste enthält (**bound=list**). Die Grenzen müssen aus zwei Fließkommaelementen bestehen, wobei das erste kleiner oder gleich dem zweiten Element sein muß.

HInweis: Graphische Wahl einer ersten Näherung.

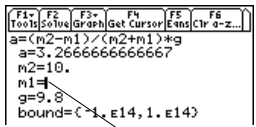
Nach der unbekannten Variablen auflösen

Nachdem Sie eine Gleichung im numerischen Gleichungslöser eingegeben und die Werte für die bekannten Variablen festgesetzt haben, können Sie nun nach der unbekannten Variablen auflösen.

Ermittlung der Lösung

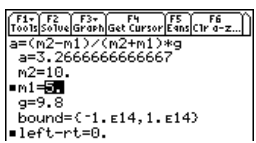
Nachdem alle bekannten Variablen definiert sind:

1. Setzen Sie den Cursor auf die unbekannte Variable.



Setzen Sie den Cursor auf die Variable, nach welcher aufgelöst werden soll.

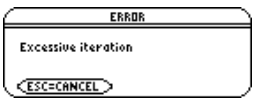
2. Drücken Sie F2 **Solve**.
3. ▪ markiert die Lösung und **left**–**rt**. Die Markierung ▪ wird entfernt, wenn Sie einen Wert bearbeiten, den Cursor auf die Gleichung setzen oder den Gleichungslöser verlassen.



HInweis: Drücken Sie zum Abbrechen einer Berechnung ON. Die unbekannte Variable zeigt den Wert an, der zum Zeitpunkt der Unterbrechung geprüft wurde.

Die linke und die rechte Seite der Gleichung werden unter Verwendung der Lösung und der eingegebenen Werte getrennt ausgewertet. **left**–**rt** zeigt die Differenz und somit die Genauigkeit der Lösung an. Je kleiner der Wert, desto genauer ist die Lösung. Bei einer exakten Lösung ist **left**–**rt =0**.

Wenn Sie:	Gehen Sie wie folgt vor:
Nach anderen Werten auflösen möchten	Ändern Sie die Gleichung oder die Variablenwerte.

Wenn Sie:	Gehen Sie wie folgt vor:
Für eine Gleichung mit mehreren Lösungen eine andere Lösung finden möchten	Geben Sie eine andere Näherung bzw. einen neuen Satz Randwerte nahe der gewünschten Lösung ein.
Folgende Meldung erhalten:  	Drücken Sie ESC. Die unbekannte Variable zeigt den Wert an, der geprüft wurde, als der Fehler auftrat. • Möglicherweise ist der Wert von **left**–**rt** als Ergebnis klein genug. • Falls nicht, geben Sie andere Randwerte ein.

Hinweis: Die Gleichung wird mittels eines iterativen Verfahrens aufgelöst. Kann das Verfahren nicht gegen eine Lösung konvergieren, so tritt dieser Fehler auf.

Die Lösung graphisch darstellen

Nachdem die bekannten Variablen definiert wurden, können Sie die Lösung einer Gleichung sowohl vor als auch nach dem Lösen nach der unbekannten Variablen jederzeit graphisch darstellen. Durch die graphische Darstellung der Lösungen sehen Sie, wie viele Lösungen vorhanden sind, und Sie haben die Möglichkeit, mit dem Cursor passende erste Näherung bzw. Grenzen zu wählen.

Anzeige des Graphen

Belassen Sie den Cursor im numerischen Gleichungslöser auf der unbekannten Variablen. Drücken Sie F3, und wählen Sie:

1:Graph View
– oder –
3:ZoomStd
– oder–
4:ZoomFit

Graph View verwendet die aktuellen Fenstervariablenwerte.

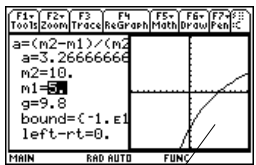
Näheres zu **ZoomStd** und **ZoomFit** finden Sie in *Grundlagen der graphischen Darstellung von Funktionen*.

Der Graph wird im geteilten Bildschirm angezeigt, in welchem:

- Die unbekannte Variable auf der x-Achse und
- **left**–**rt** auf der y-Achse geplottet sind.

Lösungen für die Gleichung liegen bei **left**–**rt=0** vor, wo der Graph die x-Achse schneidet.

Hinweis: Näheres finden Sie in *Geteilte Bildschirme*.



Es werden die aktuellen Grafikformat-Einstellungen verwendet.

Sie können den Graphen mit Hilfe des frei beweglichen Cursors, der Tracing- oder Zooming-Funktion etc. untersuchen. Eine Erklärung hierzu finden Sie in *Grundlagen der graphischen Darstellung von Funktionen.*

Wie sich der Graph auf verschiedene Einstellungen auswirkt

Wenn Sie einen Graphen im numerischen Gleichungslöser anzeigen,

- So werden folgende Modi automatisch auf diese Einstellungen gesetzt:

Modus	Einstellung
Graph	FUNCTION Etwaige im Y= Editor gewählte Funktionen werden nicht graphisch dargestellt.
Split Screen	LEFT-RIGHT
Number of Graphs	1

  Hinweis: Haben Sie zuvor andere Moduseinstellungen verwendet, müssen Sie diese manuell neu wählen.

- Die Auswahl aller Statistik-Plots wird aufgehoben.
- Es kann vorkommen, dass der Graphikbildschirm auch nach dem Verlassen des numerischen Gleichungslösers weiterhin die Lösung der Gleichung anzeigt und etwaige gewählte Y=-Funktionen ignoriert. Öffnen Sie in diesem Fall den Y= Editor, und kehren Sie dann zum Graphikbildschirm zurück. Der Graph wird auch durch Wechseln des Graph-Modus oder die Verwendung von **ClrGraph** im Hauptbildschirm (F4 **5**) oder die Ausführung eines Programms zurückgesetzt.

Wahl einer neuen Anfangsnäherung aus der Graphik

So wählen Sie mit Hilfe des Graphikcursors eine Anfangsnäherung:

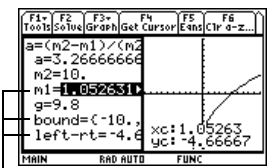
1. Setzen Sie den Cursor (den Freihand- oder Trace-Cursor) auf den Punkt, der als neue Anfangsnäherung verwendet werden soll.

2. Aktivieren Sie den Bildschirm des numerischen Gleichungslösers mit [2nd] [⇔].

3. Vergewissern Sie sich, dass der Cursor auf der unbekannten Variablen sitzt, und drücken Sie [F4].

   Hinweis: Die Cursor-Koordinate xc ist der Wert der unbekannten Variablen und yc der Wert von **left**–**rt**.

4. Drücken Sie [F2], um die Gleichung erneut zu lösen.



[F4] setzt den xc-Wert des Grafikcursors als Ausgangsschätzung und den yc-Wert als left–rt. Die Werte xmin und xmax des Graphen werden als Grenzen gesetzt.

Rückkehr zur Vollbildanzeige

Ausgangsposition ist der geteilte Bildschirm:

- Um den Bildschirm des numerischen Gleichungslösers im Vollbildmodus anzuzeigen, aktivieren Sie den Bildschirm mit [2nd] [⇔], drücken [F3] und wählen dann **2:Clear Graph View**.
  – oder –

- Drücken Sie 2nd [QUIT] zweimal, um den Hauptbildschirm anzuzeigen.

Löschen der Variablen vor dem Verlassen des Gleichungslösers

Wenn Sie eine Gleichung auswerten, bleiben deren Variablen auch nach Verlassen des numerischen Gleichungslösers weiterhin bestehen. Enthält die Gleichung aus einem Zeichen bestehende Variablen, so können deren Werte spätere symbolische Berechnungen ungewollt beeinflussen. Sie sollten deshalb vor dem Verlassen der numerischen Auflösungsfunktion:

1. 2nd [F6]
   F6
   drücken, um alle aus einem Zeichen bestehenden Variablen aus dem aktuellen Verzeichnis zu löschen.
2. Zur Bestätigung des Vorgangs ENTER drücken.

   Die Anzeige kehrt zur Zeile **eqn:** des Gleichungslösers zurück.

Hinweis: Mit folgenden Tasten können Sie aus einem Zeichen bestehende Variablen im Gleichungslöser löschen:
2nd [F6]; F6.

Zahlensysteme

Zahlensysteme eingeben und konvertieren

Unabhängig vom jeweiligen Basis-Modus muß bei der Eingabe einer Dual- oder Hexadezimalzahl stets das entsprechende Präfix verwendet werden.

Eingabe einer Dual- oder Hexadezimalzahl

Geben Sie eine Dualzahl in folgender Form ein:

0b *binaryNumber* (for example: **0b11100110**)

- *binaryNumber*: Dualzahl mit bis zu 32 Stellen
- **0b**: Null (nicht Buchstabe O) und Buchstabe b

Geben Sie eine Hexadezimalzahl in folgender Form ein:

0h *hexadecimalNumber* (for example: **0h89F2C**)

- *hexadecimalNumber*: Hexadezimalzahl mit bis zu 8 Stellen
- **0h**: Null (nicht Buchstabe O und Buchstabe h

Hinweis: Die Zeichen **b** und **h** im Präfix sowie die Hex-Zeichen **A – F** können in Groß- oder Kleinbuchstaben eingegeben werden.

Geben Sie eine Zahl ohne Präfix **0b** oder **0h** ein, wie z.B. 11, so wird diese stets als Dezimalzahl behandelt. Lassen Sie das Präfix **0h** bei einer ein Zeichen von **A – F** enthaltenden Hexadezimalzahl weg, dann wird der gesamte Eintrag oder ein Teil davon als Variable behandelt.

Zahlensysteme konvertieren

Verwenden Sie den Konvertierungsoperator ▶.

integerExpression ▶ **Bin** *integerExpression* ▶ **Dec** *integerExpression* ▶ **Hex**	Zur Eingabe von ▶ drücken Sie 2nd ▶. Sie können Basiskonvertierungen auch aus dem MATH/Base-Menü wählen.

So konvertieren Sie beispielsweise 256 von dezimal in binär:

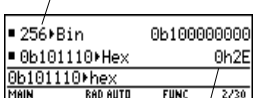
256 ▶ Bin

Hinweis: Handelt es sich bei Ihrer Eingabe nicht um eine ganze Zahl, wird Domain error gemeldet.

So konvertieren Sie 101110 von binär in hexadezimal:

0b101110 ▶ Hex

Für einen binären oder hexadezimalen Eintrag muß das Präfix 0b bzw. 0h verwendet werden.



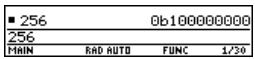
In den Ergebnissen wird zur Kennzeichnung des Zahlensystems das Präfix 0b bzw. 0h verwendet.

Alternative Konvertierungsmethode

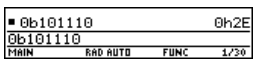
Anstelle der Verwendung von ▶:

1. Stellen Sie **Base**-Modus mit MODE auf das Zahlensystem ein, in welches konvertiert werden soll.
2. Tippen Sie im Hauptbildschirm die zu konvertierende Zahl ein (mit dem richtigen Präfix), und drücken Sie ENTER.

Wenn Base mode = BIN:



Wenn Base mode = HEX:

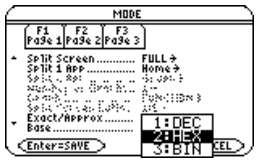


Mathematische Operationen mit Hexadezimal- oder Dualzahlen durchführen

Bei jeder Operation mit ganzen Zahlen kann eine Hexadezimal- oder Dualzahl eingegeben werden. Die Ergebnisse werden gemäß dem jeweiligen Basismodus angezeigt. Beachten Sie aber, dass die Größe der Ergebnisse bei Base = HEX oder BIN durch bestimmte Grenzen beschränkt ist.

Einstellung des Basismodus für angezeigte Ergebnisse

1. Drücken Sie MODE F2, um **Seite 2** des Bildschirms **MODE** anzuzeigen.
2. Gehen Sie zum **Base**-Modus, drücken Sie ⊙, und wählen Sie die erforderliche Einstellung.
3. Drücken Sie ENTER zum Schließen des Bildschirms **MODE**.

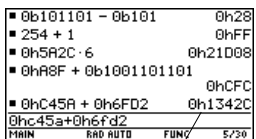


Der **Base**-Modus regelt nur das Anzeigeformat ganzzahliger Ergebnisse.

Hinweis: Der Base-Modus wirkt sich nur auf die Ausgabe aus. Zur Eingabe einer Hexadezimal- oder Dualzahl muß stets das Präfix **0h** bzw. **0b** verwendet werden.

Bruch- und Fließkomma-Ergebnisse werden stets in dezimaler Form angezeigt.

Wenn Base-Mode = HEX:



Präfix 0h im Ergebnis kennzeichnet die Basis.

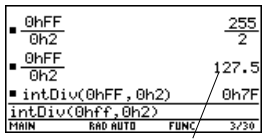
Dividieren mit Base = HEX oder BIN

Bei Base=HEX oder BIN wird ein

Divisionsergebnis nur dann in hexadezimaler oder binärer Form angezeigt, wenn es ganzzahlig ist.

Um sicherzustellen, dass eine Division stets eine ganze Zahl ergibt, verwenden Sie **intDiv( )** anstelle von ÷.

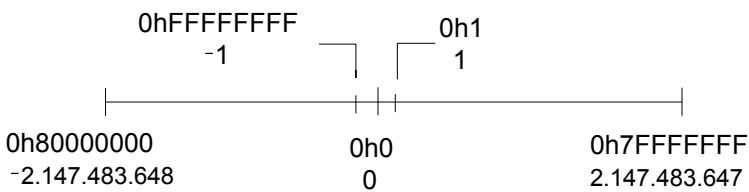
Wenn Base mode = HEX:



Drücken Sie ◆ ENTER, um das Ergebnis in der Form APPROXIMATE anzuzeigen.

Größenbeschränkungen bei Base = HEX oder BIN

Bei Base=HEX oder BIN wird ein ganzzahliges Ergebnis intern als 32-Bit-Dualzahl mit Vorzeichen mit folgendem Bereich (in hexadezimaler und dezimaler Form dargestellt) gespeichert:



Ist der Absolutwert eines Ergebnisses zu groß, um in einer 32-Bit-Dualform mit Vorzeichen gespeichert zu werden, wird eine symmetrische Modulo-Operation durchgeführt, um das Ergebnis in den erforderlichen Bereich zu bringen. Jede Zahl, die größer als 0h7FFFFFFF ist, wird davon betroffen. So werden zum Beispiel 0h80000000 bis 0hFFFFFFFF zu negativen Zahlen.

Bits vergleichen oder manipulieren

Mit folgenden Operatoren und Funktionen lassen sich die Bits in Dualzahlen vergleichen oder manipulieren. Eine ganze Zahl können Sie in jedem Zahlensystem eingeben. Für die Bit-Operationen werden Ihre Einträge automatisch in Dualzahlen konvertiert, und die Ergebnisse werden gemäß dem Basismodus angezeigt.

Boolesche Operationen

Operator mit Syntax	Beschreibung
not *ganze_Zahl*	Gibt das Einerkomplement zurück, wobei alle Bits umgekehrt sind.
(-) *ganze_Zahl*	Gibt das Zweierkomplement zurück, bei dem es sich um das Einerkomplement + 1 handelt.
ganze_Zahl1 **and** *ganze_Zahl2*	Das Ergebnis eines **and**-Vergleichs der einzelnen Bits ist dann 1, wenn beide Bits 1 sind; anderenfalls ist das Ergebnis 0. Der zurückgegebene Wert stellt die Bit-Ergebnisse dar.
ganze_Zahl1 **or** *ganze_Zahl2*	Das Ergebnis eines **or**-Vergleichs der einzelnen Bits ist dann 1, wenn mindestens eines der beiden Bits 1 ist; es ist nur dann 0, wenn wenn beide Bits 0 sind. Der zurückgegebene Wert stellt die Bit-Ergebnisse dar.

Operator mit Syntax	Beschreibung
ganze_Zahl1 **xor** *ganze_Zahl2*	Das Ergebnis eines **xor**-Vergleichs der einzelnen Bits ist dann 1, wenn eines der Bits (nicht aber beide) 1 ist; es ist 0, wenn entweder beide Bits 0 oder beide Bits 1 sind. Der zurückgegebene Wert stellt die Bit-Ergebnisse dar.

Hinweis: Sie können diese Operatoren aus dem MATH/Base-Menü wählen. Ein Beispiel für die Verwendung jedes einzelnen Operators finden Sie im Modul *Technische Referenz*.

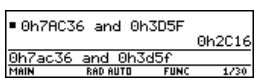
Nehmen Sie an, Sie geben ein:

0h7AC36 **and** 0h3D5F

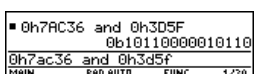
Intern werden die hexadezimalen ganzen Zahlen in eine 32-Bit-Dualzahl mit Vorzeichen konvertiert.

Dann werden die sich entsprechenden Bits verglichen.

Wenn Base mode = HEX:



Wenn Base mode = BIN:



0h7AC36 = 0b00000000000001111010110000110110
and **and**
0h3D5F 0b00000000000000000011110101011111
0b00000000000000000010110000010110 = 0h2C16

Führende Nullen werden im Ergebnis nicht angezeigt.

Das Ergebnis wird gemäß dem Base-Modus angezeigt.

Hinweis: Geben Sie eine ganze Zahl ein, die zum Speichern in einer 32-Bit-Dualform mit Vorzeichen zu groß ist, so wird eine symmetrische Modulo-Operation vorgenommen, die den Wert in den erforderlichen Bereich bringt.

Das Ergebnis wird gemäß dem Base-Modus angezeigt.

Bits rotieren und verschieben

Funktion mit Syntax	Beschreibung
rotate(*ganze_Zahl***)** – oder – **rotate(***ganze_Zahl, Anzahl_Rotationen***)**	Wenn *Anzahl_Rotationen*: • weggelassen wird, — rotieren die Bits um eine Stelle nach rechts (Standard ist ‑1). • negativ ist, — rotieren die Bits um die angegebene Anzahl von Stellen nach rechts. • positiv ist, — rotieren die Bits um die angegebene Anzahl von Stellen nach links. In einer Rotation nach rechts nimmt das ganz rechts stehende Bit die Stelle des ganz links stehenden Bits ein; umgekehrt bei einer Linksrotation.

Funktion mit Syntax	Beschreibung
shift(*ganze_Zahl***)** – oder – **shift(***ganze_Zahl,Anzahl_Verschiebungen***)**	Wenn *Anzahl_Verschiebungen*: • weggelassen wird, — werden die Bits um eine Stelle nach rechts verschoben (Standard ist ⁻1). • negativ ist, — werden die Bits um die angegebene Anzahl von Stellen nach rechts verschoben. • positiv ist, — werden die Bits um die angegebene Anzahl von Stellen nach links verschoben. Bei einer Verschiebung nach rechts wird das ganz rechts stehende Bit abgeschnitten, und als ganz links stehendes Bit wird 0 oder 1 eingesetzt. Bei einer Linksverschiebung wird das ganz links stehende Bit abgeschnitten, und als Bit ganz rechts wird eine 0 eingesetzt.

Hinweis: Sie können diese Funktionen aus dem MATH/Base-Menü wählen. Ein Beispiel für die Verwendung jeder Funktion finden Sie im Modul *Technische Referenz*.

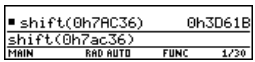
Nehmen Sie an, Sie geben ein:

shift(0h7AC36**)**

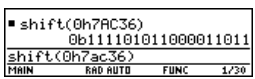
Intern wird die hexadezimale ganze Zahlin eine 32-Bit-Dualzahl mit Vorzeichen konvertiert.

Dann wird die Verschiebung auf die Dualzahl angewendet.

Wenn Base mode = HEX:



Wenn Base mode = BIN:



Jedes Bit wird nach rechts verschoben.

7AC36 = 0b00000000000000111101011000011011 0

0 wird eingesetzt, wenn Bit ganz links 0 ist, und 1, wenn es 1 ist.

Abgeschnitten

b00000000000000111101011000011011 = 0h3D61B

Führende Nullen werden im Ergebnis nicht angezeigt.

Das Ergebnis wird gemäß dem Base-Modus angezeigt.

Hinweis: Geben Sie eine ganze Zahl ein, die zum Speichern in einer 32-Bit-Dualform mit Vorzeichen zu groß ist, so wird eine symmetrische Modulo-Operation vorgenommen, die den Wert in den erforderlichen Bereich bringt.

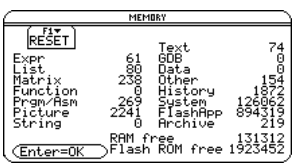
Speicher- und Variablenmanagement

Den Speicher überprüfen und zurücksetzen

Der **MEMORY** Bildschirm zeigt die Nutzung des Speichers an. Sie sehen hier, wieviel Speicher (in Byte) alle Variablen jedes Datentyps derzeit belegen, ob sie im RAM-Speicher oder im Benutzerarchiv gespeichert sind. Mit diesem Bildschirm können Sie den Speicher auch zurücksetzen.

Den MEMORY Bildschirm aufrufen

Drücken Sie 2nd [MEM]. Der folgende Bildschirm stammt von einem Voyage™ 200. (Die Zahlen auf Ihrem Bildschirm **MEMORY** können sich von den angezeigten unterscheiden.)



Prgm/Asm: Umfasst die für den TI-89 Titanium / Voyage™ 200 geschriebenen Programme und alle geladenen Assemblersprachen-Programme
History: Größe der Eingabe-/Antwortpaare, die im Protokoll-Bereich des Hauptbildschirms gespeichert sind
FlashApp: Größe der Flash-Anwendungen
RAM free: Freier Speicherplatz im RAM-Speicher
Flash-ROM free: Freie Kapazität im Flash-ROM

Hinweis: Zur Anzeige der Größe einzelner Variabler und um festzustellen, ob sich diese im Benutzerarchiv befinden, verwenden Sie den **VAR-LINK** Bildschirm.

Zum Schließen des Bildschirms drücken Sie ENTER. Um den Speicher zurückzusetzen, gehen Sie wie folgt vor.

Den Speicher zurücksetzen

Ausgangspunkt ist der **MEMORY** Bildschirm:

1. Drücken Sie F1.
2. Wählen Sie den gewünschten Menüpunkt aus.

Speicher	Beschreibung
RAM	**1:All RAM:** Durch Zurücksetzen des RAM werden sämtliche darin enthaltenen Daten und Programme aus dem Arbeitsspeicher gelöscht. **2:Default:** Systemvariablen und -modi werden auf ihre werksseitig definierten Einstellungen zurückgesetzt. Dies gilt nicht für benutzerdefinierte Variablen, Funktionen oder Verzeichnisse.
Flash-ROM	**1: Archive:** Durch Zurücksetzen dieses Archivspeichers werden alle Daten und Programme aus dem Flash-ROM gelöscht. **2:Flash Apps:** Durch Zurücksetzen dieses Speichers werden alle Flash-Anwendungen aus dem Flash-ROM gelöscht. **3:Both:** Durch zurücksetzen dieser beiden Speicher werden sämtliche Daten, Programme und Flash-Anwendungen aus dem Flash-ROM gelöscht.
All Memory	Durch Zurücksetzen werden alle Daten, Programme und Flash-Anwendungen aus dem RAM und aus dem Flash-ROM gelöscht.

Wichtig: Um einzelne (anstatt alle) Variablen zu löschen, verwenden Sie **VAR-LINK**.

3. Bestätigen Sie nach Aufforderung mit [ENTER].

   Der TI-89 Titanium / Voyage™ 200 zeigt eine Meldung an, wenn das Zurücksetzen abgeschlossen ist.

   Hinweis: Sie annullieren das Zurücksetzen, indem Sie [ESC] statt [ENTER] drücken.

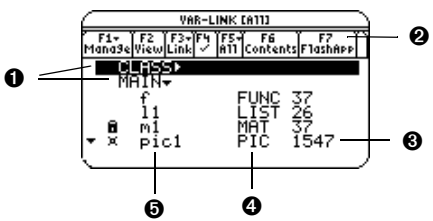
4. Quittieren Sie die Meldung durch Drücken von [ENTER].

Den VAR-LINK Bildschirm anzeigen

Der **VAR-LINK** Bildschirm zeigt die zur Zeit definierten Variablen und Verzeichnisse an. Wenn Sie diesen Bildschirm aufgerufen haben, können Sie verschiedene Verwaltungsoperationen für Variablen und/oder Verzeichnisse vornehmen.

Den VAR-LINK Bildschirm aufrufen

Drücken Sie 2nd [VAR-LINK] Standardmäßig zeigt der **VAR-LINK** Bildschirm die Liste sämtlicher benutzerdefinierter Variablen in allen Verzeichnissen und mit allen Datentypen an.



❶ Verzeichnisnamen (alphabetisch sortiert)
❷ Zeigt installierte Flash-Anwendungen an
❸ Größe in Byte
❹ Datentyp
❺ Variablennamen (alphabetisch sortiert)

Symbol...	Bedeutung...
▸	Komprimierte Verzeichnisdarstellung (rechts neben dem Ordnernamen).

Symbol...	Bedeutung...
▾	Erweiterte Verzeichnisdarstellung (rechts neben Verzeichnisnamen).
▾	Bildlauf zur Anzeige weiterer Variablen und/oder Verzeichnisse (im unteren linken Teil des Bildschirms).
✓	Wenn mit F4 ausgewählt.
🔒	Gesperrt
⌧	Archiviert

Die Liste scrollen:

- Drücken Sie ⊖ oder ⊕. (Mit 2nd ⊖ bzw. 2nd ⊕ können Sie seitenweise scrollen.)
  – oder –
- Geben Sie einen Buchstaben ein. Wenn Variablennamen existieren, die damit beginnen, rückt der Cursor zum ersten Variablennamen vor, der mit diesem Buchstaben beginnt und markiert den Namen.

Hinweis: Geben Sie einen Buchstaben mehrmals hintereinander ein, um die Variablennamen zu durchlaufen, die mit diesem Buchstaben beginnen.

Anzeige der Variablentypen auf dem VAR-LINK Bildschirm

Typ	Beschreibung
ASM	Assemblersprachen-Programm
DATA	Daten
EXPR	Term bzw. Ausdruck
FUNC	Funktion
GDB	Graphikeinstellungen
LIST	Liste
MAT	Matrix
PIC	Grafikbild
PRGM	Programm
STR	Zeichenkette (String)
TEXT	Text-Variable

Typen, die nicht oben aufgeführt werden, sind verschiedene, durch Anwendungen genutzte Datentypen.

Den VAR-LINK Bildschirm schließen

Um den **VAR-LINK** Bildschirm zu verlassen (zu schließen) und zur aktuellen Anwendung zurückzukehren, verwenden Sie [ENTER] oder [ESC] wie nachfolgend erläutert.

Drücken Sie:	Um:
[ENTER]	Den markiert angezeigten Variablen- oder Verzeichnis namen an die Cursorposition in der aktuellen Anwendung zu kopieren.
[ESC]	Zur aktuellen Anwendung zurückzukehren, ohne den markierten Namen dort einzufügen.

Umgehen mit Variablen und Verzeichnissen mit VAR-LINK

Sie können sich auf dem **VAR-LINK** Bildschirm den Inhalt einer Variablen anzeigen lassen. Sie können auch einen oder mehrere angezeigte Posten auswählen und die nachfolgend erläuterten Operationen ausführen.

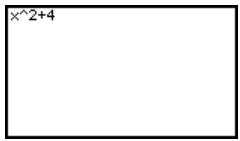
Den Inhalt einer Variablen anzeigen

Sie können den Inhalt aller Variablentypen außer **ASM**, **DATA**, **GDB**, und durch Flash Apps erzeugte Variablen anzeigen lassen. Eine **DATA**-Variable beispielsweise müssen Sie im Daten/Matrix-Editor öffnen.

1. Bringen Sie auf dem **VAR-LINK** Bildschirm den Cursor auf die gewünschte Variable, um sie zu markieren.

2. Drücken Sie folgende Tasten:
   [2nd] [F6]
   [F6]

   Wenn sie ein Verzeichnis markieren, wird die Anzahl der Variablen in diesem Verzeichnis angezeigt.



3. Drücken Sie eine beliebige Taste, um zu **VAR-LINK** zurückzukehren.

Hinweis: Sie können den Inhalt einer Variablen in diesem Bildschirm nicht ändern.

Listenposten auswählen

Für andere Operationen wählen Sie die gewünschte(n) Variable(n) bzw. das/die Verzeichnis/se aus.

Auswählen von:	Vorgehensweise:
Eine einzige Variable oder ein einziges Verzeichnis	Bringen Sie den Cursor in die gewünschte Zeile, drücken Sie dann [F4].

Auswählen von:	Vorgehensweise:
Eine Gruppe von Variablen oder Verzeichnis	Markieren Sie jedes einzelne Element, und drücken Sie F4. Links von jedem ausgewählten Element erscheint ein ✓. Wenn Sie ein Verzeichnis auswählen, werden alle Variablen in diesem Verzeichnis ausgewählt. Verwenden Sie F4, um ein Element auszuwählen bzw. eine Auswahl aufzuheben.
Alle Variablen und alle Verzeichnis 	Drücken Sie ▶, um den Ordner zu öffnen und dann F5 **All**, und wählen Sie **1:Select All**. Durch Auswahl von **3:Select Current** wird die letzte Gruppe vom Elementen ausgewählt, die bei der aktuellen **VAR-LINK**-Sitzung an Ihr Gerät übertragen wurde. Durch Auswahl von **4:Expand All** bzw. **5:Collapse All** werden Ordner bzw. Flash-Anwendungen erweitert oder komprimiert angezeigt.

Hinweis: Mit ◀ oder ▶ erfolgt ein Wechsel zwischen erweiterter oder komprimierter Darstellung eines durch Hervorhebung markierten Verzeichnisses.

Verzeichnisse und Variablen

Mit Verzeichnissen können Sie Variablen auf einfache Weise verwalten, indem Sie Variablen in zusammengehörige Gruppen aufteilen.

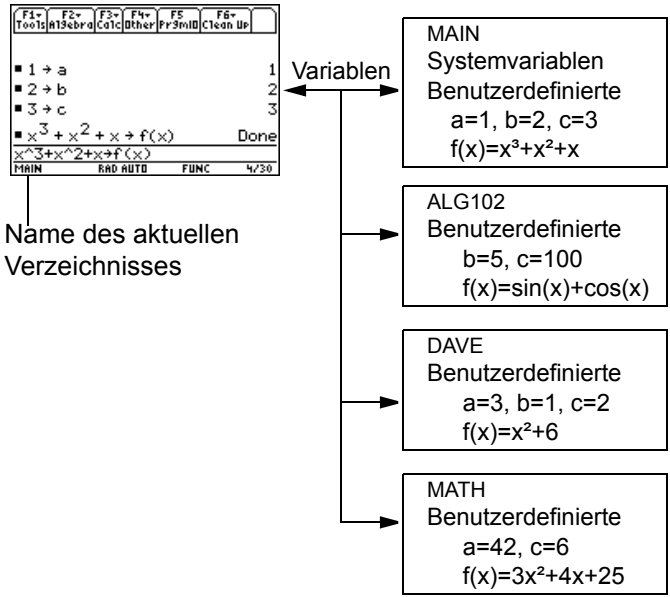
Der TI-89 Titanium / Voyage™ 200 Graphischer Rechner verfügt über ein vorgegebenes Verzeichnis namens **MAIN**. Wenn Sie nicht gerade weitere Ordner erstellen und einen vom Benutzer erstellten Ordner zum aktuellen Ordner wählen, werden alle Variablen per

Vorgabe im Ordner **MAIN** gespeichert. Eine Systemvariable oder eine Variable mit reserviertem Namen kann hingegen nur im Verzeichnis **MAIN** gespeichert werden.

Beispiele für Variablen, die nur in MAIN gespeichert werden können
Fenstervariablen (**xmin**, **xmax** ,etc.)
Tabellen-Setup-Variablen (**TblStart**, Δ**Tbl**, etc.)
Y= Editor-Funktionen (**y1(x)**, etc.)

Wenn Sie weitere Verzeichnisse erstellen, können Sie separate Sätze benutzerdefinierter Variablen speichern (einschließlich selbstdefinierter Funktionen). Sie können beispielsweise je ein Verzeichnis für die verschiedenen Anwendungen des TI-89 Titanium / Voyage™ 200 erstellen (Math, Text-Editor etc.) oder auch für verschiedene Klassen. Eine benutzerdefinierte Variable kann in jedem vorhandenen Verzeichnis gespeichert werden.

Die benutzerdefinierten Variablen in einem Verzeichnis sind unabhängig von den Variablen in anderen Verzeichnissen. Deshalb können in Verzeichnissen separate Variablensätze mit gleichen Variablennamen aber unterschiedlichen Werten gespeichert werden.



Sie können innerhalb eines Verzeichnisses kein weiteres Verzeichnis erstellen.

Auf die Systemvariablen im Verzeichnis **MAIN** kann unabhängig vom aktuellen Verzeichnis stets direkt zugegriffen werden.

Hinweis: Benutzerdefinierte Variablen werden, falls nicht anders von Ihnen festgelegt, im aktuellen Verzeichnis gespeichert.

Ein Verzeichnis vom VAR-LINK Bildschirm aus erstellen

1. Drücken Sie 2nd [VAR-LINK].

2. Drücken Sie F1 **Manage**, und wählen Sie **5:Create Folder**.

3. Geben Sie einen eindeutigen Verzeichnisnamen mit bis zu acht Zeichen ein, und drücken Sie zweimal ENTER.

Nach der Erstellung eines Verzeichnisses in **VAR-LINK** ist dieses nicht automatisch das aktuelle Verzeichnis.

Ein Verzeichnis vom Hauptbildschirm aus erstellen

Geben Sie auf dem Startbildschirm des Rechners den Befehl **NewFold** ein.

NewFold *Verzeichnisname*

└ Name des zu erstellenden Verzeichnisses. Dieser neue Verzeichnis ist dann automatisch das aktuelle Verzeichnis.

Das aktuelle Verzeichnis mit dem Hauptbildschirm einstellen

Geben Sie die Funktion **setFold** auf dem Startbildschirm des Rechners ein.

setFold (*Verzeichnisname*)

└ Bei der Funktion **setFold** müssen Sie den Verzeichnisnamen in runde Klammern setzen.

Bei Ausführung von **setFold** wird der Name des zuvor aktuellen Verzeichnisses zurückgegeben.

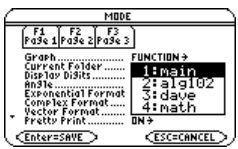
Das aktuelle Verzeichnis mit dem Dialogfeld MODE einstellen

1. Drücken Sie MODE.

2. Markieren Sie die **Current Folder**-Einstellung.

3. Drücken Sie ▶, um ein Menü der vorhandenen Verzeichnisse zu öffnen.

   Hinweis: Möchten Sie das Menü oder das Dialogfeld verlassen, ohne Änderungen zu speichern, drücken Sie ESC.



4. Wählen Sie das gewünschte Verzeichnis. Sie haben folgende Möglichkeiten:
   - Markieren Sie den Verzeichnisnamen, und drücken Sie ENTER.

     – oder –

   - Drücken Sie die entsprechende Ziffer bzw. den Buchstaben des Verzeichnisses.

5. Drücken Sie ENTER, um die Änderungen zu speichern und das Dialogfeld zu schließen.

Variable oder Verzeichnis umbenennen

Bitte beachten Sie, daß beim Wählen eines Verzeichnisses mit F4 automatisch auch die in diesem Verzeichnis befindlichen Variablen gewählt werden. Falls erforderlich, können Sie mit F4 die Auswahl einzelner Variablen aufheben.

1. Wählen Sie in **VAR-LINK** die Variablen und/oder Ordner.
2. Drücken Sie F1 **Manage** und wählen Sie **3:Rename**.
3. Geben Sie einen eindeutigen Namen ein, und drücken Sie zweimal ENTER.

   Wenn Sie mehrere Posten ausgewählt haben, werden Sie für jeden zur Eingabe eines neuen Namens aufgefordert.

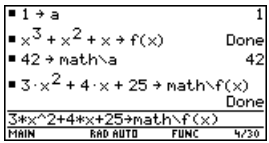
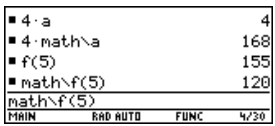
Variablen aus anderen Verzeichnissen verwenden

Sie können auf eine benutzerdefinierte Variable oder Funktion zugreifen, die sich nicht im aktuellen Verzeichnis befindet. Geben Sie anstelle des bloßen Variablennamens den gesamten Pfadnamen an.

Ein Pfadname hat folgende Form:

Verzeichnisname \ *Variablenname*
– oder –
Verzeichnisname \ *Funktionsname*

Beispiel:

Wenn Current Folder = MAIN	Verzeichnis und Variable
	MAIN a=1 f(x)=x³+x²+x
	MATH a=42 f(x)=3x²+4x+25

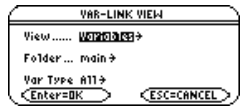
Eine Liste der vorhandenen Verzeichnisse und Variablen können Sie einsehen, indem Sie 2nd [VAR-LINK] drücken. Mit dem **VAR-LINK**-Bildschirm können Sie eine Variable markieren und ENTER drücken, um den Namen dieser Variablen in die Eingabezeile der gerade geöffneten Anwendung einzufügen. Beim Einfügen eines Variablennamens, der sich nicht im aktuellen Verzeichnis befindet, wird der Pfad (*Verzeichnisname\Variablenname*) eingefügt.

Nur bestimmtes Verzeichnis und/oder Variablentyp bzw. Flash-Anwendung anzeigen

Wenn Sie eine große Zahl von Variablen oder Verzeichnis angelegt haben, kann es schwierig sein, eine bestimmte Variable zu finden. Aus diesem Grund können Sie die Ansicht des **VAR-LINK** Bildschirms ändern, um nur bestimmte Informationen anzeigen zu lassen.

Ausgangspunkt ist der **VAR-LINK** Bildschirm:

1. Drücken Sie F2 **View**.
2. Markieren Sie die Einstellung, die Sie ändern möchten, und drücken Sie ⊙. Es wird ein Menü mit Auswahlmöglichkeiten angezeigt.



View — Ermöglicht die Auswahl anzuzeigender Variablen, Flash-Anwendungen oder Systemvariablen.

Hinweis: Um sich die Systemvariablen (Y= Editor Funktionen, Window-Variablen etc.) anzeigen zu lassen, wählen Sie **3:System**.

Folder — Enthält immer die Einträge **1:All** und **2:main**, weitere Verzeichnis jedoch nur, wenn Sie Verzeichnis angelegt haben.

Var Type — Listet die Variablentypen auf.

↓ — zeigt an, daß Sie durch Scrollen weitere Variablentypen erreichen.

3. Wählen Sie die gewünschte Einstellung.
4. Wenn wieder der **VAR-LINK VIEW** Bildschirm angezeigt wird, drücken Sie ENTER.

Auf dem daraufhin aktualisierten Bildschirm **VAR-LINK** werden ausschließlich das gewählte Verzeichnis, Variablentyp oder die Flash-Anwendung angezeigt.

Variable von einem Verzeichnis in ein anderes kopieren oder verschieben

Es muß außer **MAIN** noch mindestens ein weiteres Verzeichnis vorhanden sein. Mit **VAR-LINK** können Sie Variable nicht innerhalb desselben Verzeichnisses kopieren.

1. Wählen Sie auf dem **VAR-LINK** Bildschirm die gewünschten Variablen aus.
2. Drücken Sie F1 **Manage**, und wählen Sie **2:Copy** oder **4:Move**.
3. Wählen Sie den Zielordner.

4. Drücken Sie ENTER. Die kopierten oder verschobenen Variablen behalten ihre ursprünglichen Namen.

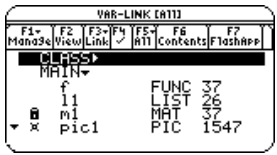
   Hinweis: Um innerhalb eines Verzeichnisses eine Kopie einer Variable zu erstellen, verwenden Sie STO▸ (z. B. a1→a2) oder den Befehl **CopyVar** auf dem Hauptbildschirm.

Sperren oder entsperren von Variablen, Ordnern und Flash-Anwendungen

Ist eine Variable gesperrt, können Sie diese weder löschen, umbenennen noch ihr einen neuen Wert zuweisen. Sie können sie jedoch kopieren, verschieben und ihren Inhalt anzeigen. Ist ein Verzeichnis gesperrt, können Sie darin befindliche Variable normal benutzen (sofern sie nicht gesperrt sind), aber Sie können das Verzeichnis nicht löschen.

1. Wählen Sie im **VAR-LINK** Bildschirm die gewünschten Variablen, Verzeichnis bzw. die Flash-Anwendung aus.
2. Drücken Sie F1 **Manage**, und wählen Sie **6:Lock Variable** oder **7:UnLock Variable**.

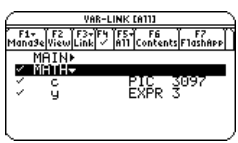
- weist auf gesperrte Variable oder Ordner im RAM hin.
- kennzeichnet eine archivierte Variable, die automatisch gesperrt ist.



Ein Verzeichnis mit dem VAR-LINK Bildschirm löschen

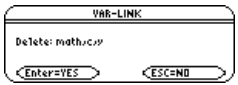
Wenn Sie auf dem Bildschirm **VAR-LINK** ein Verzeichnis löschen, werden auch alle Variablen in diesem Verzeichnis gelöscht. Das Verzeichnis **MAIN** können Sie nicht löschen.

1. Drücken Sie 2nd [VAR-LINK].
2. Drücken Sie F4 um das/die zu löschenden Verzeichnisse zu wählen. (Die Variablen des Verzeichnisses werden automatisch gewählt.)



3. Drücken Sie F1 **1:Delete** oder ←.

4. Drücken Sie ENTER um das Löschen des Verzeichnisses einschließlich aller enthaltenen Variablen zu bestätigen.



Löschen einer Variablen oder eines Verzeichnisses vom Hauptbildschirm aus

Vor dem Löschen eines Verzeichnisses vom Hauptbildschirm aus, müssen Sie alle Variablen löschen, die in diesem Verzeichnis gespeichert sind.

- Geben Sie zum Löschen einer Variablen auf dem Startbildschirm des Rechners (Hauptbildschirm) den Befehl **DelVar** ein.

  DelVar *var1* [, *var2*] [, *var3*] ...

- Geben Sie zum Löschen eines leeren Verzeichnisses auf dem Startbildschirm des Rechners den Befehl **DelFold** ein.

  DelFold *Verzeichnis1* [, *Verzeichnis2*] [, *Verzeichnis3*] ...

  Hinweis: Das Verzeichnis **MAIN** können Sie nicht löschen.

Einen Variablennamen in eine Anwendung kopieren

Nehmen Sie an, Sie geben einen Term im Hauptbildschirm ein, haben aber den Namen einer Variablen vergessen, die Sie benutzen möchten. Sie können den **VAR-LINK** Bildschirm aufrufen, die Variable dort aus einer Liste auswählen und den Variablennamen unmittelbar in die Eingabezeile des Hauptbildschirms übernehmen.

In welche Anwendungen können Sie Namen übernehmen?

Sie können in folgende Anwendungen einen Variablennamen an der aktuellen Cursorposition einfügen.

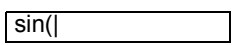
- Hauptbildschirm, Y= Editor, Tabelleneditor oder Daten/Matrix-Editor — Der Cursor muß sich in der Eingabezeile befinden.
- Texteditor, Window-Editor, Numerischen Gleichungslöser oder Programmeditor — Der Cursor kann sich an einer beliebigen Stelle auf dem Bildschirm befinden.

In vielen Flash-Anwendungen können Sie einen Variablennamen auch einfach an der aktuellen Cursorposition einfügen.

Vorgehensweise

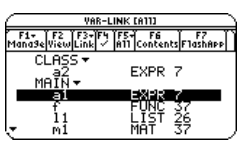
Ausgangspunkt ist eine der vorgenannten Anwendungen:

1. Setzen Sie den Cursor an die Stelle, an der der Variablenname eingefügt werden soll.

2. Drücken Sie 2nd [VAR-LINK].
3. Markieren Sie die gewünschte Variable.

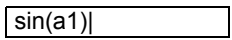
   Hinweis: Sie können auch Verzeichnisnamen markieren und einfügen.

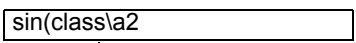
4. Drücken Sie [ENTER] um den Variablennamen zu übernehmen.

   Hinweis: Eingefügt wird der Name der Variablen, nicht ihr Inhalt. (Benutzen Sie [2nd] [RCL] statt [2nd] [VAR-LINK], um ihren Inhalt aufzurufen.)

5. Fahren Sie mit der Eingabe des Terms fort.



Wenn Sie einen Variablennamen übernehmen, der sich nicht im aktuellen Verzeichnis befindet, wird der Pfad der Variablen eingefügt.



CLASS ist nicht das aktuelle Verzeichnis. Wenn Sie die Variable a2 in CLASS markieren, wird der Pfad eingefügt.

Eine Variable archivieren und aus dem Archiv entnehmen

Verwenden Sie zum interaktiven Archivieren einer Variablen und zum Entnehmen einer Variablen aus dem Archiv den **VAR-LINK** Bildschirm. Diese Vorgänge sind auch im Hauptbildschirm oder von einem Programm aus möglich.

Warum Variablen archivieren?

Das Benutzerdatenarchiv dient zum:

- Speichern von Daten, Programmen oder anderen Variablen an einem sicheren Ort, an welchem sie weder bearbeitet noch versehentlich gelöscht werden können.
- Freiräumen von RAM-Speicherplatz durch das Archivieren von Variablen. Zum Beispiel:
  - Sie können Variablen archivieren, auf die Sie zwar zugreifen, die Sie aber weder bearbeiten noch ändern müssen. Ebenso Variablen, die zwar gegenwärtig nicht verwendet werden, aber für eine künftige Verwendung aufbewahrt werden müssen.

    Hinweis: Systemvariable oder Variable mit reservierten Namen können nicht archiviert werden.
  - Wenn Sie zusätzliche Programme für Ihren TI-89 Titanium / Voyage™ 200 Graphischer Rechner erwerben, besonders, wenn sie groß sind, kann es sein, dass Sie RAM-Speicherplatz freiräumen müssen, bevor die Installation dieser Programme möglich ist.

Zusätzlicher RAM-Speicherplatz kann die Ausführungszeit für bestimmte Berechnungstypen verbessern.

Im VAR-LINK Bildschirm

Zum Archivieren oder Entnehmen aus dem Archiv

1. Drücken Sie 2nd [VAR-LINK], um den **VAR-LINK** Bildschirm anzuzeigen.

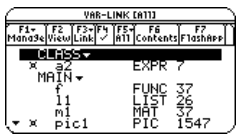
2. Wählen Sie eine oder mehrere Variablen aus einem oder mehreren verschiedenen Verzeichnissen. (Durch die Wahl eines Verzeichnisnamens können Sie ein komplettes Verzeichnis wählen.)

   Hinweis: Um eine einzelne Variable zu wählen, markieren Sie diese. Zum Wählen mehrerer Variablen markieren Sie jede gewünschte Variable und drücken F4 ✓.

3. Drücken Sie F1, und wählen Sie entweder:

   8:Archive Variable
   – oder –
   9:Unarchive Variable

Wenn Sie **8:Archive Variable** wählen, werden die Variablen in das Benutzerarchiv übertragen.



⌑ = archivierte Variable

Auf eine archivierte Variable können Sie wie auf jede gesperrte Variable zugreifen. Eine archivierte Variable befindet sich in jedem Fall weiterhin in ihrem ursprünglichen Verzeichnis; sie wird einfach nur im Benutzerarchiv anstatt im RAM-Speicher gespeichert.

Hinweis: Eine archivierte Variable wird automatisch gesperrt. Sie können zwar auf die Variable zugreifen, sie aber weder bearbeiten noch löschen.

Im Hauptbildschirm oder von einem Programm aus

Verwenden Sie die Befehle **Archive** und **Unarchiv**.

Archive *variable1*, *variable2*, …

Unarchiv *variable1*, *variable2*, …

Wenn eine “Abfallentsorgungs”-Meldung angezeigt wird

Wenn Sie das Benutzerarchiv häufig verwenden, erhalten Sie möglicherweise ab und zu eine “Abfallentsorgungs”-Meldung. Sie wird angezeigt, wenn Sie versuchen, eine Variable zu archivieren und nicht genug freier Archivspeicherplatz verfügbar ist. Der TI-89 Titanium / Voyage™ 200 versucht, die archivierten Variablen umzuordnen, um Speicherplatz freizuräumen.

Reaktion auf die “Abfallentsorgungs”-Meldung

Wenn Sie nebenstehende Meldung erhalten:

- Drücken Sie ENTER, um den Archiviervorgang fortzusetzen.
  – oder –
- Drücken Sie zum Abbrechen ESC.

Ob die Variable nach der Abfallbeseitigung archiviert werden kann, hängt davon ab, wieviel Speicherplatz freigeräumt wurde. Falls sie nicht archiviert wird, sollten Sie einige Variablen aus dem Archiv entnehmen und es erneut versuchen.

Warum wird die Abfallentsorgung nicht automatisch ohne Meldung durchgeführt?

Die Meldung:

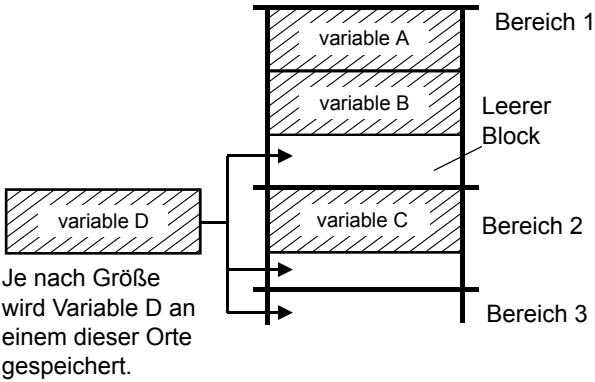
- Teilt Ihnen mit, warum ein Archivierungsvorgang länger als üblich dauert. Sie warnt Sie auch davor, dass der Vorgang fehlschlägt, wenn nicht genügend Speicherplatz vorhanden ist.
- Kann Sie ebenso warnen, wenn das Programm in einer Schleife festhängt, die das Benutzerarchiv wiederholt füllt. Löschen Sie den Archiviervorgang und suchen Sie nach der Ursache.

Warum ist die Abfallentsorgung erforderlich?

Das Benutzerarchiv ist in Sektoren unterteilt. Zunächst werden die Variablen der Reihe nach in Sektor 1 gespeichert. Dies erfolgt bis zum Ende des Sektors. Sobald dieser Sektor nicht mehr über genug Speicherplatz verfügt, wird die nächste Variable am Anfang des nächsten Sektors gespeichert. Dabei entsteht in der Regel ein leerer Block am Ende des vorigen Sektors.

Jede Variable, die Sie archivieren, wird im ersten leeren Block der erforderlichen Größe gespeichert.

Hinweis: Eine archivierte Variable wird innerhalb eines zusammenhängenden Blocks eines Sektors gespeichert. Sie kann die Sektorgrenze nicht überlappen.

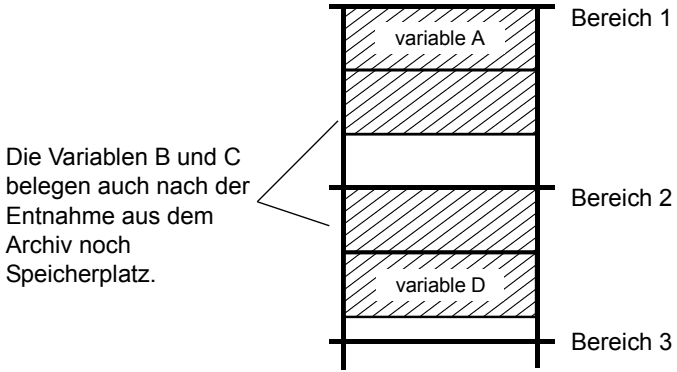


Dieser Vorgang wird bis zum Ende des letzten Sektors fortgesetzt. Je nach Größe der einzelnen Variablen können die leeren Blöcke eine bedeutende Speicherplatzmenge darstellen.

Hinweis: Eine Abfallentsorgung erfolgt, wenn die zu archivierende Variable größer als jeder leere Block ist.

Wie sich das Entnehmen einer Variablen aus dem Archiv auf den Vorgang auswirkt

Wenn Sie eine Variable aus dem Archiv entnehmen, wird diese ins RAM kopiert und nicht wirklich aus dem Benutzerarchiv gelöscht.

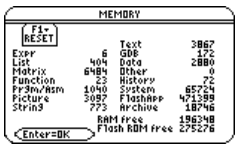


Aus dem Archiv entnommene Variablen sind als “löschbereit” markiert; d.h. dass sie bei der nächsten Abfallentsorgung gelöscht werden.

Wenn der MEMORY Bildschim genügend freien Speicherplatz anzeigt

Auch wenn der **MEMORY** Bildschirm anzeigt, dass zum Archivieren einer Variablen genügend Speicherplatz vorhanden ist, kann es sein, dass Sie eine Abfallentsorgungs-Meldung erhalten.

Dieser Speicherbildschirm des TI-89 Titanium zeigt den nach der Entfernung aller als “löschbereit” markierter Variablen verfügbaren Speicherplatz an.



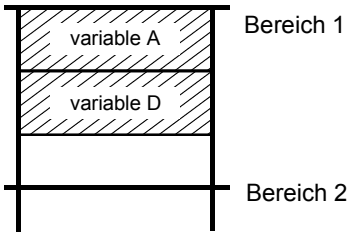
Wenn Sie eine Variable aus dem Flash-ROM entnehmen, vergrößert sich unverzüglich die Archive free-Menge. Der Platz ist aber erst nach der nächsten Abfallentsorgung verfügbar.

Der Voyage™ 200 Graphischer Rechner stellt dem Benutzer über 2,7 MB frei verfügbaren Flash-ROM-Speicher bereit. Die gesamten 2,7 MB können für Flash Apps verwendet werden, aber nur etwa 1 MB davon kann für Datenarchive des Benutzers verwendet werden.

Der Abfallentsorgungs-Vorgang

Der Abfallentsorgungs-Vorgang:

- Löscht die aus dem Archiv entnommenen Variablen aus dem Benutzerarchiv.
- Ordnet die übrigen Variablen neu in aufeinanderfolgenden Blöcken an.



Speicherfehler beim Zugriff auf eine archivierte Variable

Eine archivierte Variable wird wie eine gesperrte Variable behandelt. Sie können auf sie zugreifen, sie aber weder bearbeiten noch löschen. In einigen Fällen kann beim Versuch, auf eine archivierte Variable zuzugreifen, jedoch ein Speicherfehler auftreten.

Wodurch wird der Speicherfehler verursacht?

Die **Memory Error**-Meldung wird dann angezeigt, wenn nicht genügend freier RAM-Speicherplatz vorhanden ist, um auf die archivierte Variable zuzugreifen. Sie fragen sich nun vielleicht, was der RAM-Speicher mit einer im Benutzerarchiv gespeicherten Variable zu tun hat. Dies erklärt sich dadurch, daß folgende Operationen nur dann durchgeführt werden können, wenn sich eine Variable im RAM-Speicher befindet.

- Öffnen einer Textvariablen im Text-Editor.
- Öffnen einer Datenvariablen, einer Liste oder einer Matrix im Daten/Matrix-Editor.
- Öffnen eines Programms oder einer Funktion im Programm-Editor.
- Ausführung eines Programms oder Bezugnahme auf eine Funktion.

Hinweis: Wie weiter unten beschrieben, ist das Öffnen oder Ausführen einer archivierten Variablen nur durch eine temporäre Kopie möglich. Sie können keine Änderungen an der Variablen speichern.

Damit Sie Variablen nicht unnötigerweise aus dem Archiv entnehmen müssen, nimmt TI-89 Titanium / Voyage™ 200 eine “versteckte” Kopie vor. Wenn Sie beispielsweise ein Programm ausführen, das im Benutzerdatenarchiv gespeichert ist, führt TI-89 Titanium / Voyage™ 200 folgende Schritte durch:

1. Kopiert das Programm in RAM.
2. Führt das Programm aus.
3. Löscht die Kopie nach Programmende aus dem RAM-Speicher.

Die Fehlermeldung wird angezeigt, wenn nicht genügend RAM-Speicherplatz für die temporäre Kopie vorhanden ist.

Hinweis: Außer bei Programmen und Funktionen wird durch die Bezugnahme auf eine archivierte Variable keine Kopie davon hergestellt. Wenn die Variable ab archiviert ist, wird Sie bei der Durchführung von **6∗ab** nicht kopiert.

Beheben des Fehlers

So räumen Sie den für den Zugriff auf die Variable erforderlichen RAM-Speicherplatz frei:

1. Stellen Sie über den **VAR-LINK** Bildschirm ([2nd] [VAR-LINK]) fest, wie groß die archivierte Variable ist, auf welche Sie zugreifen möchten.
2. Prüfen Sie über den **MEMORY** Bildschirm ([2nd] [MEM]) wie groß RAM free ist.
3. Räumen Sie den erforderlichen Speicherplatz frei, indem Sie:

- Unbenötigte Variable aus dem RAM-Speicher löschen.
- Große Variable oder Programme archivieren (vom RAM-Speicher in das Benutzerarchiv übertragen).

Hinweis: In der Regel muß RAM free größer als die archivierte Variable sein.

Aktivitäten

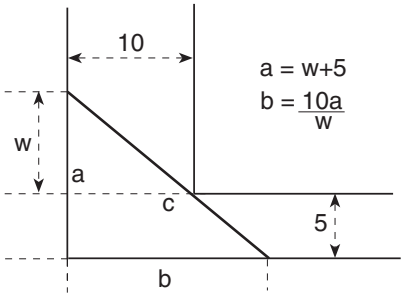
Die Stange-Ecke-Aufgabe

An einer Gebäudeecke treffen ein 10 Fuß breiter Korridor und ein 5 Fuß breiter Korridor aufeinander. Ermitteln Sie die maximale Länge einer Stange, die um diese Ecke transportiert werden kann, wobei die Stange waagerecht bleiben muß.

Maximale Länge der Stange im Korridor

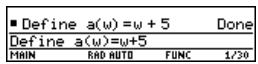
Die maximale Länge einer Stange c ist die kürzeste Strecke, die von einem Außenrand eines Korridors ausgeht, die Innenkante berührt und an der gegenüberliegenden Korridorwand endet (wie in der nachfolgenden Abbildung dargestellt).

Verwenden Sie ähnliche Dreiecke und den Satz des Pythagoras, um die Länge ***c*** abhängig von ***w*** zu finden. Ermitteln Sie anschließend die Werte, bei denen die erste Ableitung von **c(w)** null ergibt. Der Minimalwert von **c(w)** ist die maximale Länge der Stange.

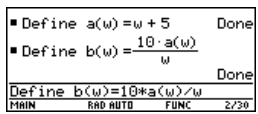


1. Definieren Sie den Term für die Seite ***a*** abh. von ***w***, und speichern Sie ihn in **a(w)**.

   Hinweis: Benutzen Sie beim Definieren einer Funktion Namen, die mehrere Zeichen lang sind.

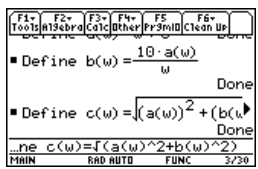


2. Definieren Sie den Term für die Seite ***b*** abh. von ***w***, und speichern Sie ihn in **b(w)**.



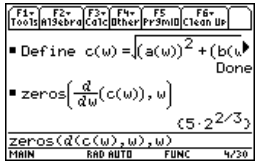
3. Verwenden Sie den Term für Seite ***c*** abhängig von ***w***, und speichern Sie ihn als **c(w)**.

   Geben Sie folgendes ein:
   Define c(w)= √(a(w)^2+b(w)^2)



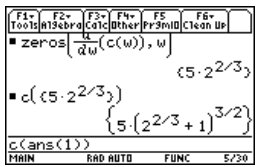
4. Verwenden Sie den Befehl **zeros( )** zur Berechnung der Werte, an denen die erste Ableitung von c(w) Null wird, um das Minimum von c(w) zu finden.

   Hinweis: Die maximale Länge der Stange ist der Minimalwert von **c(w)**.



5. Berechnen Sie die exakte Maximallänge der Stange.

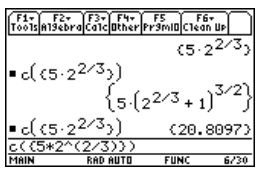
   Geben Sie ein: **c** ([2nd] [ANS])



6. Berechnen Sie die Maximallänge der Stange numerisch.

   Ergebnis: Ungefähr 20,8097 Fuß.

   Hinweis: Kopieren Sie das Ergebnis von Schritt 4, fügen Sie es in der Eingabezeile innerhalb der Klammern von c( ) ein und drücken Sie [◆] [ENTER].



Herleitung der “quadratischen Formel”

In dieser Anwendung wird gezeigt, wie Sie die “quadratische Formel”

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

herleiten. Nähere Einzelheiten zu den Befehlen in diesem Beispiel finden Sie in *Symbolisches Rechnen*.

Berechnungen zur Herleitung der “quadratischen Formel”

Führen Sie die folgenden Schritte aus, um die “quadratische Formel” durch quadratische Ergänzung herzuleiten.

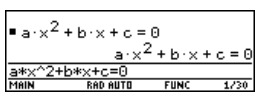
1. Löschen Sie alle aus einem Zeichen bestehenden Variablen aus dem aktuellen Verzeichnis.

   2nd F6

   F6

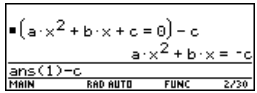
   Wählen Sie **1:Clear a-z**, und bstätigen Sie die Auswahl mit ENTER.

2. Geben Sie im Hauptbildschirm die allgemeine quadratische Gleichung ein: ax2+bx+c=0.



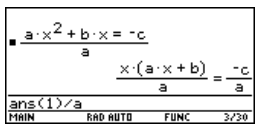
3. Subtrahieren Sie links und rechts vom Gleichheitszeichen den gleichen Wert.
   [2nd] [ANS] [−] [alpha] **C**
   [2nd] [ANS] [−] **C**

   Hinweis: In diesem Beispiel wird das Ergebnis der letzten Berechnung (die letzte Antwort) benutzt, um weitere Berechnungen vorzunehmen. Diese Besonderheit des TI-89 Titanium / Voyage™ 200 verringert den Eingabeumfang und Eingabefehler.

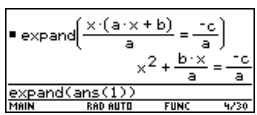


4. Dividieren Sie beide Gleichungsseiten durch den führenden Koeffizienten a.

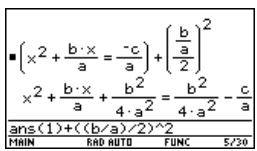
   Hinweis: Verwenden Sie wie in Schritt 3 die letzte Antwort ([2nd] [ANS]) auch in den Schritten 4 bis einschließlich 9.



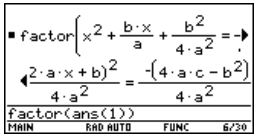
5. Benutzen Sie die Funktion **expand( )**, um das Ergebnis der letzten Antwort auszumultiplizieren



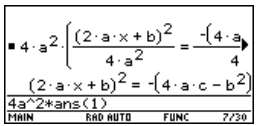
6. Ergänzen Sie quadratisch durch Addition von ((b/a)/2)2 auf beiden Seiten der Gleichung.



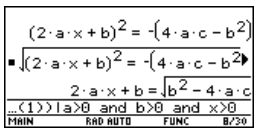
7. Vereinfachen Sie mit der Funktion **factor( )**.



8. Multiplizieren Sie beide Seiten der Gleichung mit $4a^2$.

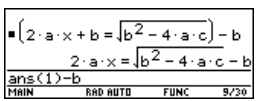


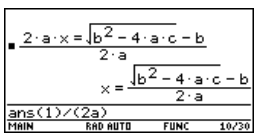
9. Ermitteln Sie die Quadratwurzel für beide Seiten der Gleichung, und zwar mit der Beschränkung, daß **a>0** und **b>0** und **x>0**.



10. Lösen Sie nach x auf, indem Sie von beiden Seiten b subtrahieren und dann eine Division durch 2a vornehmen.

    Hinweis: Dies ist aufgrund der Beschränkung in Schritt 9 nur eine der beiden Lösungen.





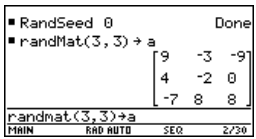
Untersuchung einer Matrix

Das folgende Beispiel illustriert Matrixbefehle.

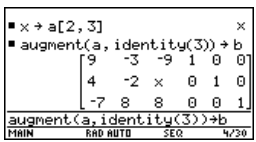
Untersuchung einer 3x3-Matrix

Führen Sie die nachfolgenden Schritte aus, um eine Zufallsmatrix zu erzeugen und ihre Inverse zu bestimmen, falls sie existiert.

1. Benutzen Sie auf dem Hauptbildschirm **RandSeed**, um den Zufallszahlengenerator auf die Werkseinstellung zurückzusetzen. Erzeugen Sie anschließend mit **randMat( )** eine 3x3-Matrix mit zufälligen Werten, und speichern Sie sie in a.



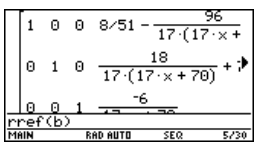
2. Ersetzen Sie das Element **[2,3]** der Matrix durch die Variable x, und benutzen Sie anschließend den Funktion **augment( )**, um die 3x3-Einheitsmatrix anzufügen und das Ergebnis in b zu speichern.



3. Benutzen Sie **rref( )**, um die Matrix b in eine Diagonalform zu bringen:

   Das Ergebnis enthält die Einheitsmatrix in den ersten drei Spalten und a^-1 in den letzten drei Spalten.

   Hinweis: Benutzen Sie den Cursor im Protokoll-Bereich, um das Ergebnis zu scrollen.

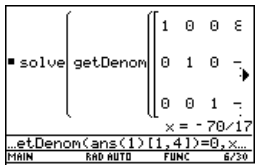


4. Lösen Sie nach dem Wert von x auf, für den die Inverse der Matrix nicht existiert.

   Geben Sie ein:
   solve(getDenom(2 ± [1,4])=0,x)

   Ergebnis: **x=L70/17**

   Hinweis: Benutzen Sie den Cursor im Protokoll-Bereich, um das Ergebnis zu scrollen.



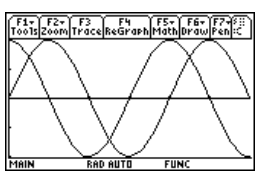
Untersuchung von cos(x) = sin(x)

Im folgenden Beispiel kommen zwei Möglichkeiten zur Ermittlung der Lösung zum Einsatz, wann für x zwischen 0 und 3π cos(x) = sin(x) gilt.

Verfahren 1: Graphisches Verfahren

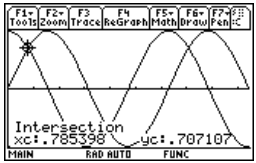
Führen Sie die folgenden Schritte aus, um zu ermitteln, wo sich die Kurven der Funktionen **y**1(x)=cos(x) und y2(x)=sin(x) schneiden.

1. Setzen Sie im Y= Editor **y1(x)=cos(x)** und **y2(x)=sin(x)**.
2. Setzen Sie im Window- Editor **xmin=0** und **xmax=3p**.
3. Drücken Sie [F2], und wählen Sie **A:ZoomFit**.



4. Ermitteln Sie die Schnittpunkte der beiden Funktionen.

   Hinweis: Drücken Sie F5, und wählen Sie **5:Intersection**. Gehen Sie gemäß der Eingabeaufforderungen vor, um die beiden Kurven und die obere und untere Grenze für den Schnittpunkt A auszuwählen.



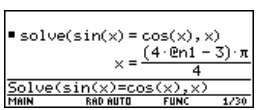
5. Notieren Sie sich die x- und die y-Koordinate. (Wiederholen Sie die Schritte 4 und 5, um weitere Schnittpunkte zu finden.)

Verfahren 2: Symbolische Berechnung

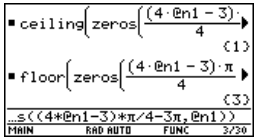
Führen Sie die folgenden Schritte aus, um die Gleichung sin(x)=cos(x) nach x zu lösen.

1. Geben Sie auf dem Hauptbildschirm **solve(sin(x)= cos(x),x)** ein.

   Die Lösungen für x liegen dort, wo @n1 eine ganze Zahl ist.

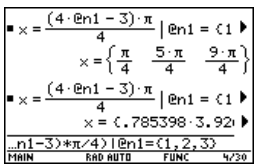


2. Ermitteln Sie wie abgebildet mit den Funktionen **ceiling( )** und **floor( )** für die Schnittpunkte die erste ganze Zahl, die größer und die erste, die kleiner als der gesuchte Wert ist.



   Hinweis: Bringen Sie den Cursor in den Protokoll-Bereich, um die letzte Antwort zu markieren. Drücken Sie ENTER, um das Ergebnis der allgemeinen Lösung zu kopieren.

3. Geben Sie wie abgebildet die allgemeine Gleichung für x ein, und wenden Sie die Beschränkung für **@n1** an.



   Vergleichen Sie das Ergebnis mit Verfahren 1.

   Hinweis: Der Operator “with” wird wie folgt erzeugt: [ctrl] [|]; [doc] [2nd] [']].

Ermitteln der kleinsten Oberfläche eines Quaders

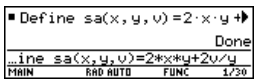
Im folgenden Beispiel wird die Bestimmung der kleinsten Oberfläche eines Parallelepiped mit einem konstanten Volumen von V beschrieben. Nähere Erläuterungen zu den Schritten dieses Beispiels finden Sie in *Symbolisches Rechnen* und *3D-Darstellungen*.

Untersuchung des 3D-Graphs der Quaderoberfläche

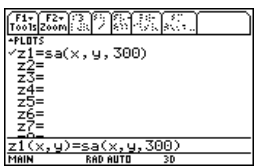
Führen Sie die folgenden Schritte aus, um eine Funktion für die Oberfläche eines Quaders zu definieren, um eine 3D-Graphik zu zeichnen und mit dem **Trace**-Tool einen Punkt nahe der kleinstmöglichen Oberfläche zu ermitteln.

1. Definieren Sie auf dem Hauptbildschirm die Funktion sa(x,y,v) zur Berechnung der Oberfläche eines Quaders.

   Geben Sie ein: define **define sa(x,y,v)=2∗x∗y + 2v/x+2v/y**

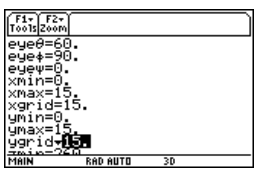


2. Wählen Sie den Graphikmodus 3D-Graph. Geben Sie dann die Funktion für **z1(x,y)** wie hier gezeigt mit dem Volumen **v=300** ein.

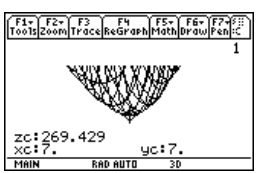


3. Setzen sie die Fenstervariablen auf:

   eye= [60,90,0]
   x= [0,15,15]
   y= [0,15,15]
   z= [260,300]
   ncontour= [5]



4. Zeichnen Sie die Funktion, und benutzen Sie **Trace**, um den Cursor so nahe wie möglich an den kleinsten Wert der Oberflächenberechnungsfunktion zu setzen.

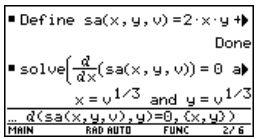


Die kleinste Oberfläche analytisch ermitteln

Sie lösen diese Aufgabe analytisch, indem sie die nachfolgenden Schritte auf dem Hauptbildschirm ausführen.

1. Lösen Sie unter Verwendung von *x* und *y* nach v auf.

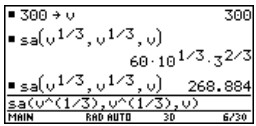
   Geben Sie ein: s**olve(d(sa(x,y,v),x)=0** and **d(sa(x,y,v),y)=0, {x,y})**



2. Ermitteln Sie die kleinste Oberfläche für *v* gleich 300.

   Geben Sie ein: **300→v**
   Geben Sie ein: **sa(v^(1/3), v^(1/3),v)**

   Hinweis: Drücken Sie [ENTER], um das exakte Ergebnis in symbolischer Form zu erhalten. Drücken Sie [◆] [ENTER], um das approximierte Ergebnis in Dezimalform zu erhalten.



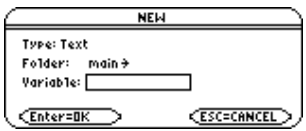
Ein Lernskript mit dem Texteditor ausführen

Im folgenden Beispiel wird der Ablauf eines Beispiel-Skripts mit Hilfe des Text-Editors beschrieben. Nähere Erläuterungen zur Benutzung des Texteditors finden Sie im Modul *Texteditor*.

Ein Lernskript ausführen

Führen Sie die folgenden Schritte aus, um mit dem Texteditor ein Skript zu verfassen, jede Zeile zu testen und die Ergebnisse im Protokoll-Bereich des Hauptbildschirms zu verfolgen.

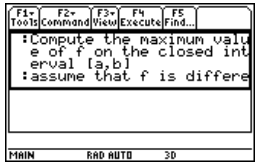
1. Öffnen Sie den Texteditor, und erzeugen Sie eine neue Variable mit dem Namen demo1.



2. Geben Sie im Texteditor folgende Zeilen ein.

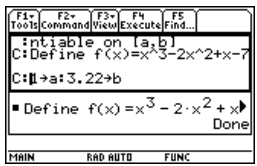
```
Compute the maximum value of f on the closed interval [a,b]
:assume that f is differentiable on [a,b]
:define f(x)=x^3−2x^2+x−7
C: 1→a:3.22→b
C: d(f(x),x)→df(x)
C: zeros(df(x),x)
C: f(ans(1))
C: f({a,b})
: The largest number from the previous two commands is
the maximum value of the function. The smallest number is
the minimum value.
```

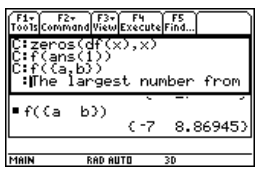
3. Drücken Sie F3, und wählen Sie **1:Script view**, um den Texteditor und den Hauptbildschirm auf einem geteilten Bildschirm anzuzeigen. Bringen Sie den Cursor in die erste Zeile des Texteditors.



4. Drücken Sie wiederholt F4, um jeweils eine Skriptzeile auszuführen.

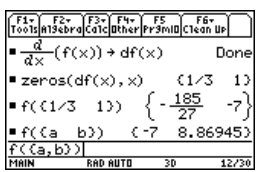
   Hinweis: Drücken Sie F3, und wählen Sie **2:Clear split,** um den Texteditorbildschirm wieder in voller Größe anzuzeigen.





5. Wechseln Sie zum Hauptbildschirm, um sich die Ergebnisse des Skripts auf einem Bildschirm in voller Größe anzeigen zu lassen.

   Hinweis: Drücken Sie zweimal 2nd [QUIT], um den Hauptbildschirm anzuzeigen.



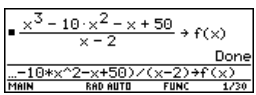
Zerlegung einer rationalen Funktion

Im folgenden Beispiel wird die Zerlegung einer rationalen Funktion in einen Quotienten und einen Rest analysiert. Nähere Erläuterungen zu den Schritten dieses Beispiels finden Sie in *Graphische Darstellung von Funktionen*, und in *Symbolisches Rechnen*.

Eine rationale Funktion zerlegen

Gehen Sie wie folgt vor, um die Zerlegung der rationalen Funktion **f(x)=(x^3−10x^2−x+50)/(x−2)** zu untersuchen:

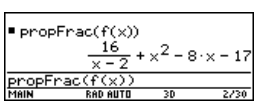
1. Geben Sie auf dem Hauptbildschirm die rationale Funktion wie nachfolgend gezeigt ein, und speichern Sie sie in der Funktion f(x).

   Geben Sie ein:
   (x^3−10x^2−x+50)/(x−2)→f(x)

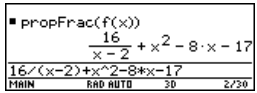
   Hinweis: Die tatsächlichen Eingaben sind in den Beispielbildschirmen invers dargestellt.

2. Benutzen Sie **propFrac**, um die Funktion in Quotienten und Rest zu zerlegen.

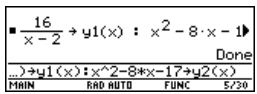
3. Kopieren Sie die letzte Antwort in die Eingabezeile.
   —oder—
   Geben Sie ein: **16/(x−2)+x^2−8∗x−17**

   Hinweis: Bringen Sie den Cursor in den Protokoll-Bereich, um die letzte Antwort zu markieren. Drücken Sie ENTER, um sie in die Eingabezeile zu kopieren.

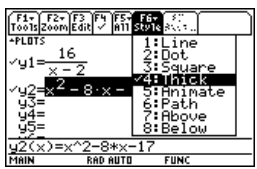


4. Bearbeiten Sie in der Eingabezeile die letzte Antwort. Speichern Sie wie abgebildet den Rest in y1(x) und den Quotienten in y2(x).

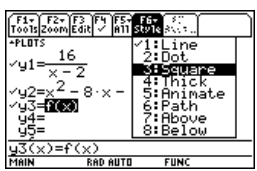
   Geben Sie ein: **16/(x−2)→y1(x): x^2−8∗x−17→y2(x)**



5. Wählen Sie im Y= Editor als Graphzeichenstil thick für **y2(x)**.

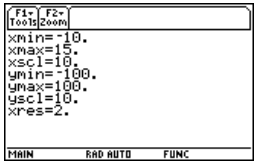


6. Fügen Sie die ursprüngliche Funktion f(x) als **y3(x)** hinzu, und wählen Sie als Graphstil square.



7. Setzen Sie im Window-Editor die Fenstervariablen auf:

   x= [L10,15,10]
   y= [L100,100,10]

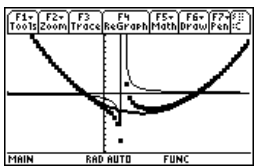


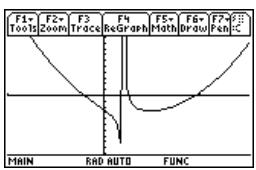
8. Zeichnen Sie die Graphen.

   Hinweis: Stellen Sie sicher, dass der Graphikmodus auf Function eingestellt ist.

Wie Sie sehen, wird das Gesamtverhalten der Funktion f(x) im wesentlichen vom Quotienten y2(x) bestimmt. Der rationale Term ist für sehr große bzw. sehr kleine Werte von *x* im wesentlichen eine quadratische Funktion.

Der untere Graph ist der separat im Line-Stil gezeichnete Graph **y3(x)=f(x)**.





Statistische Untersuchungen: Daten nach Klassen filtern

Das folgende Beispiel bezieht sich auf eine statistische Erfassung des Gewichts von Schülern unter Einbeziehung von Klassen zur Datenfilterung. Nähere Erläuterungen zu den Befehlen dieses Beispiels finden Sie in *Daten/Matrizen-Editor* und *Statistik und Datenplots*.

Daten nach Klassen filtern

Die Schüler sind in eine von acht Klassen aufgeteilt, für die als Kriterien das Geschlecht und die Jahrgangsstufe benutzt wurden. Die Daten (Gewicht in Pfund) und die Klassen werden im Daten/Matrizen-Editor eingegeben.

Tabelle 1: Klassen und ihre Beschreibung

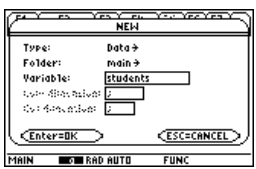
Klasse (C2)	Schuljahr und Geschlecht
1	9. Jgst., Jungen
2	9. Jgst., Mädchen
3	10. Jgst., Jungen
4	10. Jgst., Mädchen
5	11. Jgst., Jungen
6	11. Jgst., Mädchen
7	12. Jgst., Jungen
8	12. Jgst, Mädchen

Tabelle 2: C1 (Gewicht der einzelnen Schüler in Pfund) und ihre C2 (Klasse)

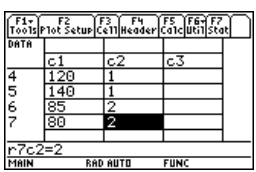
C1	C2	C1	C2	C1	C2	C1	C2
110	1	115	3	130	5	145	7
125	1	135	3	145	5	160	7
105	1	110	3	140	5	165	7
120	1	130	3	145	5	170	7
140	1	150	3	165	5	190	7
85	2	90	4	100	6	110	8
80	2	95	4	105	6	115	8
90	2	85	4	115	6	125	8
80	2	100	4	110	6	120	8
95	2	95	4	120	6	125	8

Führen Sie folgende Schritte aus, um das Gewicht der Schüler mit dem Schuljahr zu vergleichen.

1. Starten Sie den Daten/ Matrizen-Editor, und erstellen Sie eine neue Datenvariable mit dem Namen students.

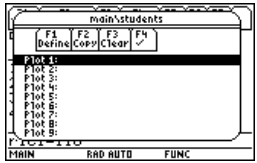


2. Geben Sie die Daten und die Klassen aus Tabelle 2 in die Spalten c1 und c2 ein.

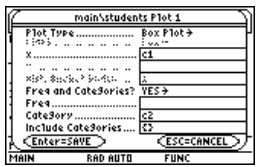


3. Öffnen Sie das Menü F2 **Plot Setup**.

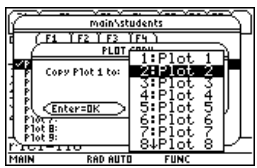
   Hinweis: Richten Sie mehrere Box-Plots ein, um verschiedene Untermengen der Gesamtdatenmenge zu vergleichen.



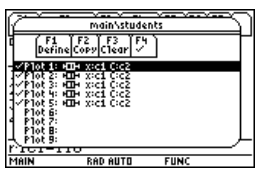
4. Definieren Sie die **Plot-** und Filterparameter für **Plot 1** wie nebenstehend abgebildet.



5. Kopieren Sie **Plot 1** nach **Plot 2**.

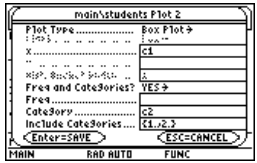


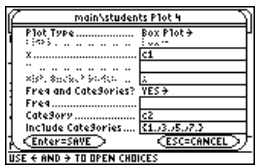
6. Wiederholen Sie Schritt 5, und kopieren Sie **Plot 1** nach **Plot 3**, **Plot 4** und **Plot 5**.



7. Drücken Sie F1, und ändern Sie Include Categories für **Plot 2** bis **Plot 5** wie folgt:

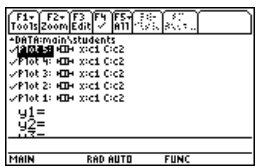
   Plot 2: {1,2}
   (9. Jgst. Jungen, Mädchen)
   Plot 3: {7,8}
   (12. Jgst. Jungen, Mädchen)
   Plot 4: {1,3,5,7}
   (alle Jungen)
   Plot 5: {2,4,6,8}
   (alle Mädchen)



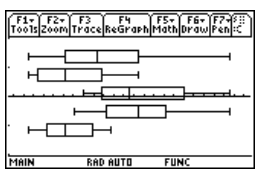


8. Heben Sie im Y= Editor die Auswahl von Funktionen, die unter Umständen in früheren Beispielen aktiviert wurden, auf.

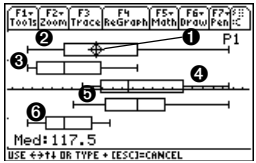
   Hinweis: Nur **Plot 1** bis **Plot 5** dürfen ausgewählt sein.



9. Zeigen Sie die Plots an, indem Sie F2 drücken und **9:Zoomdata** wählen.



10. Benutzen Sie das **Trace**-Tool, um das mittlere Gewicht der Schüler der verschiedenen Teilgruppen zu vergleichen.



❶ Median, alle Schüler
❷ alle Schüler
❸ alle 9.-Klässler
❹ alle 12.-Klässler
❺ alle Jungen
❻ alle Mädchen

CBL 2™-Programm für den TI-89 Titanium / Voyage™ 200

Im folgenden Beispiel wird ein Programm vorgestellt, welches bei Anschluß des TI-89 Titanium / Voyage™ 200 an ein Calculator-Based Laboratory™- (CBL 2™-) Gerät zum Tragen kommt. Das Programm verwendet die Experimente “Newton’s Law of Cooling” (Newtons Abkühlungsgesetz) und ist vergleichbar mit “Coffee To Go” (Kaffee zum Mitnehmen) aus dem Arbeitsbuch CBL System Experiment Workbook. Umfangreichen Text können Sie über die Tastatur Ihres Computers eingeben und dann mit TI Connect™ oder TI-GRAPH LINK™ Software an den TI-89 Titanium / Voyage™ 200 übertragen. Weitere TI-89 Titanium / Voyage™ 200 CBL 2™-Programme sind auf der TI-Website unter folgender Adresse verfügbar education.ti.com.

Programmanweisung	Beschreibung
:cooltemp()	Programmname
:Prgm	

Programmanweisung	Beschreibung
:Local i	Deklaration einer lokalen Variablen.
:setMode("Graph","FUNCTION")	TI-89 Titanium / Voyage™ 200 für graphische Darstellung einer Funktion einrichten.
:PlotsOff	Alle vorherigen Plots ausschalten.
:FnOff	Alle vorherigen Funktionen ausschalten.
:ClrDraw	Alle vorherigen Zeichnungen im Graphikildschirm löschen.
:ClrGraph	Alle vorherigen Graphen löschen.
:ClrIO	I/O-Bildschirm des TI-89 Titanium / Voyage™ 200 löschen.
:⁻10→xmin:99→xmax:10→xscl	Die Fenstervariablen einrichten.
:⁻20→ymin:100→ymax:10→yscl	
:{0}→data	Liste mit dem Namen data (Daten) erzeugen bzw. initialisieren.
:{0}→time	Liste mit dem Namen time (Zeit) erzeugen bzw. initialisieren.
:Send{1,0}	Befehl zum Löschen der CBL 2™/CBL™-Einheit senden.
:Send{1,2,1}	Kan. 2 des CBL 2™/CBL™ auf AutoID für Temperaturerfassung setzen.
:Disp "Press ENTER to start"	Benutzer zum Drücken von [ENTER] auffordern.

Programmanweisung	Beschreibung
:Disp "graphingTemperature."	
:Pause	Warten bis Benutzer bereit zum Anfangen ist.
:PtText "TEMP(C)",2,99	y-Achse des Graphen beschriften.
:PtText "T(S)",80,⁻5	x-Achse des Graphen beschriften.
:Send{3,1,⁻1,0}	Trigger Befehl an CBL 2™ senden; Daten in Echtzeit erfassen.
:For i,1,99	Die beiden nachfolgenden Anweisungen für 99 Temperaturerfassungen wiederholen.
:Get data[i]	Eine Temperatur von CBL 2™/CBL™ einlesen und in Liste speichern.
:PtOn i,data[i]	Graph der Temperaturdaten erzeugen.
:EndFor	
:seq(i,i,1,99,1)→time	Eine Liste für time (Zeit) oder Datensatz-Nummer erzeugen.
:NewPlot 1,1,time,data,,,,4	time (Zeit) und data (Daten) mit NewPlot und Trace-Tool erzeugen.
:DispG	Graph anzeigen.
:PtText "TEMP(C)",2,99	Achsen neu beschriften.
:PtText "T(S)",80,⁻5	
:EndPrgm	Programm beenden.

Außerdem können Sie mit dem Calculator-Based Ranger™ (CBR™) die mathematischen und wissenschaftlichen Beziehungen zwischen Entfernung, Geschwindigkeit, Beschleunigung und Zeit für die bei Ihrer Tätigkeit gesammelten Daten untersuchen.

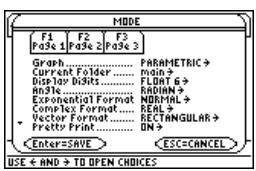
Untersuchen der Bahn eines fliegenden Baseballs

Dieses Beispiel verwendet einen geteilten Bildschirm, um gleichzeitig den parametrischen Graph und die Wertetabelle beim Flug eines Baseballs anzuzeigen.

Eine Parameterdarstellung und eine Tabelle einrichten

Führen Sie folgende Schritte aus, um die Bahn eines Balls zu verfolgen, wobei dieser eine Anfangsgeschwindigkeit von 95 Fuß pro Sekunde und einen Abwurfwinkel von 32 Grad besitzt.

1. Richten Sie die Modi für **Page 1** wie nebenstehend abgebildet ein.

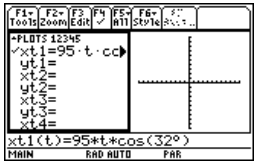


2. Richten Sie die Modi für **Page 2** wie nebenstehend abgebildet ein.

3. Geben Sie auf der linken Seite im **Y= Editor** die Gleichung für die Flugweite des Balls zum Zeitpunkt t für **xt1(t)** ein.

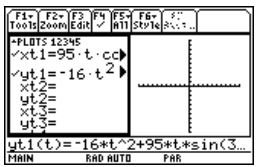
   xt1(t)=95∗t∗cos(32°)

   Hinweis: Das Grad-Symbol erhalten Sie durch Drücken von [2nd] [°].



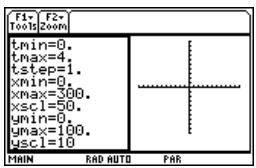
4. Geben Sie im **Y= Editor** die Gleichung für die Höhe des Balls zum Zeitpunkt t für **yt1(t)** ein.

   yt1(t)=⁻16∗t^2+95∗t∗sin(32°)



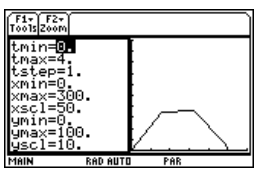
5. Setzen Sie die Fenstervariablen auf:

   t values= [0,4,.1]
   x values= [0,300,50]
   y values= [0,100,10]



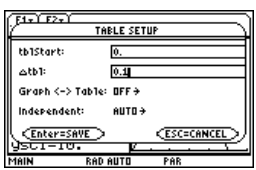
6. Wechseln Sie auf die rechte Bildschirmseite, und zeigen Sie den Graph an.

   Hinweis: Drücken Sie [2nd] [⇆].



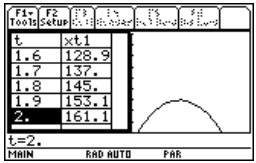
7. Rufen Sie das Dialogfeld **TABLE SETUP** auf, und ändern Sie **tblStart** auf **0** und **@tbl** auf **0.1**.

   Hinweis: Drücken Sie [♦] [TBLSET].



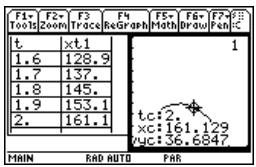
8. Zeigen Sie die Tabelle im linken Teil an, und drücken Sie ⊙, um t=2 zu markieren.

   Hinweis: Drücken Sie ♦ [TABLE].



9. Wechseln Sie zum rechten Bildschirmteil. Drücken Sie F3, und tracen Sie den Graph, um den Wert von xc und yc bei tc=2 zu ermitteln.

   Hinweis: Wenn Sie den Trace-Cursor von tc=0.0 nach tc=3.1 verschieben, sehen Sie die Ballposition zum Zeitpunkt tc.



Optionale Übung

Nehmen Sie die gleiche Ausgangsgeschwindigkeit von 95 Fuß pro Sekunde an, und ermitteln Sie den Winkel, den der Ball besitzen müßte, damit er die größte Entfernung zurücklegt.

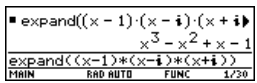
Komplexe Nullstellen eines kubischen Polynoms graphisch darstellen

Im folgenden Beispiel wird die grafische Umsetzung zum Auffinden der Nullstellen eines Polynoms dritten Grads beschrieben. Nähere Erläuterungen zu den Schritten dieses Beispiels finden Sie in *Symbolisches Rechnen* und *3D-Darstellungen*.

Komplexe Wurzeln graphisch darstellen

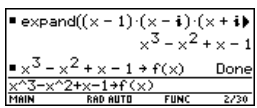
Führen Sie die folgenden Schritte aus, um das kubische Polynom (x−1)(x−i)(x+i) zu entwickeln, den absoluten Wert der Funktion zu berechnen sowie die daraus resultierende Fläche graphisch darzustellen. Benutzen Sie das **Trace**-Tool zum Untersuchen dieser Fläche.

1. Entwickeln Sie auf dem Hauptbildschirm mit dem Befehl **expand( )** das Polynom **(x−1)(x−*i*)(x+*i*)**, um das erste Polynom zu erhalten.

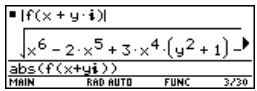


2. Kopieren Sie die letzte Antwort, fügen Sie sie in die Eingabezeile ein, und legen Sie sie als Funktion **f(x)** ab.

   Hinweis: Markieren Sie das letzte Ergebnis im Protokollbereich, und kopieren Sie es mit ENTER in die Eingabezeile.



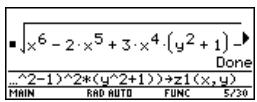
3. Ermitteln Sie den absoluten Wert von f(x+y*i*) mit der Funktion **abs( )**.



(Diese Berechnung kann ca. 2 Minuten dauern.)

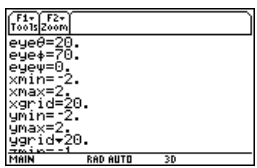
Hinweis: Der absolute Wert einer Funktion bewirkt, daß die Nullstellen in der graphischen Darstellung die x-Achse gerade berühren. Entsprechend bedingt der Absolutwert einer Funktion mit zwei Variablen, daß die Nullstellen in der graphischen Darstellung die xy-Ebene gerade berühren.

4. Kopieren Sie die letzte Antwort, fügen Sie sie in die Eingabezeile ein, und legen Sie sie als Funktion **z1(x,y)** ab.



Hinweis: Der Graph von z1(x,y) ist die resultierende Fläche.

5. Stellen Sie das Gerät auf 3D-Graphikmodus, aktivieren Sie für Graph-Format die Achsen, und setzen Sie die Fenstervariablen auf:



eye= [20,70,0]
x= [L2,2,20]
y= [L2,2,20]
z= [L1,2]
ncontour= [5]

6. Drücken Sie im **Y= Editor** folgende Tasten:

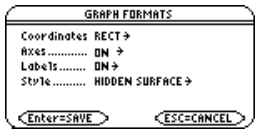
   [♦] [|]

   [♦] **F**

   und wählen Sie für das Graphik-Format folgende Einstellungen:

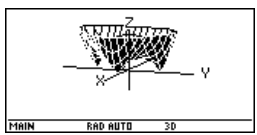
   Axes= ON
   Labels= ON
   Style= HIDDEN SURFACE

   Hinweis: Das Berechnen und Zeichnen des Graphs dauert etwa drei Minuten.

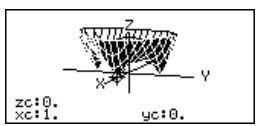


7. Erstellen Sie den Graph der Fläche.

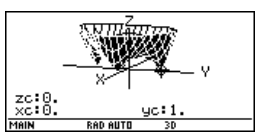
   Der 3D-Graph dient dazu, die Punkte darzustellen, an denen die Fläche die xy-Ebene berührt.



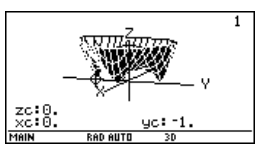
8. Untersuchen Sie mit dem Trace-Tool die Funktionswerte bei **x=1** und **y=0**.



9. Untersuchen Sie mit dem Trace-Tool die Funktionswerte bei **x=0** und **y=1**.



10. Untersuchen Sie mit dem Trace-Tool die Funktionswerte bei **x=0** und **y=L1**.



Zusammenfassung

Sie sehen, dass zc für jeden der Werte in den Schritten 8 bis 10 null ist. Die komplexen Nullstellen 1,⁻*i*, *i* des Polynoms x3−x2+x−1 können demnach visuell als die drei Punkte, an denen die Fläche die xy-Ebene berührt, dargestellt werden.

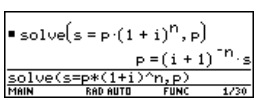
Berechnen einer Zeitrente

Das folgende Beispiel beschäftigt sich mit der Ermittlung von Zinssatz, Darlehensbetrag, Laufzeit und Zukunftswert einer Rente.

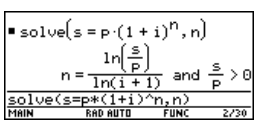
Den Zinssatz einer Jahresrate ermitteln

Führen Sie folgende Schritte aus, um den Zinssatz (i) einer Jahresrate zu berechnen, wobei das Anfangskapital (p) 1000 beträgt, die Anzahl der Zinsperioden (Jahre) (n) gleich 6 ist und der Endwert (s) sich auf 2000 beläuft.

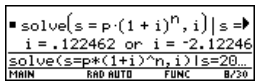
1. Geben Sie auf dem Hauptbildschirm die Gleichung zur Lösung nach p ein.



2. Geben Sie die Gleichung zur Lösung nach n ein.



3. Geben Sie die Gleichung zur Lösung nach i mit dem “with”-Operator ein.



solve(s=p∗(1+i)^n,i) | s=2000 und **p=1000** und **n=6**

Ergebnis: Der Zinssatz beträgt 12,246%.

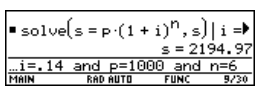
Hinweis:

- Drücken Sie [|], um den Operator “with” (|) einzugeben.
  [|]; [2nd] ['].
- Drücken Sie [♦] [ENTER], um ein Gleitkommaergebnis zu erhalten.

Den Endwert einer Jahresrente berechnen

Berechnen Sie den Endwert einer Jahresrente; verwenden Sie die Werte des vorigen Beispiels und einen Zinssatz von 14%.

Geben Sie die Gleichung zur Lösung nach **s** ein.



solve(s=p∗(1+i)^n,s) | i=.14 und **p=1000** und **n=6**

Ergebnis: Der Endwert bei einem Zinssatz von 14% beträgt 2194,97.

Berechnen des Zeitwerts eines Geldbetrags

In diesem Beispiel wird eine Funktion erstellt, die u.a. zur Berechnung der Finanzierungskosten eines PKW benutzt werden kann. Nähere Erläuterungen zu den Schritten dieses Beispiels finden Sie in *Programmierung*.

Die Funktion für den Zeitwert eines Geldbetrags

Definieren Sie im Programmeditor die nachfolgende Funktion **tvm** für den Zeitwert eines Betrags (engl. Time-Value-of-Money, daher **tvm**), wobei **t**emp1 = Anzahl der Raten, temp2 = Jahreszinssatz, temp3 = gegenwärtiger Wert, temp4 = Höhe einer Monatsrate, temp5 =Endwert und temp6 =vorschüssig oder nachschüssig (1 = Monatsbeginn, also vorschüssig, 0 = Monatsende, d.h. nachschüssig).

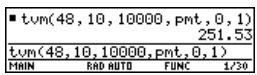
```
:tvm(temp1,temp2,temp3,temp4,temp5,temp6)
:Func
:Local tempi,tempfunc,tempstr1
:Ltemp3+(1+temp2/1200ùtemp6)ùtemp4ù((1N(1+temp2/1200)^
    (Ltemp1))/(temp2/1200))Ntemp5ù(1+temp2/1200)^(Ltemp1)
    !tempfunc
:For tempi,1,5,1
:"temp"&exact(string(tempi))!tempstr1
:If when(#tempstr1=0,false,false,true) Then
:If tempi=2
:Return approx(nsolve(tempfunc=0,#tempstr1) | #tempstr1>0 and
    #tempstr1<100)
:Return approx(nsolve(tempfunc=0,#tempstr1))
:EndIf
:EndFor
:Return "parameter error"
:EndFunc
```

Hinweis: Umfangreichen Text können Sie über die Tastatur Ihres Computers eingeben und dann mit TI Connect™ oder TI-GRAPH LINK™ Software an den TI-89 Titanium / Voyage™ 200 übertragen.

Die Höhe der Monatsrate ermitteln

Berechnen Sie die Höhe der Monatsrate für einen gebrauchten PKW im Wert von DM 10.000, für den Sie 48 Raten bei einem Jahreszinssatz von 10% zahlen.

Geben Sie auf dem Hauptbildschirm die Werte für **tvm** ein, die zur Berechnung von pmt notwendig sind.

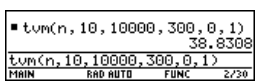


Ergebnis: Die Monatsrate beträgt 251,53.

Die Anzahl der Raten ermitteln

Berechnen Sie die Anzahl der Raten, die notwendig wären, wenn Sie monatlich DM 300 für diesen Wagen abzahlen könnten.

Geben Sie auf dem Hauptbildschirm die Werte für **tvm** ein, die zur Berechnung von n notwendig sind.



Ergebnis: Die Anzahl der Raten beträgt 38,8308.

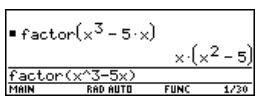
Ermitteln rationaler, reeller und komplexer Faktoren

In diesem Beispiel wird gezeigt, wie Sie rationale, reelle oder komplexe Faktoren eines Terms ermitteln. Nähere Erläuterungen zu den Schritten dieses Beispiels finden Sie in *Symbolisches Rechnen*.

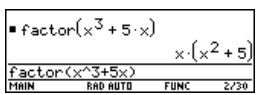
Faktoren ermitteln

Geben Sie die nachfolgend abgebildeten Terme auf dem Hauptbildschirm ein.

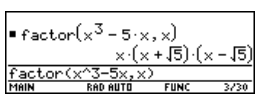
1. **factor(x^3−5x)** ENTER ergibt eine rationale Zerlegung.



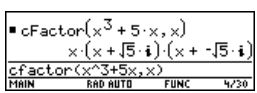
2. **factor(x^3+5x)** ENTER ergibt ebenfalls eine rationale Zerlegung.



3. **factor(x^3−5x,x)** ENTER ergibt eine reelle Zerlegung.



4. **cfactor(x^3+5x,x)** ENTER ergibt eine komplexe Zerlegung.



Simulation einer Stichprobenentnahme ohne Zurücklegen

Dieses Beispiel zeigt, wie man das Ziehen verschieden farbiger Kugeln ohne Zurücklegen aus einer Urne simulieren kann. Nähere Erläuterungen zu den Schritten dieses Beispiels finden Sie in *Programmieren*.

Eine Funktion für Stichprobenentnahme ohne Zurücklegen

Definieren Sie im Programmeditor die Funktion **drawball( )** für zwei Aufrufparameter. Der erste Parameter ist eine Liste, in der jedes Element die Anzahl der Kugeln einer bestimmten Farbe ist. Der zweite Parameter ist die Anzahl der auszuwählenden (zu ziehenden) Kugeln. Die Funktion gibt eine Liste zurück, in der jedes Element die Anzahl der für jede Farbe gezogenen Kugeln wiedergibt.

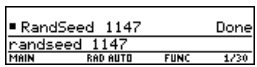
```
:drawball(urnlist,drawnum)
:Func
:Local templist,drawlist,colordim,
    numballs,i,pick,urncum,j
:If drawnum>sum(urnlist)
:Return "too few balls"
:dim(urnlist)→colordim
:urnlist→templist
:newlist(colordim)→drawlist
:For i,1,drawnum,1
:sum(templist)→numballs
:rand(numballs)→pick
(Fortsetzung in der nächsten Spalte)
```

```
:For j,1,colordim,1
:cumSum(templist)→urncum
:If pick ≤ urncum[j] Then
:drawlist[j]+1→drawlist[j]
:templist[j]−1→templist[j]
:Exit
:EndIf
:EndFor
:EndFor
:Return drawlist
:EndFunc
```

Ausführen einer Stichprobenentnahme

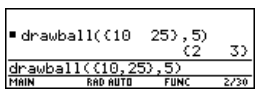
Nehmen Sie an, ein Gefäß enthält n1 Kugeln einer ersten Farbe, n2 Kugeln einer zweiten Farbe, n3 Kugeln einer dritten Farbe usw. Simulieren Sie das Ziehen von Kugeln ohne Zurücklegen.

1. Geben Sie mit dem Befehl **RandSeed** eine Ausgangsbasis für die Zufallszahlgenerierung ein.



2. Nehmen Sie an, das Gefäß enthält 10 rote und 25 weiße Kugeln , und simulieren Sie die Entnahme von 5 Kugeln ohne Zurücklegen. Geben Sie ein: **drawball({10,25},5)**.

   Ergebnis: 2 rote und drei 3 weiße Kugeln.



Rechnerkopplung

Verbinden von zwei Geräten

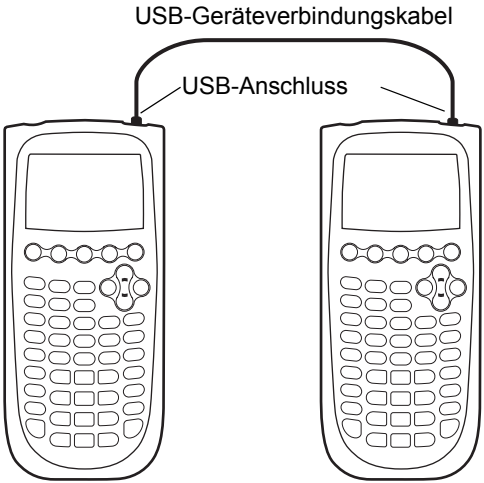
Der TI-89 Titanium und Voyage™ 200 Graphischer Rechner werden jeweils mit einem Kabel ausgeliefert, mit dem zwei Geräte verbunden werden können. Nach erfolgter Verbindung können Sie zwischen den Geräten Informationen übertragen. Dem TI-89 Titanium liegt ein USB-Geräteverbindungskabel bei. Verwenden Sie mit diesem Kabel den USB-Anschluss des Geräts. Dem Voyage™ 200 liegt ein Standard-Geräteverbindungskabel bei. Verwenden Sie mit diesem Kabel den E/A-Anschluss des Geräts.

Hinweis: Der TI-89 Titanium besitzt sowohl einen USB- als auch einen E/A-Anschluss. Sie können ihn also mit TI Graphik-Handhelds mit beiden Anschlusstypen verbinden. Wenn Sie den E/A-Anschluss verwenden möchten, wird jedoch das Standard-Geräteverbindungskabel (separat erhältlich) oder das TI-USB-Rechnerkopplungskabel (auch separat erhältlich) benötigt, das zum Anschließen an einen Computer verwendet wird.

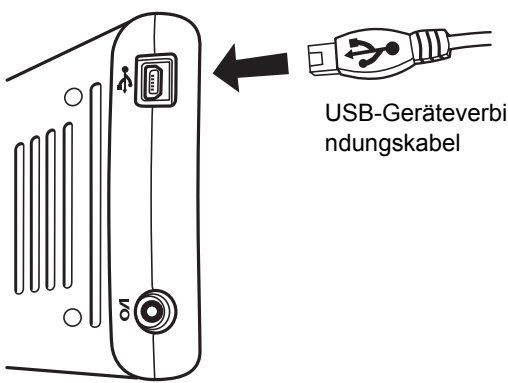
Anschließen vor dem Senden oder Empfangen

Stecken Sie je ein Kabelende mit festem Druck in die Verbindungsanschlüsse der beiden Geräte. Jedes Gerät kann senden und empfangen, abhängig davon, welche Optionen Sie im **VAR-LINK** Bildschirm auswählen.

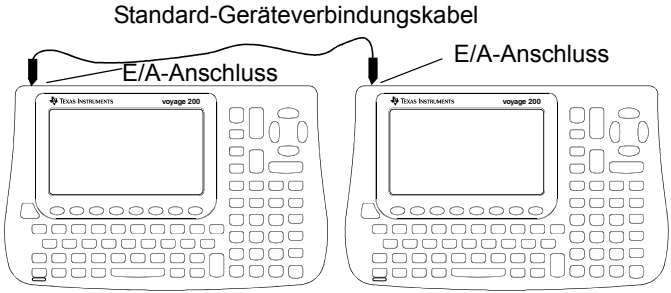
Sie können einen TI-89 Titanium oder Voyage™ 200 mit einem anderen TI-89 Titanium, Voyage™ 200, TI-89 oder TI-92 Plus verbinden.



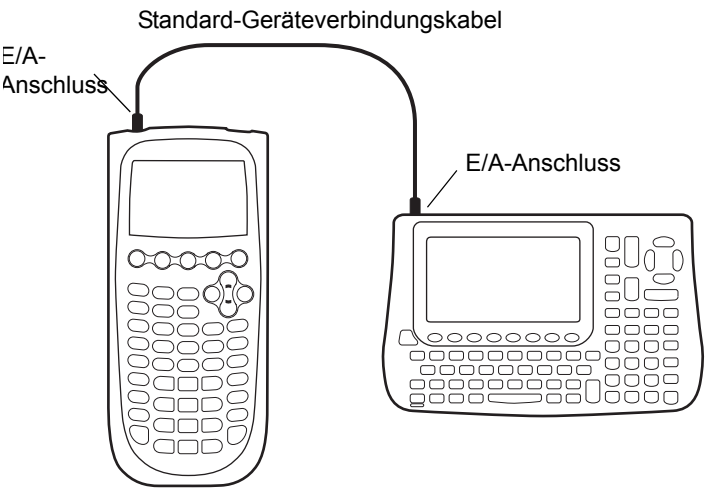
Zwei verbundene TI-89 Titanium



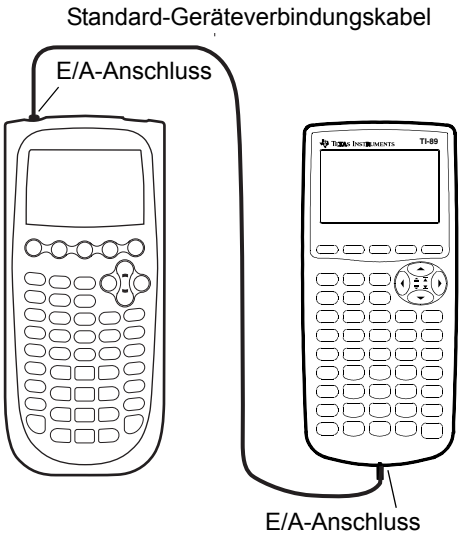
Positionieren Sie das Gerät so, dass sich die beiden USB-Symbole gegenüber liegen; dann stecken Sie den Anschluß ein.



Zwei verbundene Voyage™ 200



Ein TI-89 Titanium und ein Voyage™ 200 in verbundenem Zustand



Ein TI-89 Titanium und ein TI-89 in verbundenem Zustand

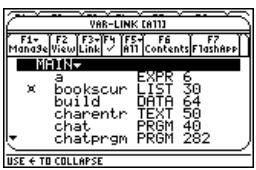
Übertragen von Variablen, Flash-Anwendungen und Ordnern

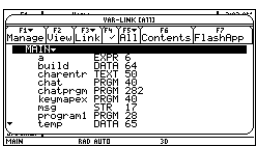
Das Übertragen von Variablen ist ein bequemer Weg, um alle Variablen, die auf dem **VAR-LINK** Bildschirm aufgeführt sind, wie Funktionen, Programme usw., auch auf einem anderen Gerät zu verwenden. Sie können auch Flash-Anwendungen (Apps) und Ordner übertragen.

Vorbereiten der Geräte

Flash-Anwendungen sind nur zwischen bestimmten Geräten übertragbar. Sie können eine App beispielsweise von einem TI-89 Titanium auf einen anderen TI-89 Titanium oder von einem TI-89 Titanium auf einen TI-89 übertragen. Sie können eine App von einem Voyage™ 200 auf einen anderen Voyage™ 200 oder von einem Voyage™ 200 auf einen TI-92 Plus übertragen.

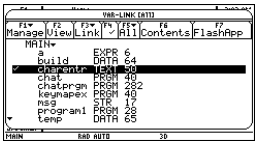
1. Verbinden Sie zwei Graphik-Handhelds mit dem geeigneten Kabel.
2. Drücken Sie auf dem *sendenden* Gerät [2nd] [VAR-LINK], um den **VAR-LINK** Bildschirm anzuzeigen.



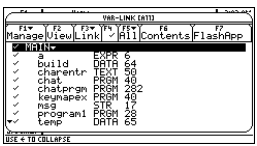


3. Wählen Sie auf dem *sendenden* Gerät die Variablen, Ordner oder Flash-Anwendungen aus, die Sie senden wollen.

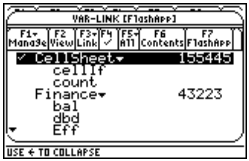
- Um eine einzelne Variablen, Flash-Anwendung oder einen Ordner auszuwählen, bewegen Sie den Cursor, um ihn hervorzuheben und drücken Sie dann F4, um daneben ein Häkchen (✓) zu platzieren.

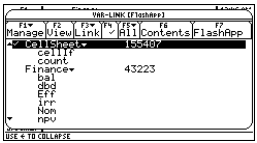


  - Auf dem standardmäßigen **VAR-LINK** Bildschirm wird dadurch der Ordner und sein Inhalt ausgewählt. Ausgeblendet angezeigte Ordner werden erweitert angezeigt, nachdem sie ausgewählt wurden.

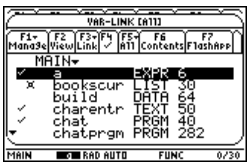


- Wenn (aus der Registerkarte F7) eine Flash-Anwendung ausgewählt wurde, wird dadurch der Anwendungsordner mit Inhalt ausgewählt. Neben dem Ordner wird ein Häkchen eingeblendet, jedoch nicht neben den Inhalten. Geschlossene Flash-Anwendungsordner werden jedoch nicht automatisch geöffnet.

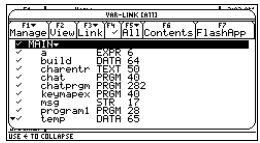




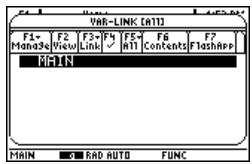
- To select multiple variables, Flash applications, or folders, highlight each one and press F4 to place a checkmark (✓) beside it.«Um mehrere Variablen, Flash-Anwendungen oder Ordner auszuwählen, markieren Sie die gewünschten, und drücken Sie dann F4, um daneben ein Häkchen (✓) einzufügen.« Benutzen Sie bei Bedarf erneut F4, um ein ausgewähltes Element, das nicht übertragen werden soll, wieder abzuwählen.

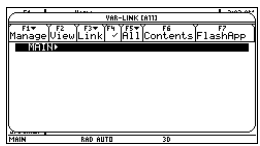


- Um alle Variablen, Flash-Anwendungen oder Ordner auszuwählen, verwenden Sie [F5] **All 1:Select All**.

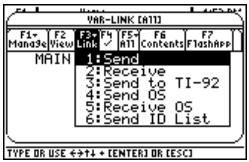


4. Drücken Sie auf dem *empfangenden* Gerät [2nd] [VAR-LINK], um den **VAR-LINK** Bildschirm anzuzeigen. (Auf dem sendenden Gerät bleibt die Anzeige des **VAR-LINK** Bildschirms bestehen.)





5. Drücken Sie auf dem *empfangenden* und auf dem sendenden Gerät F3 **Link**, um die Menüoptionen aufzurufen.



6. Wählen Sie auf dem *empfangenden* Gerät **2:Receive**.

   In der Statuszeile des empfangenden Geräts werden die Meldung **VAR-LINK: WAITING TO RECEIVE** und die Anzeige **BUSY** angezeigt

7. Wählen Sie auf dem *sendenden* Gerät **1:Send**

   Dadurch wird die Übertragung gestartet.

Während der Übertragung wird in der Statuszeile des empfangenden Geräts ein Fortschrittsbalken angezeigt.. Nach Abschluss der Übertragung wird der **VAR-LINK** Bildschirm des empfangenden Geräts aktualisiert.

Hinweis: Vor der Übertragung einer erworbenen App muss das empfangenden Gerät, falls erforderlich, über das entsprechende Zertifikat verfügen. Ein Zertifikat ist eine Datei, die von TI erstellt wird. Kostenlose Apps und Konzept-Apps benötigen keine Zertifikate.

Regeln für die Übertragung von Variablen, Flash-Anwendungen und Ordnern

Ungesperrte und nicht archivierte Variablen, die auf dem sendenden und dem empfangenden Gerät den selben Namen haben, werden vom sendenden Gerät überschrieben.

Gesperrte Variablen, die auf dem sendenden und dem empfangenden Gerät den selben Namen haben, müssen auf dem empfangenden Gerät entsperrt werden, bevor sie vom sendenden Gerät überschrieben werden können. Wenn archivierte Variablen auf dem sendenden und dem empfangenden Gerät die selben Namen haben, wird eine Meldung eingeblendet, in der Sie um Bestätigung gebeten werden, damit die Variablen überschrieben werden können.

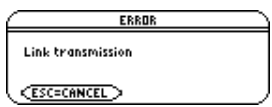
Bei Auswahl von...	geschieht folgendes:
Ungesperrte Variable	Die Variable wird in den aktuellen Ordner übertragen und ist dann auch auf dem empfangenden Gerät nicht gesperrt.
Gesperrte Variable	Die Variable wird in den aktuellen Ordner übertragen und ist dann auch auf dem empfangenden Gerät gesperrt.
Archivierte Variable	Die Variable wird in den aktuellen Ordner übertragen und ist dann auch auf dem empfangenden Gerät archiviert.
Ungesperrte Flash-Anwendung	Wenn das empfangende Gerät über das korrekte Zertifikat verfügt, wird die Flash-Anwendung übertragen. Sie ist anschließend auch auf dem empfangenden Gerät nicht gesperrt.

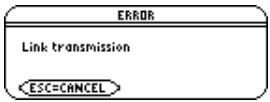
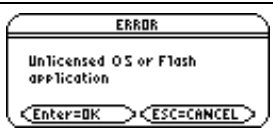
Bei Auswahl von...	geschieht folgendes:
Gesperrte Flash-Anwendung	Wenn das empfangende Gerät über das korrekte Zertifikat verfügt, wird die Flash-Anwendung übertragen. Sie ist anschließend auch auf dem empfangenden Gerät gesperrt.
Ungesperrter Ordner	Der Ordner und sein ausgewählter Inhalt werden übertragen. Der Ordner ist anschließend auch auf dem empfangenden Gerät nicht gesperrt.
Gesperrter Ordner	Der Ordner und sein ausgewählter Inhalt werden übertragen. Der Ordner wird auf dem empfangenden Gerät nicht gesperrt.

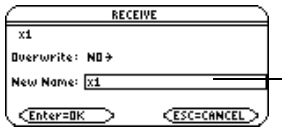
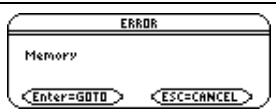
Abbrechen einer Übertragung

Am sendenden oder empfangenden Gerät:

1. Drücken Sie [ON].
   Eine Fehlermeldung wird eingeblendet.
2. Drücken Sie [ESC] oder [ENTER].



Anzeige auf:	Meldung und Bedeutung:
Sendendes Gerät	 Dies wird nach mehreren Sekunden angezeigt, wenn: • Am Verbindungsanschluss des sendenden Geräts kein Kabel angeschlossen ist. – oder – • Am anderen Ende des Kabels ist kein empfangendes Gerät angeschlossen. – oder – • Das empfangende Gerät ist nicht für den Empfang vorbereitet. Drücken Sie [ESC] oder [ENTER], um die Übertragung abzubrechen. **Hinweis:** Das sendende Gerät zeigt diese Meldung eventuell nicht immer an. Stattdessen kann solange **BUSY** gemeldet werden, bis Sie die Übertragung abbrechen.
Sendendes Gerät	 Das empfangende Gerät hat nicht das richtige Zertifikat für das Betriebssystem (BS) oder die Flash-Anwendung, die gesendet werden.

Anzeige auf:	Meldung und Bedeutung:
Empfangendes Gerät	 New Name ist nur aktiv, wen Sie Overwrite auf NO ändern Das empfangende Gerät besitzt eine Variable mit dem selben Namen wie die Variable, die gesendet wird. • Um die vorhandene Variable zu überschreiben, drücken Sie ENTER. (Standardmäßig gilt **Overwrite = YES**.) • Um die Variable unter einem anderen Namen zu speichern, setzen Sie **Overwrite = NO**. Geben Sie im Eingabefeld **New Name** einen Variablennamen an, der auf dem empfangenden Gerät noch nicht verwendet wird. Drücken Sie anschließend zweimal ENTER. • Um die Variable zu übergehen und mit der nächsten fortzusetzen, setzen Sie **Overwrite = SKIP**, und drücken Sie ENTER. • Drücken Sie ESC, um die Übertragung abzubrechen.
Empfangendes Gerät	 Das empfangende Gerät hat nicht genügend Speicher für die gesendeten Daten. Drücken Sie ESC oder ENTER, um die Übertragung abzubrechen.

Löschen von Variablen, Flash-Anwendungen oder Ordnern

1. Drücken Sie 2nd [VAR-LINK], um den **VAR-LINK** Bildschirm anzuzeigen.
2. Wählen Sie die Variablen, Ordner oder Flash-Anwendungen aus, die gelöscht werden sollen.
   - Um eine einzelne Variable, Flash-Anwendung oder einen Ordner auszuwählen, bewegen Sie den Cursor, um das Element hervorzuheben und drücken Sie dann F4, um daneben ein Häkchen (✓) zu platzieren.
     - Auf dem standardmäßigen **VAR-LINK** Bildschirm wird dadurch der Ordner und sein Inhalt ausgewählt. Ausgeblendet dargestellte Ordner werden erweitert angezeigt, nachdem sie ausgewählt wurden.
     - Wenn (aus der Registerkarte F7) eine Flash-Anwendung ausgewählt wurde, wird dadurch der App-Ordner mit Inhalt ausgewählt. Neben dem Ordner wird ein Häkchen eingeblendet, nicht jedoch neben dem Ordnerinhalt. Die Ansicht von ausgeblendet dargestellten Flash-Anwendungs-Ordnern wird nicht automatisch erweitert.

     Hinweis: Der Ordner **Main** kann nicht gelöscht werden.
   - Um mehrere Variablen, Flash-Anwendungen oder Ordner auszuwählen, markieren Sie die gewünschten, und drücken Sie dann F4, um daneben ein Häkchen (✓) einzufügen. Benutzen Sie bei Bedarf erneut F4, um ein ausgewähltes Element, das nicht gelöscht werden soll, wieder abzuwählen.
   - Um alle Variablen, Flash-Anwendungen oder Ordner auszuwählen, verwenden Sie F5 **All 1:Select All**.
3. Drücken Sie F1, und wählen Sie **1:Delete**.
   – oder –
   Drücken Sie ←. Eine Bestätigungsaufforderung wird eingeblendet.
4. Drücken Sie ENTER, um den Löschbefehl zu bestätigen.

Wo man Flash-Anwendungen (Apps) bekommt

Aktuelle Information über Flash-Anwendungen finden Sie auf der Texas Instruments Website unter education.ti.com oder kontaktieren Sie Texas Instruments über TI-Cares™.

Viele Apps benötigen kein Zertifikat mehr. Wenn Sie versuchen, eine App von einem Gerät auf ein anderes zu übertragen, und die Meldung **Unlicensed OS or Flash application** wird eingeblendet, versuchen Sie, die App erneut von der Texas Instruments Website unter education.ti.com herunterzuladen.

Sie können eine Flash-Anwendung und/oder ein Zertifikat von der Texas Instruments Website auf einen Computer herunterladen und ein USB cable oder TI-USB-Rechnerkopplungskabel verwenden, um die Anwendung oder das Zertifikat auf Ihrem TI-89 Titanium / Voyage™ 200 Graphischer Rechnerzu installieren.

Eine Anleitung zur Installation von Flash-Anwendungen finden Sie unter education.ti.com/guides.

Variablen programmgesteuert übertragen

Sie können ein Programm mit den Funktionen **GetCalc** und **SendCalc** verwenden, um eine Variable von einem Gerät auf ein anderes zu übertragen.

SendCalc sendet eine Variable an den Verbindungsanschluss, von dem ein angeschlossenes Gerät die Variable empfangen kann. Am angeschlossenen Gerät muss dabei entweder der Startbildschirm aufgerufen sein oder die Funktion **GetCalc** aus einem Programm muss ausgeführt werden.

Mit den Befehlen SendCalc bzw. GetCalc können Sie optionale Parameter verwenden, um den USB- oder den E/A-Anschluss anzugeben. (Näheres dazu finden Sie in Anhang A.) Wenn Sie diese Parameter nicht angeben, kommuniziert der TI-89 Titanium über den USB-Anschluss.

Das Chat-Programm

Das folgende Programm verwendet **GetCalc** und **SendCalc**. Das Programm richtet zwei Schleifen ein, mit denen sich die verbundenen Geräte dabei abwechseln können eine mit **msg** benannte Variable zu senden bzw. zu empfange und anzuzeigen. **InputStr** ermöglicht es den Benutzern, eine Nachricht in die Variable **msg** einzugeben.

```
:Chat()
:Prgm
:ClrIO
:Disp "On first unit to send,","
 enter 1;","On first to receive,"
:InputStr " enter 0",msg
:If msg="0" Then
:   While true
:      GetCalc msg
:      Disp msg
:      InputStr msg
:      SendCalc msg
:   EndWhile
:Else
:   While true
:     InputStr msg
:     SendCalc msg
:     GetCalc msg
:     Disp msg
:  EndWhile
:EndIf
:EndPrgm
```

❶ ❷ ❸ ❹ ❺ ❻

Hinweise:

❶ Richtet das betreffende Gerät zum Empfang und zum Anzeigen der Variable msg ein.

❷ Ermöglicht es diesem Benutzer, eine Nachricht in msg einzugeben und diese zu senden.

❸ Schleife, die von dem Gerät durchlaufen wird, das die erste Meldung empfängt.

❹ Ermöglicht es diesem Benutzer, eine Nachricht in msg einzugeben und diese zu

senden.

❺ Richtet dann dieses Gerät zum Empfang und zum Anzeigen von msg ein.

❻ Schleife, die von dem Gerät durchlaufen wird, das die erste Meldung sendete.

Zur Synchronisation von **GetCalc** und **SendCalc** sind die Schleifen so angeordnet, dass das empfangende Gerät **GetCalc** ausführt, während das sendende Gerät auf die Eingabe einer Nachricht durch den Benutzer wartet.

Ausführen des Programms

Diese Prozedur basiert auf folgenden Voraussetzungen:

- Die beiden Geräte sind mit dem Verbindungskabel verbunden.
- Das Chat-Programm ist auf beiden Geräten geladen.
  - Verwenden Sie an jedem Gerät den Programmeditor, um das Programm einzugeben.
    – oder –
  - Geben Sie das Programm an einem Gerät ein, und verwenden Sie dann **VAR-LINK**, um die Programmvariable an das andere Gerät zu übertragen.

So rufen Sie das Programm an beiden Geräten auf:

1. Geben Sie im Startbildschirm beider Geräte **chat( )** ein.
2. Antworten Sie gemäß den nachfolgenden Angaben, nachdem jedes Gerät seine Eingabeaufforderung anzeigt.

Am:	Geben Sie folgendes ein:
Gerät, das die erste Nachricht sendet.	**1** und drücken Sie ENTER.

Am:	Geben Sie folgendes ein:
Gerät, das die erste Nachricht empfängt.	**0** und drücken Sie [ENTER].

3. Geben Sie abwechselnd Nachrichten ein, und drücken Sie [ENTER] , um die Variable **msg** zum anderen Gerät zu senden.

Beenden des Programms

Da das **Chat**-Programm auf beiden Geräten eine Endlosschleife einrichtet, müssen Sie auf beiden Geräten [ON] drücken, um das Programm zu unterbrechen. Wenn Sie [ESC] drücken, um die Fehlermeldung zu bestätigen, stoppt das Programm mit dem E/A-Bildschirm. Drücken Sie [F5] oder [ESC], um zum Startbildschirm zurückzukehren.

Aktualisieren des Betriebssystems (BS)

Sie können das Betriebssystem auf Ihrem TI-89 Titanium oder Voyage™ 200 mit Hilfe Ihres Computers aktualisieren. Außerdem können Sie das BS direkt von einem Gerät zu einem Gerät gleichen Typs übertragen (z.B.von einem TI-89 Titanium auf einen TI-89 Titanium oder von einem Voyage™ 200 auf einen Voyage™ 200).

Durch das Installieren von Betriebssystemsoftware wird der gesamte Speicher eines Geräts auf die ursprünglichen Werkseinstellungen zurückgesetzt. Das bedeutet, dass alle benutzerdefinierten Variablen (die im RAM und im Benutzerdatenarchiv abgelegt waren) sowie Programme, Listen und Ordner (mit Ausnahme des Ordners Main) gelöscht werden. Auch Flash-Anwendungen können eventuell dadurch gelöscht werden. Beachten Sie die wichtigen Informationen über Batterien, bevor Sie eine Aktualisierung des Betriebssystem vornehmen.

Wichtige Informationen zum Download von Betriebssystemen

Bevor Sie einen Betriebssystem-Download beginnen, sollten Sie neue Batterien einsetzen.

Wenn das Gerät im Betriebssystem-Download-Modus arbeitet, ist die Funktion Automatic Power Down™ (APD™) deaktiviert. Falls Sie Ihr Gerät für längere Zeit im Download-Modus belassen, bevor Sie den Download-Vorgang starten, können die Batterien dadurch geschwächt werden. In diesem Fall müssen Sie die schwachen Batterien vor dem Download durch neue ersetzen.

Wenn Sie die Übertragung aus Versehen unterbrechen, bevor sie abgeschlossen ist, müssen Sie das Betriebssystem neu installieren. Achten Sie darauf, dass vor dem Beginn eines Downloads neue Batterien eingelegt werden.

Wenn Probleme auftreten, setzen Sie sich bitte mit Texas Instruments unter TI-Cares™ in Verbindung.

Backup eines Geräts vor der Installation des Betriebssystems

Wenn Sie ein Betriebssystem aktualisieren, geschieht durch diesen Vorgang folgendes:

- Alle benutzerdefinierten Variablen (die im RAM und im Benutzerdatenarchiv abgelegt waren) sowie Programme, Listen und Ordner werden gelöscht.
- Möglicherweise werden alle Flash-Anwendungen gelöscht.
- Alle Systemvariablen und Modi werden auf die ursprünglichen Werkseinstellungen zurückgesetzt. Dies entspricht dem Zurücksetzen des gesamten Speichers über den Bildschirm **MEMORY**.

So sollten vorhandene Variabeln oder Flash-Anwendungen vor der Installation der aktualisierten BS-Version gesichert werden:

- **Wichtig:** Setzen Sie neue Batterien ein.
- Übertragen Sie die Variablen oder Flash-Anwendungen auf ein anderes Gerät.
  – oder –
- Verwenden Sie ein USB cable oder TI-USB-Rechnerkopplungskabel und die Software TI Connect™ (education.ti.com/downloadticonnect), um die Variablen und/oder Flash-Anwendungen an einen Computer zu senden.

Wo man aktualisierte Betriebssystemversionen bekommt

Aktuelle Information über verfügbare aktualisierte Betriebssystemversionen finden Sie auf der Texas Instruments Website unter education.ti.com oder kontaktieren Sie Texas Instruments über TI-Cares™.

Sie können eine aktualisierte Betriebssystemversion oder Flash-Anwendung von der Texas Instruments Website auf einen Computer herunterladen und ein USB cable oder TI-USB-Rechnerkopplungskabel verwenden, um das Betriebssystem bzw. die Anwendung auf Ihrem TI-89 Titanium / Voyage™ 200 zu installieren.

Im Internet finden Sie umfassende Informationen dazu.

Übertragen des Betriebssystems (BS)

Betriebssystemsoftware kann nur von einem TI-89 Titaniumauf einen TI-89 Titanium, von einem TI-89 auf einen TI-89, von einem Voyage™ 200 auf einen Voyage™ 200 oder von einem TI-92 Plus auf einen TI-92 Plus übertragen werden.

So übertragen Sie das Betriebssystem (BS) von Gerät zu Gerät:

1. Verbinden Sie zwei gleiche Geräte, beispielsweise einen TI-89 Titanium mit einem TI-89 Titanium oder einen Voyage™ 200 mit einem Voyage™ 200.
2. Drücken Sie auf dem empfangenden und dem sendenden Gerät [2nd] [VAR-LINK], um den **VAR-LINK** Bildschirm anzuzeigen.
3. Drücken Sie auf dem empfangenden und dem sendenden Gerät [F3] **Link**, um die Menüoptionen aufzurufen.
4. Wählen Sie auf dem empfangenden Gerät **5:Receive OS**.

   Ein Warnhinweis wird eingeblendet. Drücken Sie [ESC], um den Vorgang zu stoppen oder [ENTER], um fortzusetzen. Durch Drücken von [ENTER] wird **VAR-LINK: WAITING TO RECEIVE** und in der Statusleiste des empfangenden Geräts **BUSY** angezeigt.
5. Wählen Sie auf dem sendenden Gerät **4:Send OS**.

   Ein Warnhinweis wird eingeblendet. Drücken Sie [ESC], um den Vorgang zu stoppen oder [ENTER], um die Übertragung zu starten.

Wichtig:

- Achten Sie darauf, dass Sie die Daten auf jedem empfangenden Gerät zuvor sichern und neue Batterien einsetzen.
- Vergewissern Sie sich, dass das sendende und das empfangende Gerät den **VAR-LINK** Bildschirm anzeigen.

Während der Übertragung wird auf dem empfangenden Gerät der Fortschritt der Übertragung angezeigt. Nach erfolgter Übertragung:

- Das sendende Gerät kehrt zum **VAR-LINK** Bildschirm zurück.

- Das empfangende Gerät kehrt entweder zur Apps-Arbeitsfläche oder zum Startbildschirm zurück. Sie müssen möglicherweise ♦ − (schwächer) oder ♦ + (stärker) verwenden, um den Kontrast einzustellen.

Versuchen Sie nicht, die Übertragung eines Betriebssystems abzubrechen.

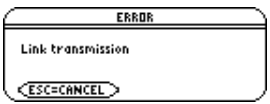
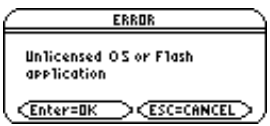
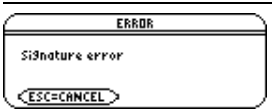
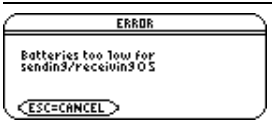
Nach dem Start der Übertragung wird das vorhandene Betriebssystem des empfangenden Geräts gelöscht. Wenn Sie die Übertragung unterbrechen, bevor sie abgeschlossen ist, funktioniert das empfangende Gerät nicht normal. In diesem Fall müssen Sie die aktualisierte Betriebssystemversion neu installieren.

Wenn Sie das Betriebssystem mehrerer Geräte aktualisieren

Um eine Aktualisierung des Betriebssystems mehrerer Geräte durchzuführen, laden Sie das Betriebssystem auf ein Gerät herunter und übertragen es dann von Gerät zu Gerät. Dieses Verfahren ist schneller als die Installation auf jedem Gerät via Computer. Aktualisierte Betriebssystemversionen werden kostenlos freigegeben und vor dem Download oder der Installation ist kein Zertifikat erforderlich.

Fehlermeldungen

Die meisten Fehlermeldungen werden auf dem sendenden Gerät angezeigt. Abhängig davon, wann der Fehler innerhalb des Übertragungsvorgangs auftritt, wird möglicherweise auch auf dem empfangenden Gerät eine Fehlermeldung angezeigt.

Fehlermeldung	Bedeutung
	Das sendende und empfangende Gerät sind nicht korrekt verbunden, oder das empfangende Gerät ist nicht für den Empfang eingestellt.
	Das Zertifikat auf dem empfangenden Gerät ist für das Betriebssystem (BS) des sendenden Geräts nicht gültig. Sie müssen ein gültiges Zertifikat besorgen und installieren. Wenn die App kein Zertifikat mehr erfordert, können Sie sie erneut von der Texas Instruments Website unter education.ti.com herunterladen und die App anschließend erneut auf Ihrem Handheld installieren.
	Während der Übertragung ist ein Fehler aufgetreten. Das aktuelle Betriebssystem auf dem empfangenden Gerät ist beschädigt. Sie müssen die Produktsoftware von einem Computer aus neu installieren.
	Ersetzen Sie die Batterien des Geräts, auf dem diese Meldung angezeigt wird.

Sammeln und Übertragen von ID-Listen

Die Option **6:Send ID List** im Menü F3 **VAR-LINK** ermöglicht es, elektronische ID-Nummern von einzelnen TI-89 Titanium, TI-89, Voyage™ 200 Graphischer Rechner oder TI-92 Plus Geräten zu sammeln.

ID-Listen und Gruppenzertifikate

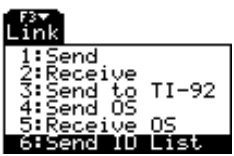
Die ID-Listenfunktion bietet eine einfache Möglichkeit, Geräte-IDs zum Zweck des Erwerbs kommerzieller Anwendungen für Gruppen zu sammeln. Nachdem die IDs gesammelt wurden, müssen sie an Texas Instruments übertragen werden, damit ein Gruppenzertifikat ausgestellt werden kann.

Ein Gruppenzertifikat ermöglicht die Verteilung erworbener Software auf mehrere Geräte der Typen TI-89 Titanium, TI-89, Voyage™ 200 oder TI-92 Plus. Die Software kann so oft es gewünscht wird geladen, vom Gerät gelöscht und erneut geladen werden, solange sie im Gruppenzertifikat aufgeführt bleibt. Sie können in ein Gruppenzertifikat neue ID-Nummern sowie neue kommerzielle Anwendungen aufnehmen.

Sammeln von ID-Listen

Sie können ein Gerät benutzen, um alle gewünschten IDs zu sammeln oder mehrere Geräte verwenden, auf diesen jeweils IDs sammeln und diese Sammlungen anschließend auf einem Gerät zusammenführen.

Um eine ID-Nummer von einem Gerät an ein anderes zu senden, verbinden Sie die beiden Geräte als Erstes mit einem USB-Geräteverbindungskabel oder Standard-Geräteverbindungskabel.

Schritt:	Am:	Folgendes tun:
1.	Sammelgerät (Empfangendes Gerät)	Rufen Sie den Startbildschirm auf. Drücken Sie: [HOME] [◆] [CALC HOME]
2.	Sendendes Gerät	a. Drücken Sie [2nd] [VAR-LINK], um den **VAR-LINK** Bildschirm anzuzeigen.
		b. Drücken Sie [F3] **Link**, und wählen Sie **6:Send ID List**. 
		Das sendende Gerät fügt eine Kopie seiner eindeutigen ID-Nummer in die ID-Liste des Sammelgeräts ein. Das sendende Gerät behält dabei seine ID-Nummer bei, die von dem Gerät nicht gelöscht werden kann.
3.	Weitere Geräte	Wiederholen Sie die Schritte 1 und 2, bis auf dem Gerät alle IDs gesammelt wurden. Je nachdem, wieviel Speicher auf dem Sammelgerät verfügbar ist, ist es möglich, mehr als 4.000 IDs zu sammeln.

Hinweise:

- Sie können weder die ID-Liste auf dem sendenden Gerät noch die auf dem Sammelgerät anzeigen.
- Jedes Mal, nachdem eine ID-Liste erfolgreich von einem Gerät zu einem anderen gesendet wurde, wird sie automatisch vom sendenden Gerät gelöscht.
- Wenn eine ID eines Geräts zweimal gesammelt wird, wird das Duplikat automatisch aus der Liste gelöscht.

Löschen des Inhalts der ID-Liste

Die ID-Liste bleibt auch nach einem erfolgreichen Upload auf den Computer auf dem Sammelgerät bestehen. Sie können die Liste vom Sammelgerät anschließend auch auf andere Computer hochladen.

So löschen Sie die ID-Liste vom Sammelgerät:

1. Drücken Sie 2nd [VAR-LINK], um den **VAR-LINK** Bildschirm anzuzeigen.
2. Drücken Sie F1 **Manage**, und wählen Sie **A:Clear ID List**.

Kompatibilität zwischen TI-89 Titanium, Voyage™ 200, TI-89 und TI-92 Plus

Im Allgemeinen und mit wenigen Ausnahmen sind Daten und Programme eines TI-89 Titanium, TI-89, Voyage™ 200 und TI-92 Plus kompatibel.

Die meisten Funktionen des TI-89 Titanium sind kompatibel mit dem TI-89, Voyage™ 200 und TI-92 Plus. Der TI-89 Titanium und der TI-89 ähneln einander, außer dass der TI-89 Titanium mehr Speicher besitzt (d.h. mehr Platz für Apps und Benutzerarchiv) und der TI-89 Titanium einen USB-Anschluss besitzt. Der Voyage™ 200 gleicht dem TI-92 Plus, außer dass er mehr Speicher, d.h. mehr Platz für Anwendungen (Apps) besitzt.

Zwischen TI-89 Titanium, TI-89, Voyage™ 200 und TI-92 Plus sind alle Daten kompatibel; bestimmte Programme, die für eines dieser Geräte geschrieben wurden laufen jedoch möglicherweise auf einem der anderen Geräte nicht oder nicht in gleicher Weise, was auf die unterschiedlichen Bildschirmgrößen und Tastaturen oder den USB-Anschluss des TI-89 Titanium zurückzuführen ist.

Ander Inkompatibilitäten können durch unterschiedliche Versionen des Betriebssystems entstehen. Die neueste Version des Betriebssystems kann von der Texas Instruments Website unter education.ti.com/downloadticonnect heruntegeladen werden.

Verbindungs- und Übertragungstabelle

An → / Von ↓	TI-89 Titanium	TI-89	Voyage™ 200	TI-92 Plus
TI-89 Titanium	BS Apps Variablen	Apps Variablen	Variablen	Variablen
TI-89	Apps Variablen	BS Apps Variablen	Variablen	Variablen
Voyage™ 200	Variablen	Variablen	BS Apps Variablen	Apps Variablen
TI-92 Plus	Variablen	Variablen	Apps Variablen	BS Apps Variablen

Anhang A: Funktionen und Anweisungen

Gruppenindex 862

Alphabetische Zusammenstellung der Anweisungen und Funktionen 866

In diesem Anhang sind sämtliche Funktionen und Anweisungen des TI-89 Titanium / Voyage™ 200 mit einer Erläuterung ihrer Syntax und ihrer Wirkungsweise zusammengestellt.

Name der Funktion bzw. der Anweisung.

Taste(n) oder Menüs. Sie können den Namen auch eingeben.

Beispiel

Circle **CATALOG**

Circle *x*, *y*, *r* [, *Zeichenmodus*]

Zeichnet einen Kreis, mit dem Mittelpunkt an den Fensterkoordinaten *x, y* und einem Radius *r*.

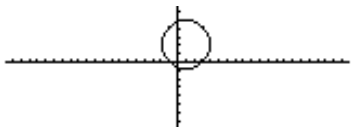
x, *y* und *r* müssen reelle Werte sein..

Bei *Zeichenmodus*= 1 wird der Kreis gezeichnet (Vorgabe).
Bei *Zeichenmodus*= 0 wird der Kreis ausgeschaltet.
Bei *Zeichenmodus*= -1 werden die Pixel invertiert.

Hinweis: Beim Neuzeichnen werden alle gezeichneten Elemente gelöscht.

In einem ZoomSqr-Ansichtfenster:

```
ZoomSqr:Circle 1,2,3 ENTER
```



Erläuterung der Funktion/der Anweisung.

Parameter in *Kursivschrift*. Parameter in [ ] sind optional. Die [ ] nicht mit eingeben.

Syntaxzeile mit Reihenfolge und Typ der Parameter, die Sie angeben. Bei mehreren Parametern ist jeweils ein Komma (,) als Trennzeichen einzugeben.

Gruppenindex

In diesem Abschnitt sind die Funktionen und Anweisungen des TI-89 Titanium / Voyage™ 200 zum schnellen Auffinden in Gruppen zusammengefaßt. Hinter der Funktion bzw. Anweisung finden Sie die Seite, auf der sie in diesem Anhang erläutert ist.

Algebra

| ("**with**")	994	**cFactor()**	871	**comDenom()**	874
cSolve()	879	**cZeros**	882	**expand()**	898
factor()	899	**getDenom()**	906	**getNum()**	907
nSolve()	929	**propFrac()**	937	**randPoly()**	943
solve()	960	**tCollect()**	970	**tExpand()**	970
zeros()	976				

Analysis

∫**()** (Integral)	989	Π**()** (Produkt)	990	Σ**()** (Summe)	990
arcLen()	868	**avgRC()**	869	***d()***	885
deSolve()	888	**fMax()**	901	**fMin()**	902
limit()	915	**nDeriv()**	925	**nInt()**	927
' (Strich)	992	**seq()**	950	**taylor()**	969

Graphische Darstellung

AndPic	867	**BldData**	870	**Circle**	872
ClrDraw	872	**ClrGraph**	873	**CyclePic**	882
DrawFunc	892	**DrawInv**	892	**DrawParm**	893
DrawPol	893	**DrawSlp**	893	**DrwCtour**	894
FnOff	902	**FnOn**	902	**Graph**	910
Line	916	**LineHorz**	916	**LineTan**	916
LineVert	917	**NewPic**	926	**PtChg**	937
PtOff	937	**PtOn**	938	**ptTest()**	938
PtText	938	**PxlChg**	938	**PxlCrcl**	938
PxlHorz	939	**PxlLine**	939	**PxlOff**	939
PxlOn	939	**pxlTest()**	940	**PxlText**	940
PxlVert	940	**RclGDB**	943	**RclPic**	944
RplcPic	948	**Shade**	955	**StoGDB**	964
StoPic	965	**Style**	965	**Trace**	973
XorPic	976	**ZoomBox**	978	**ZoomData**	979
ZoomDec	979	**ZoomFit**	980	**ZoomIn**	980
ZoomInt	980	**ZoomOut**	981	**ZoomPrev**	981
ZoomRcl	981	**ZoomSqr**	981	**ZoomStd**	982
ZoomSto	982	**ZoomTrig**	982		

Listen

+ (Addition)	982	− (Subtrak.)	983	∗ (Multiplik.)	984
/ (Division)	984	⁻ (Negation)	986	**^** (Potenz)	985
augment()	868	**crossP()**	878	**cumSum()**	881
dim()	891	**dotP()**	892	**exp▸list()**	897
left()	914	**list▸mat()**	918	**Δlist()**	917
mat▸list()	921	**max()**	922	**mid()**	923
min()	924	**newList()**	926	**polyEval()**	935
product()	936	**right()**	946	**rotate()**	946
shift()	956	**SortA**	963	**SortD**	963
sum()	966				

Mathematik allgemein

+ (Addition)	982	− (Subtrak.)	983	∗ (Multiplik.)	984
⁄ (Division)	984	⁻ (Negation)	986	% (Prozent)	986
! (Fakultät)	988	√**()** (Qdr.-wurzel)	989	**^** (Potenz)	985
° (Ordnung)	991	∠ (Winkel)	991	°, ', "	992
_ (Unterstr.)	992	▸ (Konvertieren)	993	**10^()**	993
0b, 0h	995	**▸Bin**	869	**▸Cylind**	882
▸DD	885	**▸Dec**	886	**▸DMS**	892
▸Hex	910	**▸Polar**	935	**▸Rect**	944
▸Sphere	963	**abs()**	866	**and**	866
angle()	867	**approx()**	868	**ceiling()**	870
conj()	875	**cos**	875	**cos^{-1}()**	876
cosh()	876	**cosh^{-1}()**	877	**cot()**	877
cot^{-1}()	877	**coth()**	877	**coth^{-1}()**	878
csc()	878	**csc^{-1}()**	878	**csch()**	878
csch^{-1}()	879	E	894	***e^()***	895
exact()	897	**floor()**	901	**fPart()**	903
gcd()	904	**imag()**	911	**int()**	913
intDiv()	913	**iPart()**	913	**isPrime()**	914
lcm()	914	**ln()**	918	**log()**	919
max()	922	**min()**	924	**mod()**	924
nCr()	925	**nPr()**	929	**P▸Rx()**	932
P▸Ry()	932	ʳ (rad)	991	**R▸Pθ()**	942
R▸Pr()	942	**real()**	944	**remain()**	945
rotate()	946	**round()**	947	**sec()**	949
sec^{-1}()	949	**sech()**	949	**sech^{-1}()**	949
shift()	956	**sign()**	957	**sin()**	958
sin^{-1}()	959	**sinh()**	959	**sinh^{-1}()**	959
tan()	968	**tan^{-1}()**	968	**tanh()**	968
tanh^{-1}()	969	**tmpCnv()**	971	**ΔtmpCnv()**	972
x^{-1}	994				

Matrizen

+ (Addition)	982	− (Subtrak.)	983	∗ (Multiplik)	984
⁄ (Division)	984	⁻ (Negation)	986	**.+** (Pkt.-Add.)	985
.− (Pkt.-Sub.)	985	.∗ (Pkt.-Mult.)	986	**./** (Pkt.-Div.)	986
.^ (Pkt.-Potenz)	986	**^** (Potenz)	985	**augment()**	868
colDim()	874	**colNorm()**	874	**crossP()**	878
cumSum()	881	**det()**	890	**diag()**	890
dim()	891	**dotP()**	892	**eigVc()**	895
eigVl()	895	**Fill**	901	**identity()**	911
list▸mat()	918	**LU**	921	**mat▸list()**	921
max()	922	**mean()**	922	**median()**	922
min()	924	**mRow()**	924	**mRowAdd()**	924
newMat()	926	**norm()**	928	**product()**	936
QR	941	**randMat()**	943	**ref()**	945
rowAdd()	947	**rowDim()**	947	**rowNorm()**	948
rowSwap()	948	**rref()**	948	**simult()**	958
stdDev()	964	**subMat()**	965	**sum()**	966
T	967	**unitV()**	974	**variance()**	974
x^{-1}	994				

Programmierung

=	987	**≠**	987	**<**	987
≤	988	**>**	988	**≥**	988
# (Umleitung)	990	**→** (Speichern)	995	**◕** (Kommentar)	995
and	866	**ans()**	868	**Archive**	868
checkTmr()	872	**ClockOff**	872	**ClockOn**	872
ClrErr	873	**ClrGraph**	873	**ClrHome**	873
ClrIO	873	**ClrTable**	873	**CopyVar**	875
CustmOff	881	**CustmOn**	881	**Custom**	882
Cycle	882	**dayOfWk()**	885	**Define**	886
DelFold	887	**DelVar**	887	**Dialog**	890
Disp	891	**DispG**	891	**DispHome**	891
DispTbl	892	**DropDown**	894	**Else**	895
ElseIf	896	**EndCustm**	896	**EndDlog**	896
EndFor	896	**EndFunc**	896	**EndIf**	896
EndLoop	896	**EndPrgm**	896	**EndTBar**	896
EndTry	896	**EndWhile**	896	**entry()**	896
Exec	897	**Exit**	897	**For**	903
format()	903	**Func**	904	**Get**	904
GetCalc	904	**getConfg()**	905	**getDate()**	905
getDtFmt()	906	**getDtStr()**	906	**getFold()**	906
getKey()	906	**getMode()**	907	**getTime()**	907
getTmFmt()	907	**getTmStr()**	907	**getTmZn()**	908
getType()	908	**getUnits()**	909	**Goto**	909
If	911	**Input**	912	**InputStr**	912
isClkOn()	913	**Item**	914	**Lbl**	914
left()	914	**Local**	919	**Lock**	919
Loop	920	**MoveVar**	924	**NewFold**	926
NewProb	927	**not**	928	**or**	930
Output	931	**part()**	932	**PassErr**	934
Pause	934	**PopUp**	936	**Prgm**	936
Prompt	937	**Rename**	945	**Request**	945
Return	946	**right()**	946	**Send**	949
SendCalc	950	**SendChat**	950	**setDate()**	951
setDtFmt()	951	**setFold()**	951	**setGraph()**	951
setMode()	952	**setTable()**	953	**setTime()**	953
setTmFmt()	953	**setTmZn()**	954	**setUnits()**	954
startTmr()	963	**Stop**	964	**Style**	965
switch()	966	**Table**	967	**Text**	970
Then	970	**timeCnv()**	971	**Title**	971
Toolbar	972	**Try**	973	**Unarchiv**	974
Unlock	974	**when()**	975	**While**	975
xor	976				

Statistik

! (Fakultät)	988	**BldData**	870	**CubicReg**	881
cumSum()	881	**ExpReg**	899	**LinReg**	917
LnReg	918	**Logistic**	920	**mean()**	922
median()	922	**MedMed**	923	**nCr()**	925
NewData	925	**NewPlot**	927	**nPr()**	929
OneVar	930	**PlotsOff**	935	**PlotsOn**	935
PowerReg	936	**QuadReg**	941	**QuartReg**	942
rand()	943	**randNorm()**	943	**RandSeed**	943
ShowStat	957	**SinReg**	960	**SortA**	963
SortD	963	**stdDev()**	964	**TwoVar**	973
variance()	974				

Zeichenketten

& (Verketten)	988	**#** (Umleitung)	990	**char()**	871
dim()	891	**expr()**	899	**format()**	903
inString()	912	**left()**	914	**mid()**	923
ord()	931	**right()**	946	**rotate()**	946
shift()	956	**string()**	965		

Alphabetische Zusammenstellung der Anweisungen und Funktionen

Operatoren, deren Namen nicht alphabetisch sind (wie +, ! und >) finden Sie am Ende dieses Anhangs ab Seite 982. Wenn nicht anders angegeben, wurden sämtliche Beispiele im standardmäßigen Reset-Modus ausgeführt, wobei alle Variablen als undefiniert angenommen wurden. Aus Platzgründen werden Näherungsergebnisse nur mit drei Nachkommastellen angezeigt (3.14159265359 ist hier als 3.141... abgebildet).

abs()

MATH/Number-Menü

abs(*Term1***)** ⇒ *Term*
abs(*Liste1***)** ⇒ *Liste*
abs(*Matrix1***)** ⇒ *Matrix*

Gibt den absoluten Wert des Parameters zurück.

Ist der Parameter eine komplexe Zahl, wird der Betrag der Zahl zurückgegeben.

Hinweis: Alle undefinierten Variablen werden als reelle Variablen behandelt.

```
abs({π/2,-π/3}) ENTER        {π/2   π/3}
abs(2-3i) ENTER                √13
abs(z) ENTER                   |z|
abs(x+yi) ENTER                √(x²+y²)
```

and

MATH/Test-Menü

Boolescher Term1 **and** *Term2* ⇒ *Boolescher Term*
Boolesche Liste1 **and** *Liste2* ⇒ *Boolesche Liste*
Boolesche Matrix1 **and** *Matrix2* ⇒ *Boolesche Matrix*

Gibt "wahr" oder "falsch" oder eine vereinfachte Form des ursprünglichen Terms zurück.

```
x≥3 and x≥4 ENTER                          x≥4
{x≥3,x≤0} and {x≥4,x≤-2} ENTER
                                  {x≥4   x≤-2}
```

Ganze_Zahl1 **and** *Ganze_Zahl2* ⇒ *Ganze_Zahl*

Vergleicht zwei reelle ganze Zahlen mit Hilfe einer **and**-Operation Bit für Bit. Intern werden beide ganzen Zahlen in binäre 32-Bit-Zahlen mit Vorzeichen konvertiert. Beim Vergleich der sich entsprechenden Bits ist das Ergebnis dann 1, wenn beide Bits 1 sind; anderenfalls ist das Ergebnis 0. Der zurückgegebene Wert stellt die Bit-Ergebnisse dar und wird im jeweiligen Base-Modus angezeigt.

Sie können die ganzen Zahlen in jeder Basis eingeben. Für einen binären oder hexadezimalen Eintrag ist das Präfix 0b bzw. 0h zu verwenden. Ohne Präfix werden ganze Zahlen als dezimal behandelt (Basis 10).

Geben Sie eine dezimale ganze Zahl ein, die für eine 32-Bit-Dualform mit Vorzeichen zu groß ist, dann wird eine symmetrische Modulo-Operation ausgeführt, um den Wert in den erforderlichen Bereich zu bringen.

Im Hex Modus:

```
0h7AC36 and 0h3D5F ENTER   0h2C16
```

Wichtig: Null, nicht Buchstabe O.

Im Bin Modus:

```
0b100101 and 0b100 ENTER    0b100
```

Im Dec Modus:

```
37 and 0b100 ENTER                4
```

Hinweis: Ein binärer Eintrag kann bis zu 32 Stellen haben (das Präfix 0b wird nicht mitgezählt). Ein hexadezimaler Eintrag kann bis zu 8 Stellen aufweisen.

AndPic CATALOG

AndPic *picVar*[, *Zeile, Spalte*]

Zeigt den Graphikbildschirm an und überlagert den aktuellen Graphikbildschirm ab den Pixel-Koordinaten *Zeile, Spalte* in einer logischen AND-Verknüpfung mit dem in *picVar* gespeicherten Bild.

picVar muß vom Datentyp "Bild" (PIC) sein.

Vorgabewert der Koordinaten ist (0,0), also die linke obere Bildschirmecke.

Im Graphik-Modus Funktion und im Y= Editor:

`y1(x) = cos(x)` ⊙

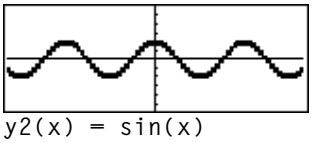
[2nd] [F6] `Style = 3:Square`

[F6] `Style = 3:Square`

[F2] `Zoom = 7:ZoomTrig`

[F1] `= 2:Save Copy As...`

`Type = Picture, Variable = PIC1`



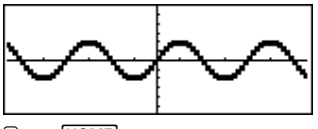
`y2(x) = sin(x)`

[2nd] [F6] `Style = 3:Square`

[F6] `Style = 3:Square`

`y1 = no checkark (F4 to deselect)`

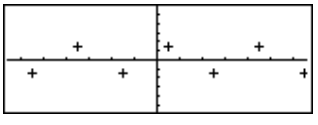
[F2] `Zoom = 7:ZoomTrig`



[HOME]

[◆] [CALC HOME]

`AndPic PIC1` [ENTER] Done



angle() MATH/Complex-Menü

angle(*Term1***)** ⇒ *Term*

Gibt den Winkel von *Term1* zurück, wobei *Term1* als komplexe Zahl interpretiert wird.

Hinweis: Alle undefinierten Variablen werden als reelle Variablen behandelt.

Im Degree-Modus für Winkel:

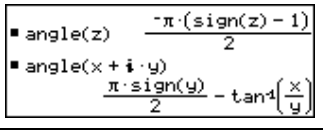
`angle(0+2i)` [ENTER] 90

Im Radian-Modus:

`angle(1+i)` [ENTER] $\frac{\pi}{4}$

`angle(z)` [ENTER]

`angle(x+iy)` [ENTER]



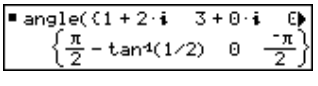
---

angle(*Liste1***)** ⇒ *Liste*
angle(*Matrix1***)** ⇒ *Matrix*

Gibt als Liste oder Matrix die Winkel der Elemente aus *Liste1* oder *Matrix1* zurück, wobei jedes Element als komplexe Zahl interpretiert wird.

Im Radian-Modus:

`angle({1+2i,3+0i,0-4i})` [ENTER]



ans() 2nd [ANS] -Taste

an() ⇒ *Wert*
ans(*Ganze_Zahl***)** ⇒ *Wert*

Gibt eine Antwort aus dem Protokoll-Bereich des Hauptbildschirms zurück.

Ganze_Zahl gibt die laufende Nummer der gewünschten Antwort an. Gültige Werte für *Ganze_Zahl* sind 1 bis 99, wobei kein Term zulässig ist. Vorgabe ist 1, also die letzte Antwort.

Um mit **ans()** die Fibonacci-Folge auf dem Hauptbildschirm zu erzeugen, geben Sie ein:

1 ENTER	1
1 ENTER	1
2nd [ANS] + 2nd [ANS] ◀ ← 2 ENTER	2
ENTER	3
ENTER	5

approx() MATH/Algebra-Menü

approx(*Term***)** ⇒ *Wert*

Gibt den Wert von *Term* ungeachtet der aktuellen Einstellung des Exact/Approx-Modus als Dezimalwert zurück, sofern möglich.

Gleichwertig damit ist die Eingabe von *Term* und Drücken von ◆ ENTER im Hauptbildschirm.

```
approx(π) ENTER                3.141...
```

---

approx(*Liste1***)** ⇒ *Liste*
approx(*Matrix1***)** ⇒ *Matrix*

Gibt, sofern möglich, eine Liste oder Matrix zurück, in der jedes Element dezimal ausgewertet wurde.

```
approx({sin(π),cos(π)}) ENTER
                          {0.  -1.}
approx([√(2),√(3)]) ENTER
                  [1.414...  1.732...]
```

Archive CATALOG

Archive *Var1* [, *Var2*] [, *Var3*] ...

Überträgt die angegebenen Variablen vom RAM-Speicher in den Benutzer-Archivspeicher.

Auf eine archivierte Variable kann ebenso wie auf eine Variable im RAM-Speicher zugegriffen werden. Sie können eine archivierte Variable allerdings weder löschen, umbenennen noch ihr einen neuen Wert zuweisen, da sie automatisch gesperrt ist.

Zum Entfernen der Variablen aus dem Archiv dient **Unarchiv**.

```
10→arctest ENTER                10
Archive arctest ENTER           Done
5*arctest ENTER                 50
15→arctest ENTER
```

```
ESC
Unarchiv arctest ENTER          Done
15→arctest ENTER                15
```

arcLen() MATH/Calculus-Menü

arcLen(*Term1***,***Var***,***Start***,***Ende***)** ⇒ *Term*

Gibt die Bogenlänge von *Term1* von *Start* bis *Ende* bezüglich der Variablen *Var* zurück.

Ungeachtet des Graphikmodus wird die Bogenlänge als Integral unter Annahme einer Definition in Modus Funktion berechnet.

```
arcLen(cos(x),x,0,π) ENTER  3.820...
arcLen(f(x),x,a,b) ENTER
```

$$\int_a^b \sqrt{\left(\frac{d}{dx}(f(x))\right)^2+1}\ dx$$

---

arcLen(*Liste1***,***Var***,***Start***,***Ende***)** ⇒ *Liste*

Gibt eine Liste der Bogenlängen für jedes Element von *Liste1* zwischen *Start* und *Ende* bezüglich der Variablen *Var* zurück.

```
arcLen({sin(x),cos(x)},x,0,π)
                  {3.820...  3.820...}
```

augment() MATH/Matrix-Menü

augment(*Liste1, Liste2***)** ⇒ *Liste*

Gibt eine neue Liste zurück, die durch Anfügen von *Liste2* ans Ende von *Liste1* erzeugt wurde.

```
augment({1,-3,2},{5,4}) ENTER
                      {1  -3  2  5  4}
```

augment(*Matrix1***,** *Matrix2***)** ⇒ *Matrix*
augment(*Matrix1***;** *Matrix2***)** ⇒ *Matrix*

Gibt eine neue Matrix zurück, die durch Anfügen von *Matrix2* an *Matrix1* erzeugt wurde. Wenn das Zeichen "," verwendet wird, müssen die Matrizen gleiche Zeilendimensionen besitzen, und *Matrix2* wird Spaltenweise an *Matrix1* angefügt. Wenn das ";" verwendet wird, müssen die Matrizen gleiche Spaltendimension haben und *Matrix2* wird zeilenweise an *Matrix1* angefügt. Verändert weder *Matrix1* noch *Matrix2*.

`[1,2;3,4]→M1` ENTER

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

`[5;6]→M2` ENTER

$$\begin{bmatrix} 5 \\ 6 \end{bmatrix}$$

`augment(M1,M2)` ENTER

$$\begin{bmatrix} 1 & 2 & 5 \\ 3 & 4 & 6 \end{bmatrix}$$

`[5,6]→M2` ENTER

$$\begin{bmatrix} 5 & 6 \end{bmatrix}$$

`augment(M1;M2)` ENTER

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$$

avgRC() CATALOG

avgRC(*Term1***,** *Var* **[,** *h***])** ⇒ *Term*

Gibt den rechtsseitigen Differenzquotienten zurück (durchschnittliche Änderungsrate).

Term1 kann eine benutzerdefinierte Funktion sein (sieh **Func**).

h ist der Schrittwert. Wird *h* nicht angegeben, wird als Vorgabewert 0,001 benutzt.

Beachten Sie, daß die ähnliche Funktion **nDeriv()** den zentralen Differenzenquotienten benutzt.

`avgRC(f(x),x,h)` ENTER

$$\frac{f(x+h) - f(x)}{h}$$

`avgRC(sin(x),x,h)|x=2` ENTER

$$\frac{\sin(h+2) - \sin(2)}{h}$$

`avgRC(x^2-x+2,x)` ENTER

$$2.\cdot(x-.4995)$$

`avgRC(x^2-x+2,x,.1)` ENTER

$$2.\cdot(x-.45)$$

`avgRC(x^2-x+2,x,3)` ENTER $2\cdot(x+1)$

▸Bin MATH/Base-Menü

Ganze_Zahl1 **▸Bin** ⇒ *Ganze_Zahl*

Konvertiert *Ganze_Zahl1* in eine Dualzahl. Dual- oder Hexadezimalzahlen weisen stets das Präfix 0b bzw. 0h auf.

Null (nicht Buchstabe O) und b oder h.
0b *binäre_Zahl*
0h *hexadezimale_Zahl*
Eine Dualzahl kann bis zu 32 Stellen aufweisen, eine Hexadezimalzahl bis zu 8.

Ohne Präfix wird *Ganze_Zahl1* als dezimal (Basis 10) behandelt. Das Ergebnis wird unabhängig vom Base-Modus binär angezeigt.

Geben Sie eine dezimale ganze Zahl ein, die für eine 32-Bit-Dualform mit Vorzeichen zu groß ist, dann wird eine symmetrische Modulo-Operation ausgeführt, um den Wert in den erforderlichen Bereich zu bringen.

`256▸Bin` ENTER `0b100000000`

`0h1F▸Bin` ENTER `0b11111`

BldData CATALOG

BldData [*DatenVar*]

Erzeugt auf Grundlage der für den Plot des aktuellen Graphs verwendeten Informationen die Datenvariable *DatenVar*. **BldData** ist in allen Graphikmodi gültig.

Wird *DatenVar* weggelassen, so werden die Daten in der Systemvariablen sysData gespeichert.

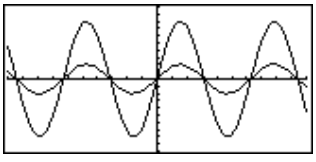
Hinweis: Wenn Sie den Daten/Matrix-Editor nach der Verwendung von **BldData** zum ersten Mal starten, wird *DatenVar* oder sysData (je nach dem mit **BldData** verwendeten Argument) als aktuelle Datenvariable gesetzt.

Die für unabhängige Variablen verwendeten anwachsenden Werte (im Beispiel rechts: x) werden gemäß den Werten der Fenstervariablen berechnet.

Näheres zu den Inkrementen, die zur Auswertung eines Graphen verwendet werden, finden Sie im modul über den entsprechenden Graphikmodus.

Im Funktions-Graphikmodus und im Radian-Modus:

`8*sin(x)→y1(x)` ENTER Done
`2*sin(x)→y2(x)` ENTER Done
`ZoomStd` ENTER



HOME
◆ [CALC HOME]

`BldData` ENTER Done
APPS 6 ENTER

DATA	x	y1	y2
	c1	c2	c3
1	-10.	4.3522	1.088
2	-9.832	3.168	.792
3	-9.664	1.8945	.47363
4	-9.496	.56769	.14192

Der 3D-Graphikmodus weist zwei unabhängige Variablen auf. Beachten Sie, daß im nebenstehenden Beispiel x konstant bleibt, während y in seinem Wertebereich erhöht wird.

Dann wächst x auf seinen nächsten Wert an, und y wird wiederum in seinem Bereich erhöht. Dieses Prinzip läuft so lange ab, bis x seinen Wertebereich durchlaufen hat.

Hinweis: Folgende Beispieldaten stammen aus einem 3D-Graph.

DATA	x	y	z1
	c1	c2	c3
1	-10.	-10.	0.
2	-10.	-8.571	5.8309
3	-10.	-7.143	8.9706
4	-10.	-5.714	9.8677

ceiling() MATH/Number-Menü

ceiling(*Term1***)** ⇒ *Ganze_Zahl*

Gibt die erste ganze Zahl zurück, die ≥ Parameter ist.

Der Parameter kann eine reelle oder eine komplexe Zahl sein.

Hinweis: Siehe auch **floor()**.

`ceiling(0.456)` ENTER 1.

---

ceiling(*Liste1***)** ⇒ *Liste*
ceiling(*Matrix1***)** ⇒ *Matrix*

Gibt eine Liste bzw. eine Matrix zurück, die für alle Elemente die erste ganze Zahl ≥ Element enthält.

`ceiling({-3.1,1,2.5})` ENTER
{-3. 1 3.}

`ceiling([0,-3.2 i;1.3,4])` ENTER

$$\begin{bmatrix} 0 & -3.\cdot i \\ 2. & 4 \end{bmatrix}$$

cFactor() MATH/Algebra/Complex-Menü

cFactor(*Term1***[, *Var*])** ⇒ *Term*
cFactor(*Liste1***[, *Var*])** ⇒ *Liste*
cFactor(*Matrix1***[, *Var*])** ⇒ *Matrix*

cFactor(*Term1***)** gibt *Term1* nach allen seinen Variablen über einem gemeinsamen Nenner faktorisiert zurück.

Term1 wird soweit wie möglich in lineare rationale Faktoren zerlegt, selbst, wenn dies die Einführung neuer nicht-reeller Zahlen bedeutet. Diese Alternative ist angemessen, wenn Sie die Faktorisierung bezüglich mehr als einer Variablen vornehmen möchten.

```
cFactor(a^3*x^2+a*x^2+a^3+a)
ENTER
    a·(a + -i)·(a + i)·(x + -i)·(x + i)

cFactor(x^2+4/9) ENTER
          (3·x + -2·i)·(3·x + 2·i)
                    9

cFactor(x^2+3) ENTER          x^2 + 3

cFactor(x^2+a) ENTER          x^2 + a
```

---

cFactor(*Term1,Var***)** gibt *Term1* nach der Variablen *Var* faktorisiert zurück.

Term1 wird soweit wie möglich in Faktoren zerlegt, die linear in *Var* sind, mit möglicherweise nicht-reellen Konstanten, selbst wenn irrationale Konstanten oder Unterterme, die in anderen Variablen irrational sind, eingeführt werden.

Die Faktoren und ihre Terme werden mit *Var* als Hauptvariable sortiert. Gleichartige Potenzen von *Var* werden in jedem Faktor zusammengefaßt. Beziehen Sie *Var* ein, wenn die Faktorisierung nur bezüglich dieser Variablen benötigt wird und Sie irrationale Terme in anderen Variablen akzeptieren möchten, um die Faktorisierung bezüglich *Var* so weit wie möglich vorzunehmen. Es kann sein, dass als Nebeneffekt in gewissem Umfang eine Faktorisierung nach anderen Variablen auftritt.

```
cFactor(a^3*x^2+a*x^2+a^3+a,x)
ENTER
          a·(a^2 + 1)·(x + -i)·(x + i)

cFactor(x^2+3,x) ENTER
              (x + √3·i)·(x + -√3·i)

cFactor(x^2+a,x) ENTER
               (x + √a·-i)·(x + √a·i)
```

Bei der Einstellung AUTO für den Modus Exact/Approx ermöglicht die Einbeziehung von *Var* auch eine Näherung mit Gleitkommakoeffizienten in Fällen, wo irrationale Koeffizienten nicht explizit bezüglich der in den TI-89 integrierten Funktionen ausgedrückt werden können. Selbst wenn es nur eine Variable gibt, kann das Einbeziehen von *Var* eine vollständigere Faktorisierung ermöglichen.

```
cFactor(x^5+4x^4+5x^3-6x-3)
ENTER
            x^5 + 4·x^4 + 5·x^3 - 6·x - 3

cFactor(ans(1),x) ENTER
  (x - .965)·(x + .612)·(x + 2.13)·
              (x + 1.11 - 1.07·i)·
               (x + 1.11 + 1.07·i)
```

Hinweis: Siehe auch **factor()**.

char() MATH/String-Menü

char(*Ganze_Zahl***)** ⇒ *Zeichen*

Gibt ein Zeichen (einen "Character") zurück, die das Zeichen mit der Nummer *Ganze_Zahl* aus dem Zeichensatz des TI-89 Titanium / Voyage™ 200 enthält. Eine vollständige Aufstellung der Zeichen des TI-89 Titanium / Voyage 200 und der Zeichencodes finden Sie in Anhang B.

Der gültige Wertebereich für *Ganze_Zahl* ist 0–255.

```
char(38) ENTER                "&"

char(65) ENTER                "A"
```

checkTmr() CATALOG

checkTmr(*starttime***)** ⇒ *integer*

Gibt eine ganze Zahl (Integer) zurück, welche die Anzahl der seit dem Start des Timers vergangen Sekunden darstellt. *starttime* ist eine Integer-Zahl die durch die Funktion **startTmr()** zurückgegeben wird.

Sie können auch eine Liste oder Matrize von *starttime*-Integern verwenden. Gültige *starttime* Integer-Werte liegen zwischen 0 und der aktuellen Zeit der Uhr.

Sie können mehrere Timer gleichzeitig laufen lassen.

Hinweis: Siehe auch **startTmr()** und **timeCnv()**.

```
startTmr() ENTER                148083315

checkTmr(148083315)                  34


startTmr()→Timer1
⋮
startTmr()→Timer2
⋮
checkTmr(Timer1)→Timer1Wert
⋮
checkTmr(Timer2)→Timer2 Wert
```

Circle CATALOG

Circle *x*, *y*, *r* [, *Zeichenmodus*]

Zeichnet einen Kreis, mit dem Mittelpunkt an den Fensterkoordinaten *x, y* und Radius *r*.

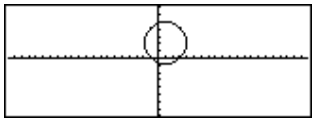
x, y und *r* müssen reelle Werte sein.

Bei *Zeichenmodus*= 1 wird der Kreis gezeichnet (Vorgabe).
Bei *Zeichenmodus*= 0 wird der Kreis gelöscht.
Bei *Zeichenmodus*= -1 werden die Pixel invertiert.

Hinweis: Beim Neuzeichnen werden alle gezeichneten Elemente gelöscht. Siehe auch **PxlCrcl**.

In einem ZoomSqr-Ansichtfenster:

```
ZoomSqr:Circle 1,2,3 ENTER
```



ClockOff CATALOG

ClockOff

Schaltet die Uhr AUS (engl. OFF).

ClockOn CATALOG

ClockOn

Schaltet die Uhr AN (engl. ON).

ClrDraw CATALOG

ClrDraw

Löscht den Graphikbildschirm und setzt Smart Graph zurück, so daß beim nächsten Anzeigen des Graphikbildschirms der Graph neu gezeichnet wird.

Bei angezeigtem Graphikbildschirm können Sie alle gezeichneten Elemente (etwa Linien und Punkte) wie folgt löschen: Drücken Sie F4 (ReGraph) oder:
[2nd] [F6]
[F6]
und wählen Sie 1:ClrDraw.

ClrErr CATALOG

ClrErr

Setzt den Fehlerstatus zurück: errornum wird auf Null gesetzt, und die internen Fehlerkontextvariablen werden gelöscht.

In der Verzweigung **Else** der Anweisung **Try...EndTry** sollte **ClrErr** oder **PassErr** benutzt werden. Verwenden Sie **ClrErr**, wenn der Fehler verarbeitet oder ignoriert werden soll. Ist unbekannt, wie der Fehler behandelt werden soll, übergeben Sie ihn mit **PassErr** an die nächste Fehlerbehandlungsroutine. Wenn keine weiteren **Try...EndTry** Routinen folgen, wird das Fehlerdialogfeld wie normal angezeigt.

Hinweis: Siehe auch **PassErr** und **Try**.

Programmlisting:

```
:clearerr()
:Prgm
:PlotsOff:FnOff:ZoomStd
:For i,0,238
:Δx*i+xmin→xcord
: Try
:  PtOn xcord,ln(xcord)
: Else
:  If errornum=800 or
     errornum=260 Then
:   ClrErr ◉ clear the error
:  Else
:   PassErr ◉ pass on any other
     error
:  EndIf
: EndTry
:EndFor
:EndPrgm
```

ClrGraph CATALOG

ClrGraph

Löscht alle Funktionen oder Terme, die mit dem Befehl **Graph** gezeichnet oder mit dem Befehl **Table** erstellt wurden vom Bildschirm. (Siehe **Graph** oder **Table**).

Alle vorher ausgewählten Y= Funktionen werden gezeichnet, wenn der Graph das nächste Mal angezeigt wird.

ClrHome CATALOG

ClrHome

Löscht alle Paare, die im Protokoll-Bereich des Hauptbildschirms als Eingaben **entry()** und Antworten **ans()** gespeichert sind. Löscht die aktuelle Eingabezeile nicht.

Während der Hauptbildschirm angezeigt ist, können Sie den Protokoll-Bereich löschen, indem Sie F1 drücken und 8:Clear Home wählen.

Bei Funktionen wie **solve()**, welche "willkürliche" Konstanten oder ganze Zahlen (@1, @2 etc.) ausgeben, setzt **ClrHome** den Index wieder auf 1 zurück.

ClrIO CATALOG

ClrIO

Löscht den Programm-I/O-Bildschirm.

ClrTable CATALOG

ClrTable

Löscht alle Tabellenwerte. Gilt nur für die Einstellung ASK des Dialogfelds Table Setup.

Bei angezeigtem Tabellenbildschirm können Sie im Ask-Modus die Werte wie folgt löschen: Drücken Sie F1, und wählen Sie 8:Clear Table.

colDim() MATH/Matrix/Dimensions-Menü

colDim(*Matrix***)** ⇒ *Term*

Gibt die Anzahl der Spalten von *Matrix* zurück.

Hinweis: Siehe auch **rowDim()**.

```
colDim([0,1,2;3,4,5]) ENTER      3
```

colNorm() MATH/Matrix/Norms-Menü

colNorm(*Matrix***)** ⇒ *Term*

Gibt das Maximum der Summen der absoluten Elementwerte der Spalten von *Matrix* zurück.

Hinweis: Undefinierte Matrixelemente sind nicht zulässig. Siehe auch **rowNorm()**.

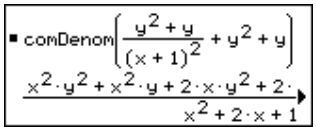
```
[1,-2,3;4,5,-6]→mat ENTER
                        [1  -2  3]
                        [4   5 -6]
colNorm(mat) ENTER               9
```

comDenom() MATH/Algebra-Menü

comDenom(*Term1*[,*Var*]**)** ⇒ *Term*
comDenom(*Liste1*[,*Var*]**)** ⇒ *Liste*
comDenom(*Matrix1*[,*Var*]**)** ⇒ *Matrix*

comDenom(*Term1***)** gibt den gekürzten Quotienten aus einem vollständig entwickelten Zähler und einem vollständig entwickelten Nenner zurück.

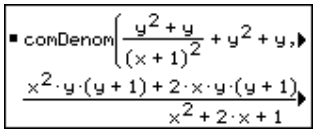
```
comDenom((y^2+y)/(x+1)^2+y^2+y
)
ENTER
```



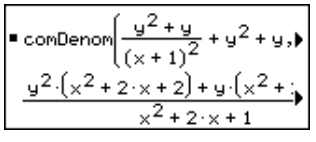
---

comDenom(*Term1,Var***)** gibt einen gekürzten Quotienten von Zähler und Nenner zurück, der bezüglich *Var* entwickelt wurde. Die Terme und Faktoren werden mit *Var* als der Hauptvariablen sortiert. Gleichartige Potenzen von *Var* werden zusammengefaßt. Es kann sein, dass als Nebeneffekt eine Faktorisierung der zusammengefaßten Koeffizienten auftritt. Verglichen mit dem Weglassen von *Var* spart dies häufig Zeit, Speicherplatz und Platz auf dem Bildschirm und macht den Term verständlicher. Außerdem werden anschließende Operationen an diesem Ergebnis schneller, und es wird weniger wahrscheinlich, daß der Speicherplatz ausgeht.

```
comDenom((y^2+y)/(x+1)
^2+y^2+y,x)
ENTER
```



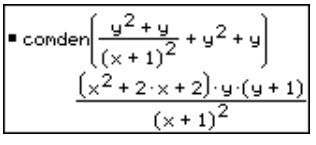

```
comDenom((y^2+y)/(x+1)
^2+y^2+y,y)
ENTER
```



Wenn *Var* nicht in *Term1* vorkommt, gibt **comDenom(***Term1,Var***)** einen gekürzten Quotienten eines nicht entwickelten Zählers und eines nicht entwickelten Nenners zurück. Solche Ergebnisse sparen meist sogar noch mehr Zeit, Speicherplatz und Platz auf dem Bildschirm. Solche partiell faktorisierten Ergebnisse machen ebenfalls anschließende Operationen mit dem Ergebnis schneller und das Erschöpfen des Speicherplatzes weniger wahrscheinlich.

```
comDenom(exprn,abc)→comden
(exprn)
ENTER                          Done

comden((y^2+y)/(x+1)^2+y^2+y)
ENTER
```



Sogar wenn kein Nenner vorhanden ist, ist die Funktion **comden** häufig ein gutes Mittel für das partielle Faktorisieren, wenn **factor()** zu langsam ist oder den Speicherplatz erschöpft.

Tipp: Geben Sie diese Funktionsdefinition **comden()** ein, und verwenden Sie sie regelmäßig als Alternative zu **comDenom()** und **factor()**.

```
comden(1234x^2*(y^3-y)+2468x*
(y^2-1))ENTER
```

$$1234 \cdot x \cdot (x \cdot y + 2) \cdot (y^2 - 1)$$

conj() MATH/Complex-Menü

conj(*Term1***)** ⇒ *Term*
conj(*Liste1***)** ⇒ *Liste*
conj(*Matrix1***)** ⇒ *Matrix*

Gibt das Konjugiert Komplexe des Parameters zurück.

Hinweis: Alle undefinierten Variablen werden als reelle Variablen behandelt.

```
conj(1+2i)ENTER
```

$$1 - 2 \cdot i$$

```
conj([2,1-3i;-i,-7])ENTER
```

$$\begin{bmatrix} 2 & 1+3 \cdot i \\ i & -7 \end{bmatrix}$$

```
conj(z)
```

$$z$$

```
conj(x+iy)
```

$$x + -i \cdot y$$

CopyVar CATALOG

CopyVar *Var1*, *Var2*

Kopiert den Inhalt von Variable *Var1* nach *Var2*. Existiert *Var2* noch nicht, wird sie von **CopyVar** erstellt.

Hinweis: CopyVar ist beim Kopieren eines Terms, einer Liste, einer Matrix oder einer Zeichenkette der Anweisung Speichern (→) ähnlich, wobei mit **CopyVar** jedoch keine Vereinfachung vorgenommen wird. Bei nicht-algebraischen Variablentypen, wie Pic- und GDB-Variablen müssen Sie **CopyVar** benutzen.

```
x+y→a ENTER                    x+y
10→x ENTER                     10
CopyVar a,b ENTER              Done
a→c ENTER                      y+10
DelVar x ENTER                 Done
b ENTER                        x+y
c ENTER                        y+10
```

cos() 2nd [COS] Taste COS Taste

cos(*Term1***)** ⇒ *Term*
cos(*Liste1***)** ⇒ *Liste*

cos(*Term1***)** gibt den Cosinus des Parameters als Term zurück.

cos(*Liste1***)** gibt in Form einer Liste für jedes Element aus *Liste1* den Cosinus zurück.

Hinweis: Der als Parameter angegebene Winkel wird gemäß der aktuellen Winkel-Moduseinstellung als Ordnung oder Bogenmaß interpretiert. Sie können ° oder ʳ benutzen, um temporär den anderen Winkel-Modus einzustellen.

Im Degree-Modus für Winkel:

```
cos((π/4)ʳ)ENTER
```

$$\frac{\sqrt{2}}{2}$$

```
cos(45)ENTER
```

$$\frac{\sqrt{2}}{2}$$

```
cos({0,60,90})ENTER    {1   1/2   0}
```

Im Radian-Modus:

```
cos(π/4)ENTER
```

$$\frac{\sqrt{2}}{2}$$

```
cos(45°)ENTER
```

$$\frac{\sqrt{2}}{2}$$

cos(*quadrat_Matrix1***)** ⇒ *quadrat_Matrix*

Gibt den Matrix-Cosinus von *quarat_Matrix1* zurück. Dies ist *nicht* gleichbedeutend mit der Berechnung des Cosinus jedes einzelnen Elements.

Wenn eine skalare Funktion f(A) auf *quadrat_Matrix1* (A) angewendet wird, erfolgt die Berechnung des Ergebnisses durch den Algorithmus:

1. Berechnung der Eigenwerte (λ_i) und Eigenvektoren (V_i) von A.

   Quadrat_Matrix1 muß diagonalisierbar sein. Sie darf auch keine symbolischen Variablen ohne zugewiesene Werte enthalten.

2. Bildung der Matrizen:

$$B = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & \lambda_n \end{bmatrix} \text{ und } X = [V_1, V_2, \dots, V_n]$$

3. Dann ist $A = X\,B\,X^{-1}$ und $f(A) = X\,f(B)\,X^{-1}$. Zum Beispiel: $\cos(A) = X\,\cos(B)\,X^{-1}$ wobei:

$$\cos(B) = \begin{bmatrix} \cos(\lambda_1) & 0 & \dots & 0 \\ 0 & \cos(\lambda_2) & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & \cos(\lambda_n) \end{bmatrix}$$

Alle Berechnungen werden unter Verwendung von Fließkomma-Operationen ausgeführt.

Im Radian-Modus:

`cos([1,5,3;4,2,1;6,⁻2,1])` ENTER

$$\begin{bmatrix} .212\dots & .205\dots & .121\dots \\ .160\dots & .259\dots & .037\dots \\ .248\dots & -.090\dots & .218\dots \end{bmatrix}$$

cos⁻¹() ▣ ◆[cos⁻¹] Taste ⌨ 2nd[cos⁻¹] Taste

cos⁻¹(*Term1***)** ⇒ *Term*
cos⁻¹(*Liste1***)** ⇒ *Liste*

cos⁻¹(*Term1***)** gibt den Winkel, dessen Cosinus *Term1* ist, als Term zurück.

cos⁻¹(*Liste1***)** gibt in Form einer Liste für jedes Element aus *Liste1* den inversen Cosinus zurück.

Hinweis: Das Ergebnis wird gemäß der aktuellen Winkel-Moduseinstellung entweder in Ordnung oder im Bogenmaß zurückgegeben.

Im Degree-Modus für Winkel:

`cos⁻¹(1)` ENTER 0

Im Radian-Modus:

`cos⁻¹({0,.2,.5})` ENTER

$\{\frac{\pi}{2}$ 1.369… 1.047…$\}$

cos⁻¹(*quadrat_Matrix1***)** ⇒ *quadrat_Matrix*

Gibt den inversen Matrix-Cosinus von *quadrat_Matrix1* zurück. Dies ist *nicht* gleichbedeutend mit der Berechnung des inversen Cosinus jedes einzelnen Elements. Näheres zur Berechnungsmethode finden Sie im Abschnitt **cos()**.

Quadrat_Matrix1 muß diagonalisierbar sein. Das Ergebnis enthält immer Fließkommazahlen.

Im Winkelmodus Radian und Komplex-Formatmodus "Rectangular":

`cos⁻¹([1,5,3;4,2,1;6,⁻2,1])` ENTER

$$\begin{bmatrix} 1.734\dots+.064\dots\cdot i & -1.490\dots+2.105\dots\cdot i & \dots \\ -.725\dots+1.515\dots\cdot i & .623\dots+.778\dots\cdot i & \dots \\ -2.083\dots+2.632\dots\cdot i & 1.790\dots-1.271\dots\cdot i & \dots \end{bmatrix}$$

cosh() MATH/Hyperbolic-Menü

cosh(*Term1***)** ⇒ *Term*
cosh(*Liste1***)** ⇒ *Liste*

cosh (*Term1***)** gibt den Cosinus hyperbolicus des Parameters als Term zurück.

cosh (*Liste1***)** gibt in Form einer Liste für jedes Element aus *Liste1* den Cosinus hyperbolicus zurück.

`cosh(1.2)` ENTER 1.810…

`cosh({0,1.2})` ENTER {1 1.810…}

cosh(*quadrat_Matrix1***)** ⇒ *quadrat_Matrix*

Gibt den Matrix-Cosinus hyperbolicus von *quadrat_Matrix1* zurück. Dies ist *nicht* gleichbedeutend mit der Berechnung des Cosinus hyperbolicus jedes einzelnen Elements. Näheres zur Berechnungsmethode finden Sie im Abschnitt **cos()**.

Quadrat_Matrix1 muß diagonalisierbar sein. Das Ergebnis enthält immer Fließkommazahlen.

Im Radian-Modus:

`cosh([1,5,3;4,2,1;6,-2,1])` ENTER

$$\begin{bmatrix} 421.255 & 253.909 & 216.905 \\ 327.635 & 255.301 & 202.958 \\ 226.297 & 216.623 & 167.628 \end{bmatrix}$$

$\cosh^{-1}()$ MATH/Hyperbolic-Menü

cosh^{-1}(*Term1***)** ⇒ *Term*
cosh^{-1}(*Liste1***)** ⇒ *Liste*

cosh^{-1}(*Term1***)** gibt den inversen coshyp des Parameters als Term zurück.

cosh^{-1}(*Liste1***)** gibt in Form einer Liste für jedes Element aus *Liste1* den inversen coshyp zurück.

`cosh-1(1)` ENTER 0

`cosh-1({1,2.1,3})` ENTER

{0 1.372… cosh^{-1}(3)}

cosh^{-1}(*quadrat_Matrix1***)** ⇒ *quadrat_Matrix*

Gibt den inversen Matrix-Cosinus hyperbolicus von *quadrat_Matrix1* zurück. Dies ist *nicht* gleichbedeutend mit der Berechnung des inversen Cosinus hyperbolicus jedes einzelnen Elements. Näheres zur Berechnungsmethode finden Sie im Abschnitt **cos()**.

Quadrat_Matrix1 muß diagonalisierbar sein. Das Ergebnis enthält immer Fließkommazahlen.

Im Winkelmodus Radian und Komplex-Formatmodus "Rectangular":

`cosh-1([1,5,3;4,2,1;6,-2,1])` ENTER

$$\begin{bmatrix} 2.525…+1.734…\cdot i & -.009…-1.490…\cdot i & … \\ .486…-.725…\cdot i & 1.662…+.623…\cdot i & … \\ -.322…-2.083…\cdot i & 1.267…+1.790…\cdot i & … \end{bmatrix}$$

cot() Menü MATH/Trig (MATHEMATIK/Trigonometrie)

cot(*Ausdruck1***)** ⇒ *Ausdruck*
cot(*Liste1***)** ⇒ *Liste*

Gibt den Kotangens von *Ausdruck1* oder eine Liste der Kotangens aller Elemente in *Liste1* zurück.

Hinweis: Das Argument wird entsprechend dem aktiven Winkelmodus als Winkel in Grad oder im Bogenmaß (Radian) interpretiert.

Im Grad-Modus:

`cot(45)` ENTER 1

Im Radian-Modus:

`cot({1,2.1,3})` ENTER

$$\frac{1}{\tan(1)} \quad -.584… \quad \frac{1}{\tan(3)}$$

$\cot^{-1}()$ Menü MATH/Trig (MATHEMATIK/Trigonometrie)

cot^{-1}(*Ausdruck1***)** ⇒ *Ausdruck*
cot^{-1}(*Liste1***)** ⇒ *Liste*

Gibt entweder den Winkel, dessen Kotangens *Ausdruck1* entspricht, oder eine Liste der inversen Kotangens aller Elemente in *Liste1* zurück.

Hinweis: Das Ergebnis wird entsprechend dem aktiven Winkelmodus als Winkel in Grad oder im Bogenmaß (Radian) ausgegeben.

Im Grad-Modus:

`cot-1(1)` ENTER 45

Im Radian-Modus:

`cot-1(1)` ENTER $\frac{\pi}{4}$

coth() Menü MATH/Hyperbolic (MATHEMATIK/Hyperbolisch)

coth(*Ausdruck1***)** ⇒ *Ausdruck*
cot(*Liste1***)** ⇒ *Liste*

Gibt den hyperbolischen Kotangens von *Ausdruck1* oder eine Liste der hyperbolischen Kotangens aller Elemente in *Liste1* zurück.

`coth(1.2)` ENTER 1.199…

`coth({1,3.2})` ENTER

$$\frac{1}{\tanh(1)} \quad 1.003…$$

coth⁻¹() Menü MATH/Hyperbolic (MATHEMATIK/Hyperbolisch)

coth⁻¹(*Ausdruck1***)** ⇒ *Ausdruck*
coth⁻¹(*Liste1***)** ⇒ *Liste*

Gibt den inversen hyperbolischen Kotangens von *Ausdruck1* oder eine Liste der inversen hyperbolischen Kotangens aller Elemente in *Liste1* zurück.

```
coth⁻¹(3.5) ENTER                                   .293…

coth⁻¹({-2,2.1,6}) ENTER
                          -ln(3)/2   .518…   ln(7/5)/2
```

crossP() MATH/Matrix/Vector ops-Menü

crossP(*Liste1***, *Liste2***)** ⇒ *Liste*

Gibt das Kreuzprodukt aus *Liste1* und *Liste2* in Form einer Liste zurück.

Liste1 und *Liste2* müssen die gleiche Dimension besitzen, und sie muß entweder 2 oder 3 sein.

```
crossP({a1,b1},{a2,b2}) ENTER
                                {0  0  a1·b2−a2·b1}

crossP({0.1,2.2,⁻5},{1,⁻.5,0})
ENTER
                                {⁻2.5  ⁻5.  ⁻2.25}
```

---

crossP(*Vektor1***, *Vektor2***)** ⇒ *Vektor*

Gibt einen Zeilen- oder Spaltenvektor zurück (ja nach den Parametern), der das Kreuzprodukt von *Vektor1* und *Vektor2* ist.

Entweder müssen *Vektor1* und *Vektor2* beide Zeilenvektoren oder beide Spaltenvektoren sein. Beide Vektoren müssen die gleiche Dimension besitzen, und sie muß entweder 2 oder 3 sein.

```
crossP([1,2,3],[4,5,6]) ENTER
                                      [⁻3  6  ⁻3]

crossP([1,2],[3,4]) ENTER
                                      [0  0  ⁻2]
```

csc() Menü MATH/Trig (MATHEMATIK/Trigonometrie)

csc(*Ausdruck1***)** ⇒ *Ausdruck*
csc(*Liste1***)** ⇒ *Liste*

Gibt den Kosekans von *Ausdruck1* oder eine Liste der Konsekans aller Elemente in *Liste1* zurück.

```
Im Grad-Modus:

csc(π/4) ENTER                                      1/sin(π/4)

Im Radian-Modus:

csc({1,π/2,π/3}) ENTER
                                   1/sin(1)   1   2·√3/3
```

csc⁻¹() Menü MATH/Trig (MATHEMATIK/Trigonometrie)

csc⁻¹(*Ausdruck1***)** ⇒ *Ausdruck*
csc⁻¹(*Liste1***)** ⇒ *Liste*

Gibt entweder den Winkel, dessen Kosekans *Ausdruck1* entspricht, oder eine Liste der inversen Kosekans aller Elemente in *Liste1* zurück.

Hinweis: Das Ergebnis wird entsprechend dem aktiven Winkelmodus als Winkel in Grad oder im Bogenmaß (Radian) ausgegeben.

```
Im Grad-Modus:

csc⁻¹(1) ENTER                                      90

Im Radian-Modus:

csc⁻¹({1,4,6}) ENTER
                          π/2   sin⁻¹(1/4)   sin⁻¹(1/6)
```

csch() Menü MATH/Hyperbolic (MATHEMATIK/Hyperbolisch)

csch(*Ausdruck1***)** ⇒ *Ausdruck*
csch(*Liste1***)** ⇒ *Liste*

Gibt den hyperbolischen Kosekans von *Ausdruck1* oder eine Liste der hyperbolischen Kosekans aller Elemente in *Liste1* zurück.

```
csch(3) ENTER                                       1/sinh(3)

csch({1,2.1,4}) ENTER
                                  1/sinh(1)   .248…   1/sinh(4)
```

csch⁻¹() Menü MATH/Hyperbolic (MATHEMATIK/Hyperbolisch)

csch⁻¹(*Ausdruck1***)** ⇒ *Ausdruck*
csch⁻¹(*Liste1***)** ⇒ *Liste*

Gibt den inversen hyperbolischen Kosekans von *Ausdruck1* oder eine Liste der inversen hyperbolischen Kosekans aller Elemente in *Liste1* zurück.

csch⁻¹(1) ENTER sinh⁻¹(1)

csch⁻¹({1,2.1,3}) ENTER

sinh⁻¹(1) .459… sinh⁻¹(1/3)

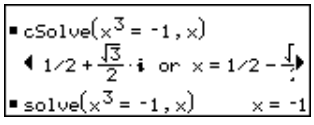
cSolve() MATH/Algebra/Complex-Menü

cSolve(*Gleichung***,** *Var***)** ⇒ *Boolescher Term*

Gibt mögliche komplexe Lösungen einer Gleichung für *Var* zurück. Das Ziel ist, Kandidaten für alle reellen und nicht-reellen Lösungen zu erhalten. Selbst wenn *Gleichung* reell ist, erlaubt **cSolve()** nicht-reelle Lösungen im reellen Modus.

Obwohl der TI-89 Titanium / Voyage™ 200 alle undefinierten Variablen so verarbeitet als wären sie reell, kann **cSolve()** Polynomgleichungen für komplexe Lösungen lösen.

cSolve(x^3=-1,x) ENTER
solve(x^3=-1,x) ENTER



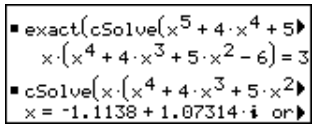
cSolve() setzt den Bereich während der Berechnung zeitweise auf komplex, auch wenn der aktuelle Bereich reell ist. Im Komplexen benutzen Bruchexponenten mit ungeradem Nenner den Hauptzweig und sind nicht reell. Demzufolge sind Lösungen mit **solve()** für Gleichungen, die solche Bruchexponenten besitzen, nicht unbedingt eine Teilmenge der mit **cSolve()** erzielten Lösungen.

cSolve(x^(1/3)=-1,x) ENTER false

solve(x^(1/3)=-1,x) ENTER x=-1

cSolve() beginnt mit exakten symbolischen Verfahren. Außer im Modus EXACT, benutzt **cSolve()** bei Bedarf auch die iterative näherungsweise polynomische Faktorisierung.

Stellenanzeigemodus (Anzeige Digits) auf Fix 2:
exact(cSolve(x^5+4x^4+5x^3-6x-3=0,x)) ENTER
cSolve(ans(1),x) ENTER



Hinweis: Siehe auch **cZeros()**, **solve()** und **zeros()**.

Hinweis: Enthält *Gleichung* Funktionen wie beispielsweise **abs()**, **angle()**, **conj()**, **real()** oder **imag()**, ist sie Unterstrich also kein Polynom, sollten Sie hinter *Var* ein zeichen _ (▤ ◆ [_], ⌨ 2nd [_]) setzen. Standardmäßig wird eine Variable als reeller Wert behandelt. Bei Verwendung von *Var_* wird die Variable als komplex behandelt.

Sie sollten *Var_* auch für alle anderen Variablen in *Gleichung* verwenden, die nicht-reelle Werte haben könnten. Anderenfalls erhalten Sie möglicherweise unerwartete Ergebnisse.

z wird als reell behandelt:

cSolve(conj(z)=1+*i*,z) ENTER z=1+*i*

z_ wird als komplex behandelt:

cSolve(conj(z_)=1+*i*,z_) ENTER
z_=1−*i*

---

cSolve(*Gleichung1* **and** *Gleichung2* [**and** *…*]**,**
{*VarOderSchätzwert1***,** *VarOderSchätzwert2* [**,** *…*]**})**
⇒ *Boolescher Term*

Gibt mögliche komplexe Lösungen eines algebraischen Gleichungssystems zurück, in dem jedes *VarOderSchätzung* eine Variable darstellt, nach der Sie die Gleichungen auflösen möchten.

Sie haben die Option, eine Ausgangsschätzung für eine Variable anzugeben. *VarOderSchätzung* muß immer folgende Form haben:

Variable
– oder –
Variable = *reelle oder nicht-reelle Zahl*

Beispiel: x ist gültig, und x=3+*i* ebenfalls.

Wenn alle Gleichungen Polynome sind und Sie KEINE Anfangsschätzwerte angeben, dann verwendet **cSolve()** das lexikalische Gröbner/Buchbergersche Eliminationsverfahren beim Versuch, **alle** komplexen Lösungen zu bestimmen.

Hinweis: In folgenden Beispielen wird ein Unterstrichzeichen _ (▣ ◆ [_], ⌨ 2nd [_]) verwendet, damit die Variablen als komplex behandelt werden.

Komplexe Lösungen können, wie aus nebenstehendem Beispiel hervorgeht, sowohl reelle als auch nicht-reelle Lösungen enthalten.

```
cSolve(u_*v_-u_=v_ and
v_^2=-u_,{u_,v_}) ENTER
```

$$u_=1/2+\frac{\sqrt{3}}{2}\cdot i \text{ and } v_=1/2-\frac{\sqrt{3}}{2}\cdot i$$

$$\text{or } u_=1/2-\frac{\sqrt{3}}{2}\cdot i \text{ and } v_=1/2+\frac{\sqrt{3}}{2}\cdot i$$

$$\text{or } u_=0 \text{ and } v_=0$$

Gleichungssysteme, die aus *Polynomen* bestehen, können zusätzliche Variablen ohne Wert aufweisen, die aber für numerische Werte stehen, welche später eingesetzt werden können.

```
cSolve(u_*v_-u_=c_*v_ and
v_^2=-u_,{u_,v_}) ENTER
```

$$u_=\frac{-(\sqrt{1-4\cdot c_}+1)^2}{4} \text{ and } v_=\frac{\sqrt{1-4\cdot c_}+1}{2}$$

or

$$u_=\frac{-(\sqrt{1-4\cdot c_}-1)^2}{4} \text{ and } v_=\frac{-(\sqrt{1-4\cdot c_}-1)}{2}$$

$$\text{or } u_=0 \text{ and } v_=0$$

Sie können auch Lösungsvariablen angeben, die in der Gleichung nicht erscheinen. Diese Lösungen verdeutlichen, daß Lösungsfamilien "willkürliche" Konstanten der Form *@k* enthalten können, wobei *k* ein ganzzahliger Index im Bereich 1 bis 255 ist. Der Index wird wieder auf 1 zurückgesetzt, wenn Sie **ClrHome** oder F1 8:Clear Home verwenden.

```
cSolve(u_*v_-u_=v_ and
v_^2=-u_,{u_,v_,w_}) ENTER
```

$$u_=1/2+\frac{\sqrt{3}}{2}\cdot i \text{ and } v_=1/2-\frac{\sqrt{3}}{2}\cdot i \text{ and } w_=@1$$

$$\text{or } u_=1/2-\frac{\sqrt{3}}{2}\cdot i \text{ and } v_=1/2+\frac{\sqrt{3}}{2}\cdot i \text{ and } w_=@1$$

$$\text{or } u_=0 \text{ and } v_=0 \text{ and } w_=@1$$

Bei Gleichungssystemen aus Polynomen kann die Berechnungsdauer oder Speicherbelastung stark von der Reihenfolge abhängen, in welcher Sie die Lösungsvariablen angeben. Übersteigt Ihre erste Wahl die Speicherkapazität oder Ihre Geduld, versuchen Sie, die Variablen in der Gleichung und/oder *VarOderSchätzungswert*-Liste umzuordnen.

Wenn Sie keine Schätzwerte angeben und eine Gleichung in einer Variablen nicht-polynomisch ist, aber alle Gleichungen in allen Lösungsvariablen linear sind, so verwendet **cSolve()** das Gaußsche Eliminationsverfahren beim Versuch, alle Lösungen zu bestimmen.

```
cSolve(u_+v_=e^(w_) and u_-v_=
i, {u_,v_}) ENTER
```

$$u_=\frac{e^{w_}}{2}+1/2\cdot i \text{ and } v_=\frac{e^{w_}-i}{2}$$

Wenn ein System weder in all seinen Variablen polynomial noch in seinen Lösungsvariablen linear ist, dann bestimmt **cSolve()** mindestens eine Lösung anhand eines iterativen näherungsweisen Verfahrens. Hierzu muß die Anzahl der Lösungsvariablen gleich der Gleichungsanzahl sein, und alle anderen Variablen in den Gleichungen müssen zu Zahlen vereinfachbar sein.

```
cSolve(e^(z_)=w_ and w_=z_^2,
{w_,z_}) ENTER
```

$$w_=.494\ldots \text{ and } z_=-.703\ldots$$

Zur Bestimmung einer nicht-reellen Lösung ist häufig ein nicht-reeller Schätzwert erforderlich. Für Konvergenz sollte ein Schätzwert ziemlich nahe bei einer Lösung liegen.

```
cSolve(e^(z_)=w_ and w_=z_^2,
{w_,z_=1+i}) ENTER
```

$$w_=.149+4.891\cdot i \text{ and } z_=1.588...+1.540...+\cdot i$$

CubicReg MATH/Statistics/Regressions-Menü

CubicReg *Liste1*, *Liste2*[, [*Liste3*] [, *Liste4*, *Liste5*]]

Berechnet die kubische polynomische Regression und aktualisiert alle Statistikvariablen.

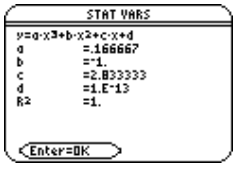
Alle Listen außer *Liste5* müssen die gleiche Dimension besitzen.

Liste1 stellt die Liste der x-Werte dar.
Liste2 stellt die Liste der y-Werte dar.
Liste3 stellt die Angaben für Häufigkeit dar.
Liste4 stellt die Klassencodes dar.
Liste5 stellt die Klassenliste dar.

Hinweis: *Liste1* bis einschl. *Liste4* müssen Variablenamen oder c1–c99 sein (Spalten in der letzten Datenvariablen, die im Daten/ Matrix-Editor angezeigt wurde). *Liste5* braucht kein Variablenname zu sein und kann nicht die Spalte c1–c99 sein.

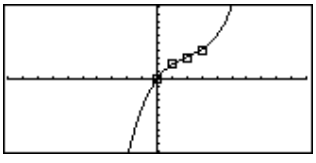
Im Funktions-Graphikmodus.

```
{0,1,2,3}→L1 ENTER        {0 1 2 ...}
{0,2,3,4}→L2 ENTER        {0 2 3 ...}
CubicReg L1,L2 ENTER          Done
ShowStat ENTER
```




```
ENTER
regeq(x)→y1(x) ENTER          Done
NewPlot 1,1,L1,L2 ENTER       Done
```

◆ [GRAPH]



cumSum() MATH/List-Menü

cumSum(*Liste1***)** ⇒ *Liste*

Gibt eine Liste der kumulierten Summen der Elemente aus *Liste1* zurück, wobei bei Element 1 begonnen wird.

```
cumSum({1,2,3,4}) ENTER {1 3 6 10}
```

---

cumSum(*Matrix1***)** ⇒ *Matrix*

Gibt eine Matrix der kumulierten Summen der Elemente aus *Matrix1* zurück. Jedes Element ist die kumulierte Summe der Spalte von oben nach unten.

```
[1,2;3,4;5,6]→m1 ENTER
```

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$$

```
cumSum(m1) ENTER
```

$$\begin{bmatrix} 1 & 2 \\ 4 & 6 \\ 9 & 12 \end{bmatrix}$$

CustmOff CATALOG

CustmOff

Entfernt eine benutzerspezifische Menüleiste.

CustmOn und **CustmOff** aktivieren ein Programm zur Steuerung einer benutzerspezifischen Menüleiste. Zum manuellen Ein- und Ausschalten einer benutzerspezifischen Menüleiste können Sie 2nd [CUSTOM] drücken. Eine benutzerspezifische Menüleiste wird außerdem automatisch entfernt, wenn Sie die Anwendung wechseln.

Siehe Beispiel für Programmlisting **Custom**.

CustmOn CATALOG

CustmOn

Aktiviert eine benutzerspezifische Menüleiste, die bereits in einem **Custom...EndCustm**-Block eingerichtet wurde.

CustmOn und **CustmOff** aktivieren ein Programm zur Steuerung einer benutzerspezifischen Menüleiste. Zum manuellen Ein- und Ausschalten einer benutzerspezifischen Menüleiste können Sie 2nd [CUSTOM] drücken.

Siehe Beispiel für Programmlisting **Custom**.

Custom 2nd [CUSTOM]-Taste

Custom
Block
EndCustm

Richtet eine Menüleiste ein, die Sie durch Drücken von 2nd [CUSTOM] aktivieren. Ist sehr ähnlich der Anweisung **ToolBar** allerdings können Title und Item keine Marken besitzen.

Block kann eine einzelne Anweisung oder eine Serie von Anweisungen sein, die durch ":" getrennt sind.

Hinweis: 2nd [CUSTOM] wirkt als Schalter. Das erstmalige Drücken ruft das Menü auf, das zweite Drücken setzt es wieder außer Kraft. Das Menü wird auch entfernt, wenn Sie die Anwendung wechseln.

Programmlisting:

```
:Test()
:Prgm
:Custom
:Title      "Lists"
:Item       "List1"
:Item       "Scores"
:Item       "L3"
:Title      "Fractions"
:Item       "f(x)"
:Item       "h(x)"
:Title      "Graph"
:EndCustm
:EndPrgm
```

Cycle CATALOG

Cycle

Übergibt die Programmsteuerung sofort an die nächste Wiederholung der aktuellen Schleife (**For**, **While** oder **Loop**).

Cycle ist außerhalb dieser drei Schleifenstrukturen (**For**, **While** oder **Loop**) nicht zulässig.

Programmlisting:

```
:◉ Sum the integers from 1 to
100
  skipping 50.
:0→temp
:For i,1,100,1
:If i=50
:Cycle
:temp+i→temp
:EndFor
:Disp temp
```

Inhalt von temp nach Ausführung: 5000

CyclePic CATALOG

CyclePic *picNameString*, *n* [, [*warten*] , [*Zyklen*], [*Richtung*]]

Zeigt alle angegebenen PIC-Variablen mit dem angegebenen Intervall. Der Benutzer kann die Zeit zwischen den Bildern, die Anzahl der Anzeigezyklen und die Richtung (umlaufend oder vorwärts und rückwärts) festlegen.

Richtung ist 1 für "umlaufend" und ⁻1 für vorwärts und rückwärts. Vorgabe = 1.

1. Speichern Sie drei Bilder mit den Namen pic1, pic2 und pic3.
2. Geben Sie ein: **CyclePic "pic",3,.5,4,⁻1**
3. Die drei Bilder (3) werden automatisch angezeigt—mit einer halben Sekunde (.5) zwischen den Bildern, vier Zyklen (4) lang sowie vorwärts -rückwärts (⁻1).

▸Cylind MATH/Matrix/Vector ops-Menü

Vektor ▸**Cylind**

Zeigt den Zeilen- oder Spaltenvektor in Zylinderkoordinaten [r∠θ, z] an.

Vektor muß genau drei Elemente besitzen. Er kann entweder ein Zeile oder eine Spalte sein.

[2,2,3] ▸Cylind ENTER

$$[2\cdot\sqrt{2}\ \angle\frac{\pi}{4}\quad 3]$$

cZeros() MATH/Algebra/Complex-Menü

cZeros(*Term***,** *Var***)** ⇒ *Liste*

Gibt eine Liste möglicher reeller und nicht-reeller Werte für *Var* zurück, die *Term*=0 ergeben. **cZeros()** tut dies durch Berechnung von **exp▸list(cSolve(***Term***=0,***Var***),***Var***)**. Ansonsten ist **cZeros()** ähnlich wie **zeros()**.

Hinweis: Siehe auch **cSolve()**, **solve()** und **zeros()**.

Stellenanzeigemodus (Anzeige Digits) auf Fix 3:

cZeros(x^5+4x^4+5x^3−6x−3,x) ENTER

{⁻2.125 ⁻.612 .965
⁻1.114−1.073·*i*
⁻1.114+1.073·*i*}

Hinweis: Ist *Term* nicht-polynomial mit Funktionen wie beispielsweise **abs()**, **angle()**, **conj()**, **real()** oder **imag()**, sollten Sie hinter *Var* ein Unterstrichzeichen _ (▦ ◆ [_], ⌨ 2nd [_]) setzen. Standardmäßig wird eine Variable als reeller Wert behandelt. Bei Verwendung von *Var_* wird die Variable als komplex behandelt.

Sie sollten *Var_* auch für alle anderen Variablen in *Term* verwenden, die nicht-reelle Werte haben könnten. Anderenfalls erhalten Sie möglicherweise unerwartete Ergebnisse.

z wird als reell behandelt:

```
cZeros(conj(z)-1-i,z) ENTER
                              {1+i}
```

z_ wird als komplex behandelt:

```
cZeros(conj(z_)-1-i,z_) ENTER
                              {1-i}
```

---

cZeros({*Term1***,** *Term2* [**,** *…*]**}, {***VarOderSchätzwert1***,** *VarOderSchätzwert2* [**,** *…*]**})** ⇒ *Matrix*

Gibt mögliche Positionen zurück, in welchen die Terme gleichzeitig Null sind. Jeder *VarOderSchätzwert* steht für eine Unbekannte, deren Wert Sie suchen.

Sie haben auch die Option, einen Ausgangsschätzwert für eine Variable anzugeben. *VarOderSchätzwert* muß immer folgende Form haben:

Variable
– oder –
Variable = reelle oder nicht-reelle Zahl

Beispiel: x ist gültig, und x=3+*i* ebenfalls.

Wenn alle Terme Polynome sind und Sie KEINE Anfangsschätzwerte angeben, dann verwendet **cZeros()** das lexikalische Gröbner/Buchbergersche Eliminationsverfahren beim Versuch, **alle** komplexen Nullstellen zu bestimmen.

Komplexe Nullstellen können, wie aus nebenstehendem Beispiel hervorgeht, sowohl reelle als auch nicht-reelle Nullstellen enthalten.

Jede Zeile der sich ergebenden Matrix stellt eine alternative Nullstelle dar, wobei die Komponenten in derselben Reihenfolge wie in der *VarOderSchätzwert*-Liste angeordnet sind. Um eine Zeile zu erhalten ist die Matrix nach [*Zeile*] zu indizieren.

Hinweis: In folgenden Beispielen wird ein Unterstreichungszeichen _ (▦ ◆ [_], ⌨ 2nd [_]) verwendet, damit die Variablen als komplex behandelt werden.

```
cZeros({u_*v_-u_-v_,v_^2+u_},
{u_,v_}) ENTER
```

$$\begin{bmatrix} 1/2 - \frac{\sqrt{3}}{2}\cdot i & 1/2 + \frac{\sqrt{3}}{2}\cdot i \\ 1/2 + \frac{\sqrt{3}}{2}\cdot i & 1/2 - \frac{\sqrt{3}}{2}\cdot i \\ 0 & 0 \end{bmatrix}$$

Zeile 2 extrahieren:

```
ans(1)[2] ENTER
```

$$\begin{bmatrix} 1/2 + \frac{\sqrt{3}}{2}\cdot i & 1/2 - \frac{\sqrt{3}}{2}\cdot i \end{bmatrix}$$

Gleichungssysteme, die aus *Polynomen* bestehen, können zusätzliche Variablen haben, die zwar ohne Werte sind, aber gegebene numerische Werte darstellen, die später eingesetzt werden können.

```
cZeros({u_*v_-u_-(c_*v_),
v_^2+u_},{u_,v_}) ENTER
```

$$\begin{bmatrix} \frac{-(\sqrt{1-4\cdot c_}+1)^2}{4} & \frac{\sqrt{1-4\cdot c_}+1}{2} \\ \frac{-(\sqrt{1-4\cdot c_}-1)^2}{4} & \frac{-(\sqrt{1-4\cdot c_}-1)}{2} \\ 0 & 0 \end{bmatrix}$$

Sie können auch unbekannte Variablen angeben, die nicht in den Termen erscheinen. Diese Nullstellen verdeutlichen, daß Nullstellenfamilien "willkürliche" Konstanten der Form *@k* enthalten können, wobei *k* ein ganzzahliger Index im Bereich 1 bis 255 ist. Der Index wird wieder auf 1 zurückgesetzt, wenn Sie **ClrHome** oder F1 8:Clear Home verwenden.

```
cZeros({u_*v_-u_-v_,v_^2+u_},
{u_,v_,w_}) ENTER
```

$$\begin{bmatrix} 1/2 - \frac{\sqrt{3}}{2}\cdot i & 1/2 + \frac{\sqrt{3}}{2}\cdot i & @1 \\ 1/2 + \frac{\sqrt{3}}{2}\cdot i & 1/2 - \frac{\sqrt{3}}{2}\cdot i & @1 \\ 0 & 0 & @1 \end{bmatrix}$$

Bei polynomialen Gleichungssystemen kann die Berechnungsdauer oder Speicherbelastung stark von der Reihenfolge abhängen, in welcher Sie die Unbekannten angeben. Übersteigt Ihre erste Wahl die Speicherkapazität oder Ihre Geduld, versuchen Sie, die Variablen in den Termen und/oder der *varOderRaten*-Liste umzuordnen.

Wenn Sie keine Schätzwerte angeben und ein Term in einer Variablen nicht-polynomial ist, aber alle Terme in allen Unbekannten linear sind, so verwendet **cZeros()** das Gaußsche Eliminationsverfahren beim Versuch, alle Nullstellen zu bestimmen.

```
cZeros({u_+v_-e^(w_),u_-v_-i},
{u_,v_}) ENTER
```

$$\begin{bmatrix} \frac{e^{w_}}{2} + 1/2\cdot i & \frac{e^{w_}-i}{2} \end{bmatrix}$$

Wenn ein System weder in all seinen Variablen polynomial noch in seinen Unbekannten linear ist, dann bestimmt, **cZeros()** mindestens eine Nullstelle anhand eines iterativen Näherungsverfahrens. Hierzu muß die Anzahl der Unbekannten gleich der Termanzahl sein, und alle anderen Variablen in den Termen müssen zu Zahlen vereinfachbar sein.

```
cZeros({e^(z_)-w_,w_-z_^2},
{w_,z_}) ENTER
```

$$\begin{bmatrix} .494\ldots & -.703\ldots \end{bmatrix}$$

Zur Bestimmung einer nicht-reellen Nullstelle ist häufig ein nicht-reeller Schätzwert erforderlich. Für Konvergenz muß ein Schätzwert ziemlich nahe bei der Nullstelle liegen.

```
cZeros({e^(z_)-w_,w_-z_^2},
{w_,z_=1+i}) ENTER
```

$$\begin{bmatrix} .149\ldots+4.89\ldots\cdot i & 1.588\ldots+1.540\ldots\cdot i \end{bmatrix}$$

d() 2nd [*d*] -Taste oder MATH/Calculus-Menü

d*(Term1*, *Var* [,*Ordnung*]) ⇒ *Term***
d*(Liste1*,*Var* [,*Ordnung*]) ⇒ *Liste***
d*(Matrix1*,*Var* [,*Ordnung*]) ⇒ *Matrix***

Gibt die erste Ableitung von *Term1* bezüglich der Variablen *Var* zurück. *Term1* kann eine Liste oder eine Matrix sein.

Ordnung ist optional und muß, wenn angegeben, eine ganze Zahl sein. Wenn sie kleiner Null ist, ist das Ergebnis eine unbestimmte Ableitung.

***d*()** folgt nicht dem normalen Auswertungsmechanismus der vollständigen Vereinfachung der Parameter mit anschließender Anwendung der Funktionsdefinition auf die vollständig vereinfachten Parameter. Statt dessen führt ***d*()** die folgenden Schritte aus:

1. Vereinfachung des zweiten Parameters nur soweit, daß sich keine nicht-Variable (keine Zahl) ergibt.
2. Vereinfachung des ersten Parameters nur soweit, daß er keinen gespeicherten Wert für die in Schritt 1 ermittelte Variable abruft.
3. Ermittlung der symbolischen Ableitung des Ergebnisses von Schritt 2 bezüglich der Variablen aus Schritt 1.
4. Setzen Sie, wenn die Variable aus Schritt 1 einen Wert oder einen mit dem "mit"-Operator (|) angegebenen Wert gespeichert hat, diesen Wert in das Ergebnis aus Schritt 3 ein.

`d(3x^3-x+7,x)` ENTER → $9x^2-1$

`d(3x^3-x+7,x,2)` ENTER → $18\cdot x$

`d(f(x)*g(x),x)` ENTER

$$\frac{d}{dx}(f(x))\cdot g(x)+\frac{d}{dx}(g(x))\cdot f(x)$$

`d(sin(f(x)),x)` ENTER

$$\cos(f(x))\frac{d}{dx}(f(x))$$

`d(x^3,x)|x=5` ENTER → 75

`d(d(x^2*y^3,x),y)` ENTER → $6\cdot y^2\cdot x$

`d(x^2,x,-1)` ENTER → $\frac{x^3}{3}$

`d({x^2,x^3,x^4},x)` ENTER → {2·x 3·x² 4·x³}

dayOfWk() CATALOG

dayOfWk(*year,month,day***)** ⇒ *integer*

Gibt eine ganze Zahl (Integer) zwischen 1 und 7 zurück, welche den Wochentag darstellt. Bestimmen Sie mit **dayOfWk()** auf welchem Wochentag ein bestimmtes Datum fällt.
Hinweis: Kann für Jahre vor 1583 fehlerhafte Ergebnisse liefern (vor-Gregorianischer Kalender).

Geben Sie das Jahr als vierstelligen Integer ein. Monat und Tag können ein- oder zweistellige Integer sein.

`dayOfWk(1948,9,6)` → 2

Integer-Werte:

1 = Sonntag
2 = Montag
3 = Dienstag
4 = Mittwoch
5 = Donnerstag
6 = Freitag
7 = Samstag

▸DD MATH/Angle-Menü

Zahl **▸DD** ⇒ *Wert*
Liste1 **▸DD** ⇒ *Liste*
Matrix1 **▸DD** ⇒ *Matrix*

Gibt das Dezimaläquivalent des Parameters zurück. Der Parameter ist eine Zahl, eine Liste oder eine Matrix, die gemäß der Moduseinstellung als Ordnung oder Bogenmaß interpretiert wird.

Hinweis: ▸DD kann auch eine Eingabe in rad akzeptieren.

Im Degree-Modus für Winkel:

`1.5° ▸DD` ENTER → 1.5°

`45° 22'14.3" ▸DD` ENTER → 45.370...°

`{45° 22'14.3",60° 0'0"} ▸DD` ENTER → {45.370... 60}°

Im Radian-Modus:

`1.5 ▸DD` ENTER → 85.9°

▸Dec MATH/Base-Menü

Ganze_Zahl1 **▸Dec** ⇒ *Ganze_Zahl*

Konvertiert *Ganze_Zahl1* in eine Dezimalzahl (Basis 10). Ein binärer oder hexadezimaler Eintrag muß stets das Präfix 0b bzw. 0h aufweisen.

Null (nicht Buchstabe O) und b oder h.

0b *binäre_Zahl*
0h *hexadezimale_Zahl*

Eine Dualzahl kann bis zu 32 Stellen aufweisen, eine Hexadezimalzahl bis zu 8.

Ohne Präfix wird *Ganze_Zahl1* als dezimal behandelt. Das Ergebnis wird unabhängig vom Base-Modus dezimal angezeigt.

```
0b10011▸Dec ENTER                    19
0h1F▸Dec ENTER                       31
```

Define CATALOG

Define *FunkName*(*Para1Name, Para2Name, ...*) = *Term*

Erzeugt *FunkName* als benutzerdefinierte Funktion. Sie können dann *FunkName*() genau wie die integrierten Funktionen benutzen. Die Funktion wertet *Term* unter Verwendung der übergebenen Parameter aus und gibt das Ergebnis zurück.

FunkName darf nicht der Name einer Systemvariablen oder einer integrierten TI-89 Titanium-Funktion sein.

Die Parameternamen sind Platzhalter. Verwenden Sie diese Namen nicht als Übergabeparameter, wenn Sie mit der Funktion arbeiten.

Hinweis: Diese Form von **Define** ist gleichwertig mit der Ausführung folgender Anweisung: *Term*→ *FunkName*(*Para1Name* ,*Para2Name*). Dieser Befehl kann auch zum Definieren einfacher Variablen benutzt werden, z. B. Define a=3.

```
Define g(x,y)=2x-3y ENTER            Done
g(1,2) ENTER                          -4
1→a:2→b:g(a,b) ENTER                  -4

Define h(x)=when(x<2,2x-3,
-2x+3) ENTER                         Done


h(-3) ENTER                           -9
h(4) ENTER                            -5


Define eigenvl(a)=
  cZeros(det(identity(dim(a)
  [1])-x*a),x) ENTER                 Done
eigenvl([-1,2;4,3]) ENTER
```

$$\left\{\frac{2\cdot\sqrt{3}-1}{11}\quad\frac{-(2\cdot\sqrt{3}+1)}{11}\right\}$$

Define *FunkName*(*Para1Name, Para2Name, ...*) = **Func**
Block
EndFunc

Ist identisch mit der vorstehenden Form von **Define** mit dem Unterschied, daß hier die benutzerdefinierte Funktion *FunkName*() einen Block aus mehreren Anweisungen ausführen kann.

Block kann eine einzelne Anweisung oder eine Serie von Anweisungen sein, die durch ":" getrennt sind. *Block* kann auch Ausdrücke und Anweisungen enthalten (wie **If**, **Then**, **Else** und **For**). Damit kann die Funktion *FunkName*() auch die Anweisung **Return** für die Rückgabe eines spezifischen Ergebnisses benutzen.

Hinweis: Es ist in der Regel einfacher, diese Art der Funktion mit dem Programmeditor statt in der Eingabezeile zu schreiben und zu bearbeiten.

```
Define g(x,y)=Func:If x>y Then
:Return x:Else:Return y:EndIf
:EndFunc ENTER                       Done

g(3,-7) ENTER                         3
```

Define *ProgName*(*Para1Name, Para2Name, ...*) **= Prgm**
Block
EndPrgm

Erzeugt *ProgName* als Programm oder Unterprogramm, kann jedoch kein Ergebnis mit **Return** zurückgeben. Kann einen Block aus mehreren Anweisungen ausführen.

Block kann eine einzelne Anweisung oder eine Serie von Anweisungen sein, die durch ":" getrennt sind. *Block* kann auch ohne Beschränkungen Ausdrücke und Anweisungen enthalten (wie **If**, **Then**, **Else** und **For**).

Hinweis: Es ist in der Regel einfacher, ein Programm mit dem Programmeditor statt in der Eingabezeile zu schreiben und zu bearbeiten.

```
Define listinpt()=prgm:Local
  n,i,str1,num:InputStr
  "Listennamen
  eingeben",str1:Input "Anzahl
  d. Elemente",n:For
  i,1,n,1:Input "Element
  "&string(i),num:
  num→#str1[i]:EndFor:EndPrgm
  ENTER
                                Done

listinpt() ENTER
                  Enter name of list
```

DelFold CATALOG

DelFold *Verzeichnisname1*[**,** *Verzeichnisname2*] [**,** *Verzeichnisname3*] **...**

Löscht die benutzerdefinierten Verzeichnisse *Verzeichnisname1, Verzeichnisname2* etc. Enthält ein Verzeichnis Variablen, wird eine Fehlermeldung angezeigt.

Hinweis: Das Verzeichnis main können Sie nicht löschen.

```
NewFold games ENTER                Done
(erzeugt das Verzeichnis games)

DelFold games ENTER                Done
(löscht das Verzeichnis games)
```

DelVar CATALOG

DelVar *Var1*[**,** *Var2*] [**,** *Var3*] **...**

Löscht die angegebenen Variablen aus dem Speicher.

```
2→ a ENTER                          2

(a+2)^2 ENTER                      16

DelVar a ENTER                     Done

(a+2)^2 ENTER                   (a + 2)²
```

deSolve() MATH/Calculus-Menü

deSolve(*DG1oder2Ordnung***,** *unabhängigeVar***,**
*abhängigeVar***)** ⇒ *eine allgemeine Lösung*

Ergibt eine Gleichung, die explizit oder implizit eine allgemeine Lösung für die gewöhnliche Differentialgleichung erster oder zweiter Ordnung angibt (DG). In der DG:

- Verwenden Sie einen Strich (', drücken Sie 2nd [′]), um die erste Ableitung der abhängigen Variablen gegenüber der unabhängigen Variabeln zu kennzeichnen.
- Kennzeichnen Sie die entsprechende zweite Ableitung mit zwei Strichen.

Das Zeichen ' wird nur für Ableitungen innerhalb von **deSolve()** verwendet. Verwenden Sie für andere Fälle ***d*()**.

Die allgemeine Lösung einer Gleichung erster Ordnung enthält eine willkürliche Konstante der Form *@k*, wobei *k* ein ganzzahliger Index im Bereich 1 bis 255 ist. Der Index wird wieder auf 1 zurückgesetzt, wenn Sie **ClrHome** oder 8:Clear Home verwenden. Die Lösung einer Gleichung zweiter Ordnung enthält zwei derartige Konstanten.

Wenden Sie **solve()** auf eine implizite Lösung an, wenn Sie versuchen möchten, diese in eine oder mehrere äquivalente explizite Lösungen zu konvertieren.

Beachten Sie beim Vergleich Ihrer Ergebnisse mit Lehrbuch- oder Handbuchlösungen bitte, daß die willkürlichen Konstanten in den verschiedenen Verfahren an unterschiedlichen Stellen in der Rechnung eingeführt werden, was zu unterschiedlichen allgemeinen Lösungen führen kann.

Hinweis: Zur Eingabe eines Strichs (') drücken Sie 2nd [′].

```
deSolve(y''+2y'+y=x^2,x,y)
ENTER
```

$$y=(@1\cdot x+@2)\cdot e^{-x}+x^2-4\cdot x+6$$

```
right(ans(1))→temp ENTER
```

$$(@1\cdot x+@2)\cdot e^{-x}+x^2-4\cdot x+6$$

```
d(temp,x,2)+2*d(temp,x)+temp-x^2
ENTER
```

$$0$$

```
delVar temp ENTER
```

Done

```
deSolve(y'=(cos(y))^2*x,x,y)
ENTER
```

$$\tan(y)=\frac{x^2}{2}+@3$$

```
solve(ans(1),y) ENTER
```

$$y=\tan^{-1}\left(\frac{x^2+2\cdot @3}{2}\right)+@n1\cdot\pi$$

Hinweis: Zur Eingabe des Zeichens @ drücken Sie:

◆ STO▸

2nd R

```
ans(1)|@3=c-1 and @n1=0 ENTER
```

$$y=\tan^{-1}\left(\frac{x^2+2\cdot(c-1)}{2}\right)$$

deSolve(*DG1Ordnung* **and** *Ausgangsbedingung*, *unabhängigeVar*, *abhängigeVar*)
⇒ *eine spezielle Lösung*

Ergibt eine spezielle Lösung, welche *DG1Ordnung* und *Ausgangsbedingung* erfüllt. Dies ist in der Regel einfacher, als eine allgemeine Lösung zu bestimmen, Anfangswerte zu ersetzen, nach der abhängigen Variablen aufzulösen und dann diesen Wert in die allgemeine Lösung einzusetzen.

Ausgangsbedingung ist eine Gleichung der Form:

abhängigeVar (*unabhängiger_Anfangswert*) = *abhängiger_Anfangswert*

Unabhängiger_Anfangswert und *abhängiger_Anfangswert* können Variablen wie beispielsweise x0 und y0 ohne gespeicherte Werte sein. Die implizite Differentiation kann bei der Prüfung impliziter Lösungen behilflich sein.

```
sin(y)=(y*e^(x)+cos(y))y'→ode ENTER
```
$$\sin(y)=(e^{x}\cdot y+\cos(y))\cdot y'$$
```
deSolve(ode and y(0)=0,x,y)→soln ENTER
```
$$\frac{-(2\cdot\sin(y)+y^{2})}{2}=-(e^{x}-1)\cdot e^{-x}\cdot\sin(y)$$
```
soln|x=0 and y=0 ENTER        true
```
```
d(right(eq)-left(eq),x)/(d(left(eq)-right(eq),y))→impdif(eq,x,y) ENTER
                              Done
```
```
ode|y'=impdif(soln,x,y) ENTER
                              true
```
```
delVar ode,soln ENTER         Done
```

---

deSolve(*DG2Ordnung* **and** *Ausgangsbedingung1* **and** *Ausgangsbedingung2*, *unabhängigeVar*, *abhängigeVar*) ⇒ *eine spezielle Lösung*

Ergibt eine spezielle Lösung, welche *DG2Ordnung* erfüllt und in einem Punkt einen angegebenen Wert der abhängigen Variablen und deren erster Ableitung aufweist.

Verwenden Sie für *Ausgangsbedingung1* die Form:

abhängige_Var (*unabhängiger_Anfangswert*) = *abhängiger_Anfangswert*

Verwenden Sie für *Ausgangsbedingung2* die Form:

abhängigeVar′ (*unabhängiger_Anfangswert*) = *Anfangswert_1Ableitung*

```
deSolve(y''=y^(-1/2) and y(0)=0 and y'(0)=0,t,y) ENTER
```
$$\frac{2\cdot y^{3/4}}{3}=t$$
```
solve(ans(1),y) ENTER
```
$$y=\frac{2^{2/3}\cdot(3\cdot t)^{4/3}}{4}\ \text{and}\ t\geq 0$$

---

deSolve(*DG2Ordnung* **and** *Randbedingung1* **and** *Randbedingung2*, *unabhängigeVar*, *abhängigeVar*) ⇒ *eine spezielle Lösung*

Ergibt eine spezielle Lösung, welche *DG2Ordnung* erfüllt und in zwei verschiedenen Punkten angegebene Werte aufweist.

```
deSolve(w''-2w'/x+(9+2/x^2)w=x*e^(x) and w(π/6)=0 and w(π/3)=0,x,w) ENTER
```
$$w=\frac{e^{\frac{\pi}{3}}\cdot x\cdot\cos(3\cdot x)}{10}$$
$$-\frac{e^{\frac{\pi}{6}}\cdot x\cdot\sin(3\cdot x)}{10}+\frac{x\cdot e^{x}}{10}$$

det() MATH/Matrix-Menü

det(*quadrat_Matrix***[, *Tol*]) ⇒ *Term***

Gibt die Determinante von *quadrat_Matrix* zurück.

Sie haben die Option, daß jedes Matrixelement als Null behandelt wird, wenn dessen absoluter Wert geringer als *Tol* ist. Diese Toleranz wird nur dann verwendet, wenn die Matrix Fließkommaelemente aufweist und keinerlei symbolische Variablen ohne zugewiesene Werte enthält. Anderenfalls wird *Tol* ignoriert.

- Wenn Sie ◆ ENTER verwenden oder den Modus auf Exact/Approx=APPROXIMATE einstellen, werden Berechnungen in Fließkomma-Arithmetik durchgeführt.
- Wird *Tol* weggelassen oder nicht verwendet, so wird die Standardtoleranz folgendermaßen berechnet:

  5E⁻14 * **max(dim(***quadrat_Matrix***)) *** **rowNorm(***quadrat_Matrix***)**

```
det([a,b;c,d]) ENTER                    a·d-b·c
det([1,2;3,4]) ENTER                         -2
det(identity(3)-x*[1,-2,3;
-2,4,1;-6,-2,7]) ENTER
        -(98·x^3-55·x^2+12·x-1)

[1E20,1;0,1] →mat1                  [1.E20  1]
                                    [0      1]
det(mat1) ENTER                              0
det(mat1,.1) ENTER                     1.E20
```

diag() MATH/Matrix-Menü

diag(*Liste***) ⇒ *Matrix***
diag(*Zeilenmatrix***) ⇒ *Matrix***
diag(*Spaltenmatrix***) ⇒ *Matrix***

Gibt eine Matrix mit den Werten der Parameterliste oder der Matrix in der Hauptdiagonalen zurück.

```
diag({2,4,6}) ENTER                 [2 0 0]
                                    [0 4 0]
                                    [0 0 6]
```

---

diag(*quadrat_Matrix***) ⇒ *Zeilenmatrix***

Gibt eine Zeilenmatrix zurück, die die Elemente der Hauptdiagonalen von *quadrat_Matrix* enthält.

Quadrat_Matrix muß eine quadratische Matrix sein.

```
[4,6,8;1,2,3;5,7,9] ENTER           [4 6 8]
                                    [1 2 3]
                                    [5 7 9]
diag(ans(1)) ENTER                  [4 2 9]
```

Dialog CATALOG

Dialog
Block
EndDlog

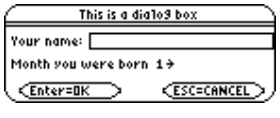
Erzeugt bei Ausführung eines Programms ein Dialogfeld.

Block kann eine einzelne Anweisung oder eine Serie von Anweisungen sein, die durch ":" getrennt sind. Gültige *Block*-Optionen im Menü F3 I/O, 1:Dialog des Programmeditors sind 1:Text, 2:Request, 4:DropDown und 7:Title.

Den Variablen im Dialogfeld können Werte zugewiesen werden, die als Vorgabewerte (Ausgangswerte) angezeigt werden. Wird ENTER gedrückt, werden die Variablen vom Dialogfeld aus aktualisiert und wird die Variable ok auf 1 gesetzt. Wird ESC gedrückt, werden die Variablen nicht aktualisiert, und die Systemvariable ok wird auf Null gesetzt.

Programmlisting:

```
:Dlogtest()
:Prgm
:Dialog
:Title    "This is a dialog box"
:Request  "Your name",Str1
:Dropdown "Month you were born",
  seq(string(i),i,1,12),Var1
:EndDlog
:EndPrgm
```



dim() MATH/Matrix/Dimensions-Menü

dim(*Liste***)** ⇒ *Ganze_Zahl*

Gibt die Dimension von *Liste* zurück.

```
dim({0,1,2}) ENTER                3
```

dim(*Matrix***)** ⇒ *Liste*

Gibt die Dimensionen von *Matrix* als Liste mit zwei Elementen zurück {Zeilen, Spalten}.

```
dim([1,-1,2;-2,3,5]) ENTER  {2 3}
```

dim(*String***)** ⇒ *Ganze_Zahl*

Gibt die Anzahl der in der Zeichenkette *String* enthaltenen Zeichen zurück.

```
dim("Hello") ENTER                5
dim("Hello"&" there") ENTER      11
```

Disp CATALOG

Disp [*TermOderString1*] [**,** *TermOderString2*] **...**

Zeigt den aktuellen Inhalt des Programm- I/O-Bildschirms an. Sind eine oder mehrere *TermOderString* angegeben, wird jeder Term bzw. jede Zeichenkette in einer eigenen Zeile auf dem Programm- I/O-Bildschirm angezeigt.

Ein Term kann Umwandlungsoperationen wie ▸**DD** und **Rect** enthalten. Sie können auch den Operator ▸ benutzen um Einheiten- und Zahlensystem-Umwandlungen durchzuführen.

Ist Pretty Print = ON, werden Terme als "pretty print" angezeigt.

Im Programm- I/O-Bildschirm können Sie F5 drücken, um den Hauptbildschirm anzuzeigen; ein Programm kann **DispHome** verwenden.

```
Disp "Hello" ENTER                Hello
Disp cos(2.3) ENTER              -.666…
{1,2,3,4}→L1 ENTER
Disp L1 ENTER              {1  2  3  4}
Disp 180_min▸_hr ENTER          3.•_hr
```

Hinweis: Zur Eingabe eines Unterstrichzeichens (_) drücken Sie:
[◆] [_]
[2nd] [_]
Das Zeichen ▸ wird mit [2nd] [▸] erzeugt.

DispG CATALOG

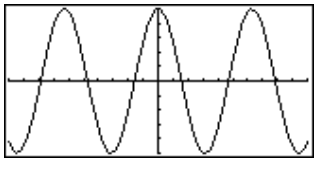
DispG

Zeigt den aktuellen Inhalt des Graphikbildschirms an.

Im Funktions-Graphikmodus:

Programmsegment:

```
  ⋮
:5*cos(x)→y1(x)
:-10→xmin
:10→xmax
:-5→ymin
:5→ymax
:DispG
  ⋮
```



DispHome CATALOG

DispHome

Zeigt den aktuellen Inhalt des Hauptbildschirms an.

Programmausschnitt:

```
  ⋮
:Disp "The result is: ",xx
:Pause "Press Enter to quit"
:DispHome
:EndPrgm
```

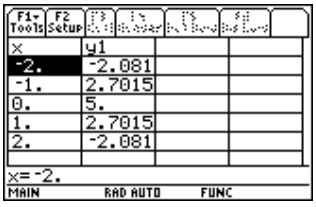
DispTbl CATALOG

DispTbl

Zeigt den aktuellen Inhalt des Tabellen-Bildschirms an.

Hinweis: Das Cursorfeld ist zum Scrollen aktiviert. Drücken Sie ESC oder ENTER, um die Ausführung fortzusetzen, wenn dieser Aufruf innerhalb eines Programms ausgeführt wurde.

```
5*cos(x)→y1(x) ENTER
DispTbl ENTER
```



▸DMS MATH/Angle-Menü

Term **▸DMS**
Liste **▸DMS**
Matrix **▸DMS**

Interpretiert den Parameter als Winkel und zeigt die entsprechenden GMS-Werte (engl. DMS) an (*GGGGGG°MM′SS.ss″*). Siehe °, ', " zur Erläuterung des DMS-Formats (Ordnung, Minuten, Sekunden).

Hinweis: **▸DMS** wandelt Bogenmaß in Ordnung um, wenn es im Rad-Modus benutzt wird. Folgt auf die Eingabe das Ordnung-Symbol (°), wird keine Umwandlung vorgenommen. Sie können **▸DMS** nur am Ende einer Eingabezeile benutzen.

Im Degree-Modus für Winkel:

```
45.371 ▸DMS ENTER          45° 22'15.6"

{45.371,60} ▸DMS ENTER
                  {45° 22'15.6"  60° }
```

dotP() MATH/Matrix/Vector ops-Menü

dotP(*Liste1***, *Liste2***)** ⇒ *Term*

Gibt das Skalarprodukt zweier Listen zurück.

```
dotP({a,b,c},{d,e,f}) ENTER
                        a·d+b·e+c·f

dotP({1,2},{5,6}) ENTER          17
```

---

dotP(*Vektor1***, *Vektor2***)** ⇒ *Term*

Gibt das Skalarprodukt zweier Vektoren zurück.

Es müssen beide Zeilenvektoren oder beide Spaltenvektoren sein.

```
dotP([a,b,c],[d,e,f]) ENTER
                        a·d+b·e+c·f

dotP([1,2,3],[4,5,6]) ENTER      32
```

DrawFunc CATALOG

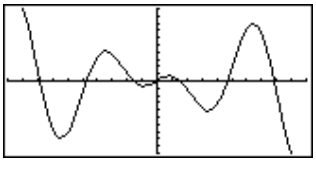
DrawFunc *Term*

Zeichnet *Term* als eine Funktion, wobei x als unabhängige Variable benutzt wird.

Hinweis: Das Neuzeichnen löscht alle gezeichneten Elemente.

Im Funktions-Graphikmodus mit einem ZoomStd-Fenster:

```
DrawFunc 1.25x*cos(x) ENTER
```



DrawInv CATALOG

DrawInv *Term*

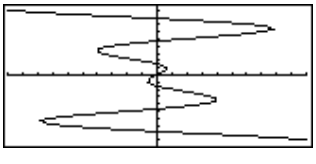
Zeichnet die Inverse von *Term,* indem die x-Werte auf der y-Achse und die y-Werte auf der x-Achse aufgetragen werden.

x ist die unabhängige Variable.

Hinweis: Das Neuzeichnen löscht alle gezeichneten Elemente.

Im Funktions-Graphikmodus mit einem ZoomStd-Fenster:

```
DrawInv 1.25x*cos(x) ENTER
```



DrawParm CATALOG

DrawParm *Term1*, *Term2* [, *tmin*] [, *tmax*] [, *tstep*]

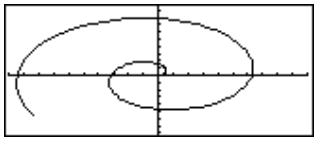
Zeichnet die Parameterdarstellung der Komponenten *Term1* und *Term2*, wobei t die unabhängige Variable ist.

Die Vorgabewerte für *tmin*, *tmax* und *tstep* sind die aktuellen Einstellungen der Window-Variablen tmin, tmax und tstep. Wenn Sie Werte für *tmin*, *tmax* und *tstep* angeben, bewirkt das keine Änderung der Fenstervariablen. Wenn der aktuelle Graphikmodus nicht "parametric" ist, sind diese drei Parameter zwingend erforderlich.

Hinweis: Das Neuzeichnen löscht alle gezeichneten Elemente.

Im Funktions-Graphikmodus mit einem ZoomStd-Fenster:

```
DrawParm
t*cos(t),t*sin(t),0,10,.1 ENTER
```



DrawPol CATALOG

DrawPol *Term*[, θ*min*] [, θ*max*] [, θ*step*]

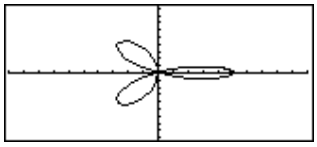
Zeichnet den Polar-Graphen von *Term* unter Verwendung von θ als unabhängiger Variablen.

Die Vorgabewerte für θ*min*, θ*max* und θ*step* sind die aktuellen Einstellungen der Window-Variablen θmin, θmax und θstep. Wenn Sie Werte für θ*min*, θ*max* und θ*step* angeben, bewirkt das keine Änderung der Fenstervariablen. Wenn der aktuelle Graphikmodus nicht "polar" ist, sind diese drei Parameter zwingend erforderlich.

Hinweis: Das Neuzeichnen löscht alle gezeichneten Elemente.

Im Funktions-Graphikmodus mit einem ZoomStd-Fenster:

```
DrawPol 5*cos(3*θ),0,3.5,.1
ENTER
```



DrawSlp CATALOG

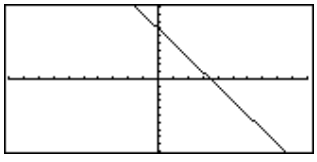
DrawSlp *x1*, *y1*, *Steigung*

Zeigt den Graphikbildschirm an und zeichnet eine Gerade mit der Formel y−y1=*Steigung*·(x−x1).

Hinweis: Das Neuzeichnen löscht alle gezeichneten Elemente.

Im Funktions-Graphikmodus mit einem ZoomStd-Fenster:

```
DrawSlp 2,3,-2 ENTER
```



DropDown CATALOG

DropDown *Titelstring*, **{***Posten1String*, *Posten2String*, **...},** *VarName*

Zeigt ein Dropdown-Menü mit dem Namen *Titelstring* an, das die Einträge **1:***Posten1String*, **2:** *Posten2String* usw. enthält. **DropDown** muß innerhalb eines Blocks **Dialog...EndDlog** benutzt werden.

Wenn *VarName* bereits existiert und einen Wert besitzt, der innerhalb des Bereichs dieser Menüpunkte liegt, wird der diesem Wert zugeordnete Menüpunkt als Vorgabeeinstellung angezeigt. Andernfalls ist der erste Menüpunkt die Voreinstellung (Standardauswahl).

Wenn Sie einen Menüpunkt auswählen, wird die Nummer des gewählten Menüpunkts in der Variablen *VarName* gespeichert. (Falls nötig, erzeugt **DropDown** die Variable *VarName*.)

Siehe **Dialog** Programmlisting-Beispiel.

DrwCtour CATALOG

DrwCtour *Term*
DrwCtour *Liste*

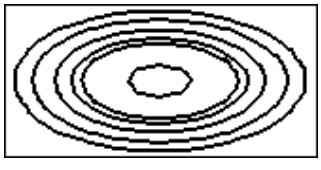
Zeichnet bei den durch *Term* oder *Liste* angegebenen z-Werten Höhenlinien im aktuellen 3D-Graphen. Der Modus 3D Graph muß bereits eingestellt sein. **DrwCtour** setzt das Graph-Format automatisch auf CONTOUR LEVELS.

Standardmäßig enthält der Graph automatisch die Anzahl der durch die Fenstervariable ncontour angegebenen gleichmäßig voneinander entfernten Höhenlinien. **DrwCtour** zeichnet mehr als nur die standardmäßigen Höhenlinien.

Zum Ausschalten der Standardhöhenlinien stellen Sie ncontour auf Null ein; verwenden Sie hierzu entweder den Fensterbildschirm, oder speichern Sie 0 in der Systemvariablen ncontour.

Im Modus 3D Graph:

```
(1/5)x^2+(1/5)y^2-10→z1(x,y)
ENTER                                  Done
-10→xmin:10→xmax ENTER                   10
-10→ymin:10→ymax ENTER                   10
-10→zmin:10→zmax ENTER                   10
0→ncontour ENTER                          0
DrwCtour {-9,-4.5,-3,0,4.5,9}
ENTER
```



- Ändern Sie den Sichtwinkel mit Hilfe des Cursors. Drücken Sie 0 (Null), um zur ursprünglichen Ansicht zurückzukehren.

  Mit folgenden Tasten wird ein neues Graphenformat gewählt:

  [I]
  F

- Drücken Sie X, Y oder Z, um entlang der entsprechenden Achsen zu blicken.

E [EE] Taste [2nd][EE] Taste

*Mantisse***E***exponent*

Eingabe einer Zahl in wissenschaftlicher Notation. Die Zahl wird interpretiert als *Mantisse* × $10^{Exponent}$.

Tipp: Wenn Sie eine Zehnerpotenz ohne Erzeugung eines Dezimalwerts eingeben möchten, benutzen Sie 10^*Ganze_Zahl*.

```
2.3E4 ENTER                              23000.
2.3E9+4.1E15 ENTER                       4.1E15
3*10^4 ENTER                             30000
```

e^() ▦ ◆ [e^x] Taste ⌨ 2nd [e^x] Taste

***e*^(*Term1*)** ⇒ *Term*

Gibt *e* hoch *Term1* zurück.

Hinweis: Das Drücken von ◆ [e^x] zum Anzeigen von e^(ist nicht das gleiche wie die Betätigung von alpha [E]. Auf dem Voyage 200 wird durch Drücken von 2nd [e^x] zur Erzeugung von e^ nicht derselbe Effekt erzielt wie durch Eingabe des Zeichens e auf der QWERTY-Tastatur.

Sie können eine komplexe Zahl in der polaren Form $re^{i\theta}$ eingeben. Verwenden Sie diese aber nur im Winkelmodus Radian, da die Form im Modus Degree einen Domain-Fehler verursacht.

```
e^(1) ENTER                                   e
e^(1.) ENTER                                  2.718...
e^(3)^2 ENTER                                 e^9
```

***e*^(*Liste1*)** ⇒ *Liste*

Gibt *e* hoch jedes Element der *Liste1* zurück.

```
e^({1,1.,0,.5}) ENTER
          {e  2.718...  1  1.648...}
```

***e*^(*quadrat_Matrix1*)** ⇒ *quadrat_Matrix*

Ergibt den Matrix-Exponenten von *quadrat_Matrix1*. Dies ist *nicht* gleichbedeutend mit der Berechnung von *e* hoch jedes Element. Näheres zum Berechnungsverfahren finden Sie im Abschnitt **cos()**.

Quadrat_Matrix1 muß diagonalisierbar sein. Das Ergebnis enthält stets Fließkommazahlen.

```
e^([1,5,3;4,2,1;6,-2,1]) ENTER
[782.209  559.617  456.509]
[680.546  488.795  396.521]
[524.929  371.222  307.879]
```

eigVc() MATH/Matrix-Menü

eigVc(*quadrat_Matrix*) ⇒ *Matrix*

Ergibt eine Matrix, welche die Eigenvektoren für eine reelle oder komplexe *quadrat_Matrix* enthält, wobei jede Spalte des Ergebnisses zu einem Eigenwert gehört. Beachten Sie, daß ein Eigenvektor nicht eindeutig ist; er kann durch einen konstanten Faktor skaliert werden. Die Eigenvektoren sind normiert, d.h. wenn $V = [x_1, x_2, \ldots, x_n]$, dann:

$$\sqrt{x_1{}^2 + x_2{}^2 + \ldots + x_n{}^2} = 1$$

Quadrat_Matrix wird zunächst mit Ähnlichkeitstransformationen bearbeitet, bis die Zeilen- und Spaltennormen so nahe wie möglich bei demselben Wert liegen. Die *Quadrat_Matrix* wird dann auf die obere Hessenberg-Form reduziert, und die Eigenvektoren werden mit einer Schur-Faktorisierung berechnet.

Im Komplex-Formatmodus "Rectangular":

```
[-1,2,5;3,-6,9;2,-5,7]→m1 ENTER
[-1  2  5]
[3  -6  9]
[2  -5  7]

eigVc(m1) ENTER
[-.800...  .767...          .767...          ]
[.484...   .573...+.052...·i  .573...-.052...·i]
[.352...   .262...+.096...·i  .262...-.096...·i]
```

eigVl() MATH/Matrix-Menü

eigVl(*quadrat_Matrix*) ⇒ *Liste*

Ergibt eine Liste von Eigenwerten einer reellen oder komplexen *quadrat_Matrix*.

Quadrat_Matrix wird zunächst mit Ähnlichkeitstransformationen bearbeitet, bis die Zeilen- und Spaltennormen so nahe wie möglich bei demselben Wert liegen. Die *Quadrat_Matrix* wird dann auf die obere Hessenberg-Form reduziert, und die Eigenvektoren werden aus der oberen Hessenberg-Matrix berechnet.

Im Komplex-Formatmodus "Rectangular":

```
[-1,2,5;3,-6,9;2,-5,7]→m1 ENTER
[-1  2  5]
[3  -6  9]
[2  -5  7]

eigVl(m1) ENTER
        {-4.409...  2.204...+.763...·i
             2.204...-.763...·i}...·i}
```

Else

Siehe **If**, Seite 911.

ElseIf **CATALOG** Siehe auch **If**, Seite 911.

If *Boolescher Term1* **Then**
Block1
ElseIf *Boolescher Term2* **Then**
Block2
⋮
ElseIf *Boolescher TermN* **Then**
BlockN
EndIf
⋮

ElseIf kann als Programmanweisung für das Verzweigen in einem Programm benutzt werden.

Programmsegment:

```
  ⋮
:If choice=1 Then
:  Goto option1
:  ElseIf choice=2 Then
:  Goto option2
:  ElseIf choice=3 Then
:  Goto option3
:  ElseIf choice=4 Then
:  Disp "Exiting Program"
:  Return
:EndIf
  ⋮
```

EndCustm Siehe **Custom**, Seite 882.

EndDlog Siehe **Dialog**, Seite 890.

EndFor Siehe **For**, Seite 903.

EndFunc Siehe **Func**, Seite 904.

EndIf Siehe **If**, Seite 911.

EndLoop Siehe **Loop**, Seite 920.

EndPrgm Siehe **Prgm**, Seite 936.

EndTBar Siehe **ToolBar**, Seite 972.

EndTry Siehe **Try**, Seite 973.

EndWhile Siehe **While**, Seite 975.

entry() **CATALOG**

entry() ⇒ *Term*
entry(*Ganze_Zahl***)** ⇒ *Term*

Gibt eine frühere Eingabe aus dem Protokoll-Bereich des Hauptbildschirms zurück.

Ganze_Zahl gibt (sofern angegeben) den Term im Protokoll-Bereich an. Der Vorgabewert ist 1, also die zuletzt ausgewertete Eingabe. Der gültige Bereich ist 1 bis 99, und *Ganze_Zahl* darf kein Term sein.

Hinweis: Wenn die letzte Eingabe noch markiert auf dem Hauptbildschirm angezeigt ist, besitzt das Drücken von ENTER die gleiche Wirkung wie das Ausführen von **entry(1)**.

Auf dem Hauptbildschirm:

`1+1/x` ENTER	$\frac{1}{x}+1$
`1+1/entry(1)` ENTER	$2-\frac{1}{x+1}$
ENTER	$\frac{1}{2\cdot(2\cdot x+1)}+3/2$
ENTER	$5/3-\frac{1}{3\cdot(3\cdot x+2)}$
`entry(4)` ENTER	$\frac{1}{x}+1$

exact() MATH/Number-Menü

exact(*Term1* **[, ***Tol***])** ⇒ *Term*
exact(*Liste1* **[, ***Tol***])** ⇒ *Liste*
exact(*Matrix1* **[, ***Tol***])** ⇒ *Matrix*

Benutzt den Rechenmodus Exact ungeachtet der Moduseinstellung von Exact/Approx und gibt nach Möglichkeit die rationale Zahl zurück, die dem Parameter äquivalent ist.

Tol legt die Toleranz bei der Umwandlung fest, wobei die Vorgabe 0 (Null) ist.

```
exact(.25) ENTER                                  1/4
exact(.333333) ENTER                         333333/1000000
exact(.33333,.001)                                1/3
exact(3.5x+y) ENTER                          7·x/2 + y
exact({.2,.33,4.125}) ENTER
                                  {1/5  33/100  33/8}
```

Exec CATALOG

Exec *String* **[,** *Term1***] [,** *Term2***] ...**

Führt einen aus einer Folge von Motorola 68000 op-Codes bestehenden *String* aus. Diese Codes fungieren als eine Art Assemblersprachen-Programm. Falls erforderlich, können Sie mit den optionalen *Termen* ein oder mehrere Argumente an das Programm übergeben.

Näheres finden Sie auf der TI-Website: **education.ti.com**

Achtung: Mit **Exec** erfolgt Zugriff auf den Mikroprozessor. Ein Fehler führt zur Sperrung des Taschenrechners und zum Datenverlust. Es empfiehlt sich daher, vor der Arbeit mit dem Befehl **Exec** eine Sicherungskopie der Taschenrechnerdaten anzulegen.

Exit CATALOG

Exit

Beendet den aktuellen **For**, **While** oder **Loop** Block.

Exit ist außerhalb dieser drei Schleifenstrukturen (**For**, **While** oder **Loop**) nicht zulässig.

Programmlisting:

```
:0→temp
:For i,1,100,1
:  temp+i→temp
:  If temp>20
:  Exit
:EndFor
:Disp temp
```

Inhalt von **temp** nach Ausführung: 21

exp▸list() CATALOG

exp▸list(*Term***,***Var***)** ⇒ *Liste*

Untersucht *Term* auf Gleichungen, die durch das Wort "or" getrennt sind und gibt eine Liste der rechten Seiten der Gleichungen in der Form *Var=Term* zurück. Dies erlaubt Ihnen auf einfache Weise das Extrahieren mancher Lösungswerte, die in den Ergebnissen der Funktionen **solve()**, **cSolve()**, **fMin()** und **fMax()** enthalten sind.

Hinweis: exp▸list() ist für die Funktionen **zeros** und **cZeros()** unnötig, da diese direkt eine Liste von Lösungswerten zurückgeben.

```
solve(x^2-x-2=0,x) ENTER  x=2 or x=-1

exp▸list(solve(x^2-x-2=0,x),x)
ENTER
                                    {-1  2}
```

expand() MATH/Algebra-Menü

expand(*Term1* **[, *Var*])** ⇒ *Term*
expand(*Liste1* **[,*Var*])** ⇒ *Liste*
expand(*Matrix1* **[,*Var*])** ⇒ *Matrix*

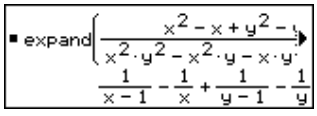
expand(*Term1*) gibt *Term1* bezüglich sämtlicher Variablen entwickelt zurück. Die Entwicklung ist eine Polynomentwicklung für Polynome und eine Partialbruchentwicklung für rationale Terme.

expand() versucht *Term1* in eine Summe und/oder eine Differenz einfacherer Terme umzuformen. Dagegen versucht **factor()** *Term1* in ein Produkt und/oder einen Quotienten einfacher Faktoren umzuformen.

```
expand((x+y+1)^2) ENTER
```

$$x^2+2\cdot x\cdot y+2\cdot x+y^2+2\cdot y+1$$

```
expand((x^2-x+y^2-y)/(x^2*y^2
  -x^2*y-x*y^2+x*y)) ENTER
```



---

expand(*Term1,Var*) entwickelt *Term* bezüglich *Var*. Gleichartige Potenzen von *Var* werden zusammengefaßt. Die Terme und Faktoren werden mit *Var* als der Hauptvariablen sortiert. Es kann sein, daß als Nebeneffekt in gewissem Umfang eine Faktorisierung oder Entwicklung der zusammengefaßten Koeffizienten auftritt. Verglichen mit dem Weglassen von *Var* spart dies häufig Zeit, Speicherplatz und Platz auf dem Bildschirm und macht den Term verständlicher.

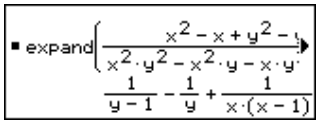
```
expand((x+y+1)^2,y) ENTER
```

$$y^2+2\cdot y\cdot(x+1)+(x+1)^2$$

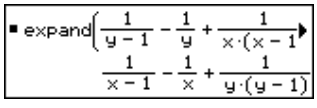
```
expand((x+y+1)^2,x) ENTER
```

$$x^2+2\cdot x\cdot(y+1)+(y+1)^2$$

```
expand((x^2-x+y^2-y)/(x^2*y^2
  -x^2*y-x*y^2+x*y),y) ENTER
```



```
expand(ans(1),x) ENTER
```



Selbst wenn es nur eine Variable gibt, kann das Einbeziehen von *Var* eine vollständigere Faktorisierung des Nenners, die für die Partialbruchentwicklung benutzt wird, ermöglichen.

Tipp: Für rationale Terme ist **propFrac()** eine schnellere aber weniger weitgehende Alternative zu **expand()**.

Hinweis: Siehe auch **comDenom()** zu einem Quotienten aus einem entwickelten Zähler und entwickeltem Nenner.

```
expand((x^3+x^2-2)/(x^2-2))
ENTER
```

$$\frac{2\cdot x}{x^2-2}+x+1$$

```
expand(ans(1),x) ENTER
```

$$\frac{1}{x-\sqrt{2}}+\frac{1}{x+\sqrt{2}}+x+1$$

expand(*Term1,*[*Var*]) vereinfacht auch Logarithmen und Bruchpotenzen ungeachtet von *Var*. Für weitere Zerlegungen von Logarithmen und Bruchpotenzen können Einschränkungen notwendig werden, um sicherzustellen, daß manche Faktoren nicht negativ sind.

expand(*Term1,* [*Var*]) vereinfacht auch Absolutwerte, **sign()** und Exponenten ungeachtet von *Var*.

Hinweis: Siehe auch **tExpand()** zur trigonometrischen Entwicklung von Winkelsummen und -produkten.

```
ln(2x*y)+√(2x*y) ENTER
```

$$\ln(2\cdot x\cdot y)+\sqrt{(2\cdot x\cdot y)}$$

```
expand(ans(1)) ENTER
```

$$\ln(x\cdot y)+\sqrt{2}\cdot\sqrt{(x\cdot y)}+\ln(2)$$

```
expand(ans(1))|y>=0 ENTER
```

$$\ln(x)+\sqrt{2}\cdot\sqrt{x}\cdot\sqrt{y}+\ln(y)+\ln(2)$$

```
sign(x*y)+abs(x*y)+e^(2x+y)
ENTER
```

$$e^{2\cdot x+y}+\text{sign}(x\cdot y)+|x\cdot y|$$

```
expand(ans(1)) ENTER
```

$$\text{sign}(x)\cdot\text{sign}(y)+|x|\cdot|y|+(e^x)^2\cdot e^y$$

expr() MATH/String-Menü

expr(*String*) ⇒ *Term*

Gibt die in *String* enthaltene Zeichenkette als Term zurück und führt diesen sofort aus.

```
expr("1+2+x^2+x") ENTER                x²+x+3
expr("expand((1+x)^2)") ENTER
                                     x²+2·x+1
"Define cube(x)=x^3"→funcstr
ENTER
                    "Define cube(x)=x^3"
expr(funcstr) ENTER                    Done
cube(2) ENTER                             8
```

ExpReg MATH/Statistics/Regressions-Menü

ExpReg *Liste1, Liste2* [, [*Liste3*] [, *Liste4, Liste5*]]

Berechnet die exponentielle Regression und aktualisiert alle Statistik-Systemvariablen.

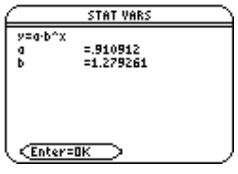
Alle Listen außer *Liste5* müssen die gleiche Dimension besitzen.

Liste1 stellt die Liste der x-Werte dar.
Liste2 stellt die Liste der y-Werte dar.
Liste3 stellt die Angaben für die Häufigkeit dar.
Liste4 stellt die Klassencodes dar.
Liste5 stellt die Klassenliste dar.

Hinweis: *Liste1* bis einschl. *Liste4* müssen Variablenamen oder c1–c99 sein (Spalten in der letzten Datenvariablen, die im Daten/ Matrix-Editor angezeigt wurde). *Liste5* braucht kein Variablenname zu sein und kann keine Spalte c1–c99 sein.

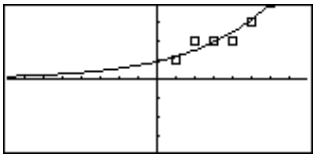
Im Funktions-Graphikmodus:

```
{1,2,3,4,5,6,7,8}→L1 ENTER
                                     {1 2 …}
{1,2,2,2,3,4,5,7}→L2 ENTER
                                     {1 2 …}
ExpReg L1,L2 ENTER                     Done
ShowStat ENTER
```




```
ENTER
Regeq(x)→y1(x) ENTER                   Done
NewPlot 1,1,L1,L2 ENTER                Done
```

▼ [GRAPH]



factor() MATH/Algebra-Menü

factor(*Term1***[,** *Var***])** ⇒ *Term*
factor(*Liste1***[,***Var***])** ⇒ *Liste*
factor(*Matrix1***[,***Var***])** ⇒ *Matrix*

factor(*Term1***)** gibt *Term1* nach allen seinen Variablen bezüglich eines gemeinsamen Nenners faktorisiert zurück.

Term1 wird soweit wie möglich in lineare rationale Faktoren aufgelöst, selbst wenn dies die Einführung neuer nicht-reeller Unterterme bedeutet. Diese Alternative ist angemessen, wenn Sie die Faktorisierung bezüglich mehr als einer Variablen vornehmen möchten.

```
factor(a^3*x^2-a*x^2-a^3+a)
ENTER
                  a·(a-1)·(a+1)·(x-1)·(x+1)
factor(x^2+1) ENTER                    x²+1
factor(x^2-4) ENTER                (x-2)·(x+2)
factor(x^2-3) ENTER                    x²-3
factor(x^2-a) ENTER                    x²-a
```

factor(*Term1,Var***)** gibt *Term1* nach der Variablen *Var* faktorisiert zurück.

Term1 wird soweit wie möglich in reelle Faktoren aufgelöst, die linear in *Var* sind, selbst wenn dadurch irrationale Konstanten oder Unterterme, die in anderen Variablen irrational sind, eingeführt werden.

Die Faktoren und ihre Terme werden mit *Var* als Hauptvariable sortiert. Gleichartige Potenzen von *Var* werden in jedem Faktor zusammengefaßt. Beziehen Sie *Var* ein, wenn die Faktorisierung nur bezüglich dieser Variablen benötigt wird und Sie irrationale Terme in anderen Variablen akzeptieren möchten, um die Faktorisierung bezüglich *Var* so weit wie möglich vorzunehmen. Es kann sein, daß als Nebeneffekt in gewissem Umfang eine Faktorisierung nach anderen Variablen auftritt.

Bei der Einstellung AUTO für den Modus Exact/Approx ermöglicht die Einbeziehung von *Var* auch eine Näherung mit Gleitkommakoeffizienten in Fällen, wo irrationale Koeffizienten nicht explizit bezüglich der integrierten Funktionen ausgedrückt werden können. Selbst wenn es nur eine Variable gibt, kann das Einbeziehen von *Var* eine vollständigere Faktorisierung ergeben.

Hinweis: Siehe auch **comDenom()** zu einer schnellen partiellen Faktorisierung, wenn **factor()** zu langsam ist oder den Speicherplatz erschöpft.

Hinweis: Siehe auch **cFactor()** zur kompletten Faktorisierung bis zu komplexen Koeffizienten, um lineare Faktoren zu erhalten.

```
factor(a^3*x^2-a*x^2-a^3+a,x) ENTER
```

$$a \cdot (a^2 - 1) \cdot (x - 1) \cdot (x + 1)$$

```
factor(x^2-3,x) ENTER
```

$$(x + \sqrt{3}) \cdot (x - \sqrt{3})$$

```
factor(x^2-a,x) ENTER
```

$$(x + \sqrt{a}) \cdot (x - \sqrt{a})$$

```
factor(x^5+4x^4+5x^3-6x-3) ENTER
```

$$x^5 + 4 \cdot x^4 + 5 \cdot x^3 - 6 \cdot x - 3$$

```
factor(ans(1),x) ENTER
```

$$(x - .964\ldots) \cdot (x + .611\ldots) \cdot (x + 2.125\ldots) \cdot (x^2 + 2.227\ldots \cdot x + 2.392\ldots)$$

---

factor(*Rationale_Zahl***)** ergibt die rationale Zahl in Primfaktoren zerlegt. Bei zusammengesetzten Zahlen nimmt die Berechnungsdauer exponentiell mit der Anzahl an Stellen im zweitgrößten Faktor zu. Das Faktorisieren einer 30-stelligen ganzen Zahl kann beispielsweise länger als einen Tag dauern und das Faktorisieren einer 100-stelligen Zahl mehr als ein Jahrhundert.

Hinweis: Um eine Berechnung anzuhalten (abzubrechen), drücken Sie auf ON.

Möchten Sie hingegen lediglich feststellen, ob es sich bei einer Zahl um eine Primzahl handelt, verwenden Sie **isPrime().** Dieser Vorgang ist wesentlich schneller, insbesondere dann, wenn *Rationale_Zahl* keine Primzahl ist und der zweitgrößte Faktor mehr als fünf Stellen aufweist.

```
factor(152417172689) ENTER
                    123457·1234577
isPrime(152417172689) ENTER false
```

Fill — MATH/Matrix-Menü

Fill *Term, MatrixVar* ⇒ *Matrix*

Ersetzt jedes Element in der Variablen *MatrixVar* durch *Term*.

MatrixVar muß bereits existieren.

```
[1,2;3,4]→amatrx ENTER        [1 2]
                              [3 4]
Fill 1.01,amatrx ENTER          Done
amatrx ENTER                  [1.01 1.01]
                              [1.01 1.01]
```

Fill *Term, ListeVar* ⇒ *Liste*

Ersetzt jedes Element in der Variablen *ListeVar* durch *Term*.

ListeVar muß bereits existieren.

```
{1,2,3,4,5}→alist ENTER
                        {1 2 3 4 5}
Fill 1.01,alist ENTER           Done
alist ENTER
     {1.01 1.01 1.01 1.01 1.01}
```

floor() — MATH/Number-Menü

floor(*Term***)** ⇒ *Ganze_Zahl*

Gibt die größte ganze Zahl zurück, die ≤ Parameter ist. Diese Funktion ist identisch mit **int()**.

Der Parameter kann eine reelle oder eine komplexe Zahl sein.

```
floor(⁻2.14) ENTER              ⁻3.
```

---

floor(*Liste1***)** ⇒ *Liste*
floor(*Matrix1***)** ⇒ *Matrix*

Gibt eine Liste oder Matrix zurück, die für jedes Element die größte ganze Zahl enthält, die kleiner oder gleich dem Element ist.

Hinweis: Siehe auch **ceiling()** und **int()**.

```
floor({3/2,0,⁻5.3}) ENTER
                          {1 0 ⁻6.}
floor([1.2,3.4;2.5,4.8]) ENTER
                          [1. 3.]
                          [2. 4.]
```

fMax() — MATH/Calculus-Menü

fMax(*Term, Var***)** ⇒ *Boolescher Term*

Gibt einen Booleschen Term zurück, der mögliche Werte von *Var* angibt, welche *Term* maximieren oder seine kleinste obere Grenze angeben.

Benutzen Sie den Operator "|" mit zur Einschränkung des Lösungsintervalls und/oder zur Angabe des Vorzeichens anderer undefinierter Variablen.

Ist der Modus Exact/Approx auf APPROX eingestellt, sucht **fMax()** iterativ nach einem annähernden lokalen Maximum. Dies ist oft schneller, insbesondere, wenn Sie den Operator "|" benutzen, um die Suche auf ein relativ kleines Intervall zu beschränken, das genau ein lokales Maximum enthält.

Hinweis: Siehe auch **fMin()** und **max()**.

```
fMax(1-(x-a)^2-(x-b)^2,x) ENTER
                          x = (a+b)/2
fMax(.5x^3-x-2,x) ENTER       x = ∞
fMax(.5x^3-x-2,x)|x≤1 ENTER
                          x = ⁻.816...
fMax(a*x^2,x) ENTER
  x=∞ or x=⁻∞ or x=0 or a=0
fMax(a*x^2,x)|a<0 ENTER       x=0
```

fMin() MATH/Calculus-Menü

fMin(*Term, Var***)** ⇒ *Boolescher Term*

Gibt einen Booleschen Term zurück, der mögliche Werte von *Var* angibt, welche *Term* minimieren oder seine kleinste untere Grenze angeben.

Benutzen Sie den Operator "|" mit zur Einschränkung des Lösungsintervalls und/oder zur Angabe des Vorzeichens anderer undefinierter Variablen.

Ist der Modus Exact/Approx auf APPROX eingestellt, sucht **fMin()** iterativ nach einem annähernden lokalen Minimum. Dies ist oft schneller, insbesondere, wenn Sie den Operator "|" benutzen, um die Suche auf ein relativ kleines Intervall zu beschränken, das genau ein lokales Minimum enthält.

Hinweis: Siehe auch **fMax()** und **min()**.

```
fMin(1-(x-a)^2-(x-b)^2,x) ENTER
                    x=∞ or x=⁻∞
fMin(.5x^3-x-2,x)|x≥1 ENTER  x=1
fMin(a*x^2,x) ENTER
  x=∞ or x=⁻∞ or x=0 or a=0
fMin(a*x^2,x)|a>0 and x>1 ENTER
                              x=1.
fMin(a*x^2,x)|a>0 ENTER       x=0
```

FnOff CATALOG

FnOff

Hebt die Auswahl aller Y= Funktionen für den aktuellen Graphikmodus auf.

Bei geteiltem Bildschirm mit 2-Graphen-modus wirkt sich **FnOff** nur auf den aktiven Graph aus.

FnOff [1] [, 2] ... [,99]

Hebt die Auswahl der angegebenen Y= Funktionen für den aktuellen Graphikmodus auf.

Im Funktions-Graphikmodus:
FnOff 1,3 ENTER hebt die Auswahl von y1(x) und y3(x) auf.

Im Graphikmodus "parametric":
FnOff 1,3 ENTER hebt die Auswahl von xt1(t), yt1(t), xt3(t) und yt3(t) auf.

FnOn CATALOG

FnOn

Wählt alle für den aktuellen Graphikmodus definierten Y= Funktionen aus.

Bei geteiltem Bildschirm mit 2-Graphen-modus wirkt sich **FnOn** nur auf den aktiven Graph aus.

FnOn [1] [, 2] ... [,99]

Wählt die angegebenen Y= Funktionen für den aktuellen Graphikmodus aus.

Hinweis: Im 3D-Graphikmodus kann jeweils nur eine Funktion ausgewählt sein. FnOn 2 wählt z. B. z2(x,y) aus und hebt die Auswahl aller vorher ausgewählten Funktionen auf. In den anderen Graph-Modi wirkt sich diese Funktion nicht auf bereits ausgewählte Funktionen aus.

For CATALOG

For *Var*, *Anfang*, *Ende* [, *Schritt*]
Block
EndFor

Führt die in *Block* befindlichen Anweisungen für jeden Wert von *Var* zwischen *Anfang* und *Ende* aus, wobei der Wert bei jedem Durchlauf um *Schritt* inkrementiert wird.

Var darf keine Systemvariable sein.

Schritt kann positiv oder negativ sein. Der Vorgabewert ist 1.

Block kann eine einzelne Anweisung oder eine Serie von Anweisungen sein, die durch ":" getrennt sind.

Programmsegment:

```
  ⋮
:0→tempsum : 1→step
:For i,1,100,step
:  tempsum+i→tempsum
:EndFor
:Disp tempsum
  ⋮
```

Inhalt von `tempsum` nach Ausführung: `5050`

Inhalt von `tempsum` bei Änderung von `step` in 2: `2500`

format() MATH/String-Menü

format(*Term*[, *FormatString*]**)** ⇒ *string*

Gibt *Term* als Zeichenkette im Format der Formatschablone zurück.

Term muß zu einer Zahl vereinfachbar sein. *FormatString* ist eine Zeichenkette und muß diese Form besitzen: "F[*n*]", "S[*n*]", "E[*n*]", "G[*n*][*c*]", wobei [ ] optionale Teile bedeutet.

F[*n*]: Festes Format (Fixed). *n* ist die Anzahl der angezeigten Nachkommastellen (nach dem Dezimalpunkt).

S[*n*]: Wissenschaftliches Format. *n* ist die Anzahl der angezeigten Nachkommastellen (nach dem Dezimalpunkt).

E[*n*]: Technisches Format (Engineering). *n* ist die Anzahl der Stellen, die auf die erste signifikante Ziffer folgen. Der Exponent wird auf ein Vielfaches von 3 gesetzt, und der Dezimalpunkt wird um Null, eine oder zwei Stellen nach rechts verschoben.

G[*n*][*c*]: Wie Fixed, unterteilt jedoch auch die Stellen links des Dezimaltrennzeichens in Dreiergruppen. *c* ist das Gruppentrennzeichen und ist auf "Komma" voreingestellt. Wenn *c* auf "Punkt" gesetzt wird, wird das Dezimaltrennzeichen zum Komma.

[R*c*]: Jeder der vorstehenden Formateinstellungen kann als Suffix das Flag R*c* nachgestellt werden, wobei *c* ein einzelnes Zeichen ist, das den Dezimalpunkt ersetzt.

```
format(1.234567,"f3") ENTER
                          "1.235"
format(1.234567,"s2") ENTER
                        "1.23E0"
format(1.234567,"e3") ENTER
                       "1.235E0"
format(1.234567,"g3") ENTER
                          "1.235"
format(1234.567,"g3") ENTER
                      "1,234.567"
format(1.234567,"g3,r:") ENTER
                          "1:235"
```

fPart() MATH/Number-Menü

fPart(*Term1***)** ⇒ *Term*
fPart(*Liste1***)** ⇒ *Liste*
fPart(*Matrix1***)** ⇒ *Matrix*

Gibt den Bruchanteil/Nachkommateil des Parameters zurück.

Bei einer Liste bzw. Matrix werden die Bruchteile aller Elemente zurückgegeben.

Der Parameter kann eine reelle oder eine komplexe Zahl sein.

```
fPart(-1.234) ENTER          -.234
fPart({1, -2.3, 7.003}) ENTER
                  {0  -.3  .003}
```

Func CATALOG

Func
Block
EndFunc

Zwingend erforderlich als erste Anweisung in einer aus mehreren Anweisungen bestehenden Funktionsdefinition.

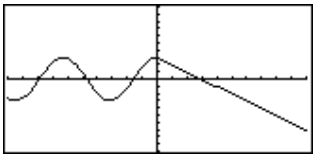
Block kann eine einzelne Anweisung oder eine Serie von Anweisungen sein, die durch ":" getrennt sind.

Hinweis: when() kann ebenfalls zum Definieren und zur graphischen Darstellung von stückweise definierten Funktionen benutzt werden.

Definieren Sie im Funktions-Graphikmodus eine stückweise definierte Funktion:

```
Define g(x)=Func:If x<0 Then
  :Return 3*cos(x):Else:Return
  3-x:EndIf:EndFunc ENTER    Done
```

```
Graph g(x) ENTER
```



gcd() MATH/Number-Menü

gcd(*Zahl1, Zahl2***)** ⇒ *Term*

Gibt den größten gemeinsamen Teiler (ggT) der beiden Parameter zurück. Der **gcd** zweier Brüche ist der **gcd** ihrer Zähler dividiert durch das kleinste gemeinsame Vielfache (**lcm**) ihrer Nenner.

In den Modi Auto oder Approximate ist der **gcd** von Fließkommabrüchen 1.0.

```
gcd(18,33) ENTER                    3
```

---

gcd(*Liste1, Liste2***)** ⇒ *Liste*

Gibt die ggT der einander entsprechenden Elemente von *Liste1* und *Liste2* zurück.

```
gcd({12,14,16},{9,7,5}) ENTER
                          {3 7 1}
```

---

gcd(*Matrix1, Matrix2***)** ⇒ *Matrix*

Gibt die ggT der einander entsprechenden Elemente von *Matrix1* und *Matrix2* zurück.

```
gcd([2,4;6,8],[4,8;12,16]) ENTER
                          [2  4]
                          [6  8]
```

Get CATALOG

Get *Var*

Übernimmt einen CBL™-(Calculator-Based Laboratory™) oder einen CBR™-(Calculator-Based Ranger™)Wert über den Kommunikationsanschluß und speichert ihn in der Variablen *Var*.

Programmsegment:

```
  :
:Send {3,1,-1,0}
:For i,1,99
:  Get data[i]
:  PtOn i,data[i]
:EndFor
  :
```

GetCalc CATALOG

GetCalc *Var*

Übernimmt einen Wert vom Kommunikationsanschluß und speichert ihn in die Variable *Var*. Dies gilt beim Verbinden von 2 Rechnern.

Hinweis: Um eine Variable einem Gerät an den Kommunikationsanschluß zu übertragen, können Sie 2nd [VAR-LINK] auf dem anderen Gerät benutzen, dort die Variable auswählen und senden, oder Sie können im anderen Gerät den Befehl **SendCalc** benutzen.

Programmsegment:

```
  :
: Disp "Press Enter when
ready"
:Pause
:GetCalc L1
:Disp "List L1 received "
  :
```

GetCalc *Var[,port]*

Ruft einen Wert von der Verbindungsschnittstelle ab und speichert ihn auf dem empfangenden TI-89 Titanium in der Variable *var*.

Ist der Anschluss nicht angegeben oder ist *port = 0*, erwartet der TI-89 Titanium die Daten von beiden Anschlüssen.

Bei *port = 1* erwartet der TI-89 Titanium die Daten vom USB-Anschluss.

Bei *port = 2* erwartet der TI-89 Titanium die Daten vom E/A-Anschluss.

getConfg() CATALOG

getConfg() ⇒ *Listenpaare*

Gibt eine Liste der Taschenrechnerattribute aus. Der Name des Attributs wird, gefolgt von dessen Wert, aufgeführt.

:

```
getConfg() ENTER
       {"Product Name" "Advanced
          Mathematics Software"
    "Version" "2.00, 09/25/1999"
        "Product ID" "03-1-4-68"
       "ID #" "01012 34567 ABCD"
                "Cert. Rev. #" 0
              "Screen Width" 160
            "Screen Height" 100
             "Window Width" 160
             "Window Height" 67
              "RAM Size" 262132
              "Free RAM" 197178
           "Archive Size" 655360
         "Free Archive" 655340}
```

```
getConfg() ENTER
       {"Product Name" "Advanced
          Mathematics Software"
    "Version" "2.00, 09/25/1999"
        "Product ID" "01-1-4-80"
       "ID #" "01012 34567 ABCD"
                "Cert. Rev. #" 0
              "Screen Width" 240
            "Screen Height" 120
             "Window Width" 240
             "Window Height" 91
              "RAM Size" 262144
              "Free RAM" 192988
           "Archive Size" 720896
         "Free Archive" 720874}
```

Hinweis: Auf Ihrem Bildschirm werden möglicherweise andere Attributwerte angezeigt. Das Attribut Cert. Rev. # erscheint nur, wenn Sie Zusatzsoftware erworben und installiert haben.

getDate() CATALOG

getDate() ⇒ *list*

Gibt eine Liste mit dem Datum zurück, das dem aktuellen Stand der Uhr entspricht. Die Liste hat das Format {*Jahr,Monat,Tag*}.

```
getDate() ENTER        {2002  2  22}
```

getDenom() MATH/Algebra/Extract-Menü

getDenom(*Term1***)** ⇒ *Term*

Transformiert *Term1* in einen Term mit gekürztem gemeinsamem Nenner und gibt dann den Nenner zurück.

```
getDenom((x+2)/(y-3)) ENTER   y-3
getDenom(2/7) ENTER            7
getDenom(1/x+(y^2+y)/y^2) ENTER
                               x·y
```

getDtFmt() CATALOG

getDtFmt() ⇒ *integer*

Gibt eine ganze Zahl (Integer) zurück, welche das Datumsformat darstellt, welches momentan für das Gerät festgelegt ist

Integer-Werte:

```
1 = MM/TT/JJ
2 = TT/MM/JJ
3 = MM.TT.JJ
4 = TT.MM.JJ
5 = JJ.MM.TT
6 = MM-TT-JJ
7 = TT-MM-JJ
8 = JJ-MM-TT
```

getDtStr() CATALOG

getDtStr([*integer***])** ⇒ *string*

Gibt eine Zeichenkette mit dem aktuellen Datum im aktuellen Datumsformat zurück. So entspricht beispielsweise die zurückgegebenen Zeichenkette *28/09/02* dem 28. Tag im September 2002 (mit dem Datumsformat DD/MM/YY).

Wenn Sie den optionalen Integer für das Datumsformat angeben, wird die Zeichenkette im angegebenen Format zurückgegeben.

Optionale Integer-Werte:

```
1 = MM/TT/JJ
2 = TT/MM/JJ
3 = MM.TT.JJ
4 = TT.MM.JJ
5 = JJ.MM.TT
6 = MM-TT-JJ
7 = TT-MM-JJ
8 = JJ-MM-TT
```

getFold() CATALOG

getFold() ⇒ *NameString*

Gibt den Namen des aktuellen Verzeichnisses als Zeichenkette zurück.

```
getFold() ENTER                "main"
getFold()→oldfoldr ENTER       "main"
oldfoldr ENTER                 "main"
```

getKey() CATALOG

getKey() ⇒ *Ganze_Zahl*

Gibt den Tastencode der gedrückten Taste zurück. Gibt 0 zurück, wenn keine Taste gedrückt wurde.

Die Modifikatortasten (Umschalttaste ⇧, Zweite Funktion 2nd, Option ◆, Buchstaben alpha und Ziehen ⎋) werden selbst nicht erkannt, aber sie ändern den Code der anschließenden Taste, d. h. ◆ X ≠ X ≠ 2nd X.

Eine Zusammenstellung der Tastencodes finden Sie in Anhang B.

Programmlisting:

```
:Disp
:Loop
:  getKey()→taste
:  while taste=0
:    getKey()→taste
:  EndWhile
:  Disp taste
:  If taste = ord("a")
:  Stop
:EndLoop
```

getMode() CATALOG

getMode(*ModusNameString***)** ⇒ *String*
getMode("ALL") ⇒ *ListederStringPaare*

Wenn der Parameter ein bestimmter Modusname ist, wird eine Zeichenkette zurückgegeben, die die aktuelle Einstellung für diesen Modus enthält.

Wenn der Parameter **"ALL"** lautet, wird eine Liste mit Zeichenkettenpaaren zurückgegeben, die die aktuellen Einstellungen aller Modi enthält. Wenn Sie die Modus-Einstellungen später wiederherstellen möchten, müssen Sie das Ergebnis von **getMode("ALL")** in eine Variable - speichern und dann die Modi mit **setMode** wiederherstellen.

Eine Zusammenstellung der Modusnamen und möglicher Einstellungen finden Sie bei **setMode()**.

Hinweis: Zum Einstellen des Modus Unit System oder um Informationen über diesen abzurufen, verwenden Sie **setUnits()** oder **getUnits()** anstelle von **setMode()** oder **getMode()**.

```
getMode("angle")ENTER   "RADIAN"

getMode("graph")ENTER "FUNCTION"

getMode("all")ENTER
              {"Graph" "FUNCTION"
       "Display Digits" "FLOAT 6"
                 "Angle" "RADIAN"
    "Exponential Format" "NORMAL"
          "Complex Format" "REAL"
    "Vector Format" "RECTANGULAR"
            "Pretty Print" "ON"
          "Split Screen" "FULL"
           "Split 1 App" "Home"
          "Split 2 App" "Graph"
        "Number of Graphs" "1"
           "Graph 2" "FUNCTION"
      "Split Screen Ratio"  "1,1"
          "Exact/Approx" "AUTO"
                "Base" "DEC"}
```

Hinweis: Auf Ihrem Bildschirm werden wahrscheinlich andere Moduseinstellungen angezeigt.

getNum() MATH/Algebra/Extract-Menü

getNum(*Term1***)** ⇒ *Term*

Transformiert *Term1* in einen Term mit gekürztem gemeinsamem Nenner und gibt dann den Zähler zurück.

```
getNum((x+2)/(y-3))ENTER    x+2

getNum(2/7)ENTER              2

getNum(1/x+1/y)ENTER        x+y
```

getTime() CATALOG

getTime() ⇒ *list*

Gibt eine Liste mit der Zeit zurück, die dem aktuellen Stand der Uhr entspricht. Die Liste hat das Format {*Stunde,Minute,Sekunde*}. Die Zeit wird im 24-Stundenformat zurückgegeben.

getTmFmt() CATALOG

getTmFmt() ⇒ *integer*

Gibt eine ganze Zahl (Integer) zurück, die dem momentan für das Gerät festgelegten Zeitformat entspricht.

Integer-Werte:

```
12 = 12-Stunden-Uhr

24 = 24-Stunden-Uhr
```

getTmStr() CATALOG

getTmStr([*integer***])** ⇒ *string*

Gibt eine Zeichenkette mit der aktuellen Zeit im aktuell festgelegten Zeitformat zurück.

Wenn Sie den optionalen Integer für das Zeitformat angeben, wird die Zeichenkette im angegebenen Format zurückgegeben.

Optionale Integer-Werte:

```
12 = 12 -Stunden-Uhr

24 = 24 -Stunden-Uhr
```

getTmZn() CATALOG

getTmZn() ⇒ *integer*

Gibt eine ganze Zahl (Integer) zurück, die der momentan für das Gerät festgelegten Zeitzone entspricht.

Der zurückgegebene Integer entspricht der Anzahl der Minuten, um welche die Zeitzone relativ zur Greenwich Mean Time (GMT) verschoben ist, die in Greenwich, England vorgegeben wird. Wenn die Zeitzone beispielsweise zur GMT um zwei Stunden verschoben ist, gibt das Gerät 120 (Minuten) zurück.

Integer für Zeitzonen westlich der GMT sind negativ.

Integer für Zeitzonen östlich der GMT sind positiv.

Wenn die Greenwich Mean Time 14:07:07 ist, ist es:

in Denver, Colorado (Mountain Daylight Time) 8:07:07 a.m.
(–360 Minuten zur GMT)

in Brüssel, Belgien (Central European Standard Time) 16:07:07 p.m.
(+120 Minuten zur GMT)

getType() CATALOG

getType(*Var***)** ⇒ *String*

Gibt eine Zeichenkette zurück, die den Datentyp der Variablen *Var* angibt.

Wurde *Var* nicht definiert, wird "NONE" zurückgegeben.

```
{1,2,3}→temp ENTER                {1 2 3}
getType(temp) ENTER               "LIST"

2+3i→temp ENTER                   2+3i
getType(temp) ENTER               "EXPR"

DelVar temp ENTER                 Done
getType(temp) ENTER               "NONE"
```

Datentyp	Variableninhalt
"ASM"	Assemblersprachen-Programm
"DATA"	Datentyp
"EXPR"	Term (einschließl. komplex/beliebig/undefiniert ∞, ⁻∞, WAHR, FALSCH, pi, *e*)
"FUNC"	Funktion
"GDB"	Graphik-Einstellung
"LIST"	Liste
"MAT"	Matrix
"NONE"	Variable existiert nicht
"NUM"	Reelle Zahl
"OTHER"	Verschiedene Datentypen für künftige Verwendung durch Software-Anwendungen
"PIC"	Bild/Graphik
"PRGM"	Programm
"STR"	Zeichenkette (String)
"TEXT"	Text
"VAR"	Name einer anderen Variablen

getUnits() CATALOG

getUnits() ⇒ *Liste*

Ergibt eine Liste von Zeichenfolgen, welche die aktuellen Standardeinheiten für alle Kategorien außer Konstanten, Temperatur, Stoffmenge, Lichtstärke und Beschleunigung enthalten. *Liste* hat die Form:

{"*System*" "*Kat1*" "*Einheit1*" "*Kat2*" "*Einheit2*" …}

Die erste Zeichenfolge gibt das System an (SI, ENG/US oder CUSTOM). Die nachfolgenden String-Paare geben eine Kategorie (wie z.B. Länge) und deren Standardeinheit (wie z.B. _m für Meter) an.

Die Standardeinheiten stellen Sie mit **setUnits()** ein.

```
getUnits() ENTER
          {"SI"  "Area"  "NONE"
            "Capacitance"  "_F"
              "Charge"  "_coul"
                          … }
```

Hinweis: Auf Ihrem Bildschirm werden möglicherweise andere Standardeinheiten angezeigt.

Goto CATALOG

Goto *LabelName*

Setzt die Programmausführung bei der Marke *LabelName* fort.

LabelName muß im selben Programm mit der Anweisung **Lbl** definiert worden sein.

Programmsegment:

```
  ⋮
:0→temp
:1→i
:Lbl TOP
:  temp+i→temp
:  If i<10 Then
:   i+1→i
:   Goto TOP
:  EndIf
:Disp temp
  ⋮
```

Graph CATALOG

Graph *Term1*[, *Term2*] [, *Var1*] [, *Var2*]

Smart Graph stellt die gewünschten Terme/ Funktionen im aktuellen Graphikmodus graphisch dar.

Terme, die mit den Befehlen **Graph** oder **Table** eingegeben werden, werden beginnend mit der Zahl 1 durchnumeriert. Sie können sie bei angezeigter Tabelle ändern oder einzeln löschen, indem Sie F4 Header drücken. Die derzeit ausgewählten Y= Funktionen werden dabei ignoriert.

Wenn Sie den optionalen Parameter *Var* weglassen, benutzt **Graph** die unabhängige Variable des aktuellen Graphikmodus.

Hinweis: Nicht alle optionalen Parameter sind in allen Modi gültig, da Sie nie alle vier Parameter gleichzeitig benutzen können.

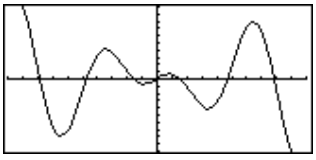
Einige gültige Formen dieser Anweisung sind:

Funktions-Graph	**Graph** *Term*, *x*
Parameterdarstellung	**Graph** *xTerm*, *yTerm*, *t*
Polar-Graph	**Graph** *Term*, θ
Folgen-Graph	Nicht zulässig.
3D-Graph	**Graph** *Term*, *x*, *y*
DGL-Graph	Nicht zulässig.

Hinweis: Benutzen Sie **ClrGraph** zum Löschen dieser Funktionen, oder wechseln Sie zum Y= Editor, um die System-Y= Funktionen erneut zu aktivieren.

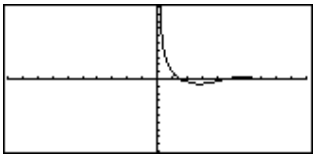
Im Funktions-Graphikmodus mit ZoomStd - Fenster:

```
Graph 1.25a*cos(a),a ENTER
```



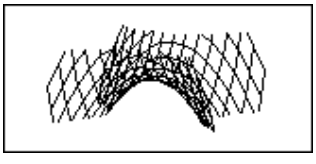
Bei Parameterdarstellung mit ZoomStd - Fenster:

```
Graph
time,2cos(time)/time,time ENTER
```



Im 3D-Graphikmodus:

```
Graph (v^2-w^2)/4,v,w ENTER
```



▸Hex MATH/Base-Menü

Ganze_Zahl1 ▸**Hex** ⇒ *Ganze_Zahl*

Wandelt *Ganze_Zahl1* in eine Hexadezimalzahl um. Dual- oder Hexadezimalzahlen weisen stets das Präfix 0b bzw. 0h auf.

Null (nicht Buchstabe O) und b oder h.

0b *binäre_Zahl*
0h *hexadezimale_Zahl*

Eine Dualzahl kann bis zu 32 Stellen aufweisen, eine Hexadezimalzahl bis zu 8.

Ohne Präfix wird *Ganze_Zahl1* als dezimal (Basis 10) behandelt. Das Ergebnis wird unabhängig vom Base-Modus hexadezimal angezeigt.

Geben Sie eine dezimale ganze Zahl ein, die für eine 32-Bit-Dualform mit Vorzeichen zu groß ist, dann wird eine symmetrische Modulo-Operation ausgeführt, um den Wert in den erforderlichen Bereich zu bringen.

```
256▸Hex ENTER                    0h100
0b111100001111▸Hex ENTER         0hF0F
```

identity() MATH/Matrix-Menü

identity(*Term***)** ⇒ *Matrix*

Gibt die Einheitsmatrix mit der Dimension *Term* zurück.

Term muß zu einer positiven ganzen Zahl auswertbar sein.

```
identity(4) ENTER
```

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

If CATALOG

If *Boolescher Term*
Anweisung

If *Boolescher Term* **Then**
Block
EndIf

Wenn *Boolescher Term* wahr ergibt, wird die Einzelanweisung *Anweisung* oder der Anweisungsblock *Block* ausgeführt und danach hinter EndIf fortgefahren.

Wenn *Boolescher Term* falsch ergibt, wird das Programm fortgesetzt, ohne daß die Einzelanweisung bzw. der Anweisungsblock ausgeführt werden.

Block kann eine einzelne Anweisung oder eine Serie von Anweisungen sein, die durch ":" getrennt sind.

Programmsegment:

```
   ⋮
: If x<0
:Disp "x is negative"
   ⋮
 –oder–
   ⋮
:If x<0 Then
:  Disp "x is negative"
:  abs(x)→x
:EndIf
   ⋮
```

If *Boolescher Term* **Then**
Block1
Else
Block2
EndIf

Wenn *Boolescher Term* wahr ergibt, wird *Block1* ausgeführt und dann *Block2* übersprungen.

Wenn *Boolescher Term* falsch ergibt, wird *Block1* übersprungen, aber *Block2* ausgeführt.

Block1 und *Block2* können einzelne Anweisungen sein.

Programmsegment:

```
   ⋮
: If x<0 Then
:  Disp "x is negative"
:    Else
:  Disp "x is positive or
zero"
:EndIf
   ⋮
```

If *Boolescher Term1* **Then**
Block1
ElseIf *Boolescher Term2* **Then**
Block2
⋮
ElseIf *Boolescher TermN* **Then**
BlockN
EndIf

Gestattet Programmverzweigungen. Wenn *Boolescher Term1* wahr ergibt, wird *Block1* ausgeführt. Wenn *Boolescher Term1* falsch ergibt, wird *Boolescher Term2* ausgewertet usw.

Programmsegment:

```
   ⋮
: If choice=1 Then
:  Goto option1
:  ElseIf choice=2 Then
:    Goto option2
:  ElseIf choice=3 Then
:    Goto option3
:  ElseIf choice=4 Then
:    Disp "Exiting Program"
:    Return
:EndIf
   ⋮
```

imag() MATH/Complex-Menü

imag(*Term1***)** ⇒ *Term*

imag(*Term1***)** gibt den imaginären Teil des Parameters zurück.

Hinweis: Alle undefinierten Variablen werden als reelle Variablen behandelt. Siehe auch **real()**.

```
imag(1+2i) ENTER        2
imag(z) ENTER           0
imag(x+iy) ENTER        y
```

imag(*Liste1***)** ⇒ *Liste*

Gibt eine Liste der Imaginärteile der Elemente zurück.

```
imag({-3,4-i,i}) ENTER    {0 -1 1}
```

imag(*Matrix1***)** ⇒ *Matrix*

Gibt eine Matrix der Imaginärteile der Elemente zurück.

```
imag([a,b;ic,id]) ENTER      [0 0]
                             [c d]
```

Input CATALOG

Input

Hält das Programm an, zeigt den aktuellen Graphikbildschirm an und gibt Ihnen so die Möglichkeit, die Variablen *xc* und *yc* (sowie im Polar-Koordinatenmodus auch *rc* und θ*c*) durch Neupositionieren des Graphikcursors zu aktualisieren.

Das Programm wird fortgesetzt, wenn Sie ENTER drücken.

Programmsegment:

```
  :
:◉ Get 10 points from the Graph
   Screen
:For i,1,10
:  Input
:  xc→XLIST[i]
:  yc→YLIST[i]
:EndFor
  :
```

Input [*AuffordString*,] *Var*

Input [*AuffordString*], *Var* hält das Programm an, zeigt *AuffordString* auf dem Programm-I/O-Bildschirm an, wartet auf Ihre Eingabe und speichert diese in der Variablen *Var*.

Wenn Sie *AuffordString* weglassen, wird "?" als Eingabeaufforderung angezeigt.

Programmsegment:

```
  :
:For i,1,9,1
:  "Enter x" & string(i)→str1
:  Input str1,#(right(str1,2))
:EndFor
  :
```

InputStr CATALOG

InputStr [*AuffordString*,] *Var*

Hält das Programm an, zeigt *AuffordString* auf dem Programm-I/O-Bildschirm an, wartet auf Ihre Eingabe und speichert diese in der Variablen *Var*.

Wenn Sie *AuffordString* weglassen, wird "?" als Eingabeaufforderung angezeigt.

Hinweis: Der Unterschied zwischen **Input** und **InputStr** ist, daß **InputStr** die Eingabe stets als Zeichenkette speichert und daher keine " " erforderlich sind.

Programmsegment:

```
  :
:InputStr "Enter Your Name",str1
  :
```

inString() MATH/String-Menü

inString(*QuellString*, *TeilString*[, *Start*]**)** ⇒ *Ganze_Zahl*

Gibt die Position des Zeichens von *QuellString* zurück, an der das erste Vorkommen von *TeilString* beginnt.

Start legt fest (sofern angegeben), an welcher Zeichenposition innerhalb von *QuellString* die Suche beginnt. Vorgabe = 1 (erstes Zeichen von *QuellString*).

Enthält *QuellString* die Zeichenkette *TeilString* nicht, oder ist *Start* > Länge von *QuellString*, wird 0 zurückgegeben.

```
inString("Hello there","the")
ENTER                          7

"ABCEFG"→s1:If inString(s1,
"D")=0:Disp "D not found." ENTER
                    D not found.
```

int() CATALOG

int(*Term***)** ⇒ *Ganze_Zahl*
int(*Liste1***)** ⇒ *Liste*
int(*Matrix1***)** ⇒ *Matrix*

Gibt die größte ganze Zahl zurück, die kleiner oder gleich dem Parameter ist. Diese Funktion ist identisch mit **floor()**.

Der Parameter kann eine reelle oder eine komplexe Zahl sein.

Für eine Liste oder Matrix wird für jedes Element die größte ganze Zahl zurückgegeben, die kleiner oder gleich dem Element ist.

```
int(-2.5) ENTER                    -3.
int([-1.234,0,0.37]) ENTER
                                [-2. 0 0.]
```

intDiv() CATALOG

intDiv(*Zahl1***, *Zahl2***)** ⇒ *Ganze_Zahl*
intDiv(*Liste1***, *Liste2***)** ⇒ *Liste*
intDiv(*Matrix1***, *Matrix2***)** ⇒ *Matrix*

Gibt den mit Vorzeichen versehenen ganzzahligen Teil von Parameter 1 dividiert durch Parameter 2 zurück.

Für eine Liste oder Matrix wird für jedes Elementpaar der mit Vorzeichen versehene ganzzahlige Teil von Parameter 1 dividiert durch Parameter 2 zurückgegeben.

```
intDiv(-7,2) ENTER                 -3
intDiv(4,5) ENTER                   0
intDiv({12,-14,-16},{5,4,-3})
ENTER
                                {2 -3 5}
```

integrate() Siehe ∫(), Seite 989.

iPart() MATH/Number-Menü

iPart(*Zahl***)** ⇒ *Ganze_Zahl*
iPart(*Liste1***)** ⇒ *Liste*
iPart(*Matrix1***)** ⇒ *Matrix*

Gibt den Ganzzahlanteil des Parameters zurück.

Für eine Liste oder Matrix wird der Ganzzahlanteil jedes Elements zurückgegeben.

Der Parameter kann eine reelle oder eine komplexe Zahl sein.

```
iPart(-1.234) ENTER                -1.
iPart({3/2,-2.3,7.003}) ENTER
                                {1 -2. 7.}
```

isClkOn() CATALOG

isClkOn() ⇒ *true,false*

Stellt fest, ob die Uhr AN oder AUS ist. Gibt Wahr zurück, wenn die Uhr AN ist und Falsch, wenn die Uhr AUS ist.

isPrime() MATH/Test-Menü

IsPrime(*Zahl***)** ⇒ *Boolescher konstanter Term*

Gibt "wahr" oder "falsch" zurück, um anzuzeigen, ob es sich bei *Zahl* um eine ganze Zahl ≥ 2 handelt, die nur durch sich selbst oder 1 ganzzahlig teilbar ist.

Übersteigt *Zahl* ca. 306 Stellen und hat sie keine Faktoren ≤ 1021, dann zeigt **isPrime(***Zahl***)** eine Fehlermeldung an.

Möchten Sie lediglich feststellen, ob es sich bei *Zahl* um eine Primzahl handelt, verwenden Sie **isPrime()** anstelle von **factor()**. Dieser Vorgang ist wesentlich schneller, insbesondere dann, wenn *Zahl* keine Primzahl ist und ihr zweitgrößter Faktor ca. fünf Stellen übersteigt.

```
IsPrime(5) ENTER                    true
IsPrime(6) ENTER                   false
```

Funktion zum Auffinden der nächsten Primzahl nach einer angegebenen Zahl:

```
Define nextPrim(n)=Func:Loop:
n+1→n:if isPrime(n):return n:
EndLoop:EndFunc ENTER           Done

nextPrim(7) ENTER                    11
```

Item CATALOG

Item *itemNameString*
Item *itemNameString***,** *Label*

Nur gültig innerhalb eines Blocks **Custom...EndCustm** oder **ToolBar...EndTBar**. Richtet ein Dropdown-Menüelement ein, mit dem Sie Text an der Cursorposition einfügen (**Custom**) oder zu einem Label verzweigen (**ToolBar**) können.

Hinweis: Das Verzweigen zu einer Marke ist innerhalb eines **Custom** Blocks nicht zulässig.

Siehe Beispiel zu **Custom**.

Lbl CATALOG

Lbl *LabelName*

Definiert in einem Programm eine Marke mit dem Namen *LabelName*.

Mit der Anweisung **Goto** *LabelName* können Sie die Programmausführung an der Anweisung fortsetzen, die unmittelbar auf die Marke folgt.

Für *LabelName* gelten die gleichen Benennungsregeln wie für einen Variablennamen.

Programmsegment:

```
  ⋮
:Lbl lbl1
:InputStr "Enter password",
str1
:If str1≠password
:  Goto lbl1
:Disp "Welcome to ..."
  ⋮
```

lcm() MATH/Number-Menü

lcm(*Zahl1***,** *Zahl2***)** ⇒ *Term*
lcm(*Liste1***,** *Liste2***)** ⇒ *Liste*
lcm(*Matrix1***,** *Matrix2***)** ⇒ *Matrix*

Gibt das kleinste gemeinsame Vielfache der beiden Parameter zurück. Das **lcm** zweier Brüche ist das **lcm** der Zähler dividiert durch den ggT (**gcd)** ihrer Nenner. Das **lcm** von Dezimalbruchzahlen ist ihr Produkt.

Für zwei Listen oder Matrizen wird das kleinste gemeinsame Vielfache der entsprechenden Elemente zurückgegeben.

```
lcm(6,9) ENTER                       18

lcm({1/3,-14,16},{2/15,7,5})
ENTER
                          {2/3 14 80}
```

left() MATH/String-Menü

left(*QuellString***[,** *Anzahl***])** ⇒ *String*

Gibt *Anzahl* Zeichen zurück, die links in der Zeichenkette *QuellString* enthalten sind.

Wenn Sie *Anzahl* weglassen, wird der gesamte *QuellString* zurückgegeben.

```
left("Hello",2) ENTER              "He"
```

left(*Liste1*[**,** *Anzahl*]**)** ⇒ *Liste*

Gibt *Anzahl* Elemente zurück, die links in *Liste1* enthalten sind.

Wenn Sie *Anzahl* weglassen, wird die gesamte *Liste1* zurückgegeben.

```
left({1,3,-2,4},3) ENTER
                          {1 3 -2}
```

left(*Vergleich***)** ⇒ *Term*

Gibt die linke Seite einer Gleichung oder Ungleichung zurück.

```
left(x<3) ENTER                    x
```

limit() MATH/Calculus-Menü

limit(*Term1***,** *Var***,** *Punkt*[**,** *Richtung*]**)** ⇒ *Term*
limit(*Liste1***,** *Var***,** *Punkt*[**,** *Richtung*]**)** ⇒ *Liste*
limit(*Matrix1***,** *Var***,** *Punkt*[**,** *Richtung*]**)** ⇒ *Matrix*

Gibt den angeforderten Grenzwert zurück.

Richtung: negativ=von links, positiv=von rechts, ansonsten=beide. (Wird keine Angabe gemacht, gilt für *Richtung* die Vorgabe beide.)

```
limit(2x+3,x,5) ENTER              13
limit(1/x,x,0,1) ENTER              ∞
limit(sin(x)/x,x,0) ENTER           1
limit((sin(x+h)-sin(x))/h,h,0)
ENTER
                                 cos(x)
limit((1+1/n)^n,n,∞) ENTER          e
```

Grenzen bei positiv ∞ und negativ ∞ werden stets zu einseitigen Grenzen von der endlichen Seite aus umgewandelt.

Je nach den Umständen gibt **limit()** sich selbst oder undef zurück, wenn kein eindeutiger Grenzwert ermittelt werden kann. Das heißt nicht unbedingt, daß es keinen eindeutigen Grenzwert gibt. undef bedeutet lediglich, daß das Ergebnis entweder eine unbekannte Zahl endlicher oder unendlicher Größenordnung ist, oder es ist die Gesamtmenge dieser Zahlen.

limit() arbeitet mit Verfahren wie der Regel von L'Hospital; es gibt daher eindeutige Grenzwerte, die es nicht ermitteln kann. Wenn *Term1* über *Var* hinaus weitere undefinierte Variablen enthält, müssen Sie möglicherweise Einschränkungen dafür verwenden, um ein brauchbareres Ergebnis zu erhalten.

```
limit(a^x,x,∞) ENTER             undef
limit(a^x,x,∞)|a>1 ENTER            ∞
limit(a^x,x,∞)|a>0 and a<1
ENTER                               0
```

Grenzwerte können sehr anfällig für Rundungsfehler sein. Vermeiden Sie nach Möglichkeit die Einstellung APPROX für den Modus Exact/Approx sowie Näherungszahlen beim Berechnen von Grenzwerten. Andernfalls kann es sein, daß Grenzen, die Null oder unendlich sein müssten, dies nicht sind und umgekehrt endliche Grenzwerte ungleich Null nicht erkannt werden.

Line CATALOG

Line *xStart*, *yStart*, *xEnd*, *yEnd*[, *Zeichenmodus*]

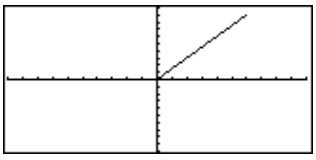
Öffnet den Graphikbildschirm und zeichnet, löscht oder invertiert eine Strecke zwischen den Fensterkoordinaten (*xStart*, *yStart*) und (*xEnd*, *yEnd*), wobei beide Endpunkte eingeschlossen sind.

Wenn *Zeichenmodus* = 1, wird die Strecke gezeichnet (Vorgabe).
Wenn *Zeichenmodus* = 0, wird die Strecke gelöscht.
Wenn *Zeichenmodus* = ⁻1, wird die Strecke invertiert (d. h. auf "ein" gesetzt, wenn sie "aus" ist und umgekehrt)

Hinweis: Beim Neuzeichnen werden alle gezeichneten Elemente gelöscht. Siehe auch **PxlLine**.

Zeichnen Sie im ZoomStd-Fenster eine Strecke und löschen Sie sie anschließend.

`Line 0,0,6,9` ENTER



HOME

◆ [CALC HOME]

`Line 0,0,6,9,0` ENTER

LineHorz CATALOG

LineHorz *y* [, *Zeichenmodus*]

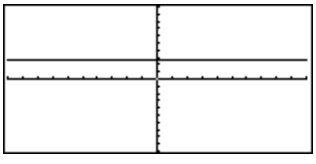
Öffnet den Graphikbildschirm und zeichnet, löscht oder invertiert eine horizontale Gerade durch die Fensterposition *y*.

Wenn *Zeichenmodus* = 1, wird die Gerade gezeichnet (Vorgabe).
Wenn *Zeichenmodus* = 0, wird die Gerade gelöscht.
Wenn *Zeichenmodus* = ⁻1, wird die Gerade invertiert (d. h. auf "ein" gesetzt, wenn sie "aus" ist und umgekehrt).

Hinweis: Beim Neuzeichnen werden alle gezeichneten Elemente gelöscht. Siehe auch **PxlHorz**.

In einem ZoomStd-Fenster :

`LineHorz 2.5` ENTER



LineTan CATALOG

LineTan *Term1*, *Term2*

Öffnet den Graphikbildschirm und zeichnet eine Tangente für *Term1* durch den angegebenen Punkt.

Term1 ist ein Term oder der Name einer Funktion, wobei x als unabhängige Variable angenommen wird und *Term2* der x-Wert des Berührpunkts der Tangente.

Hinweis: Im gezeigten Beispiel wird *Term1* separat gezeichnet. **LineTan** zeichnet nicht *Term1.*

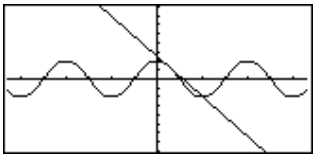
Im Funktions-Graphikmodus mit einem ZoomTrig -Fenster :

`Graph cos(x)`

HOME

◆ [CALC HOME]

`LineTan cos(x),π/4` ENTER



LineVert CATALOG

LineVert *x* [**,** *Zeichenmodus*]

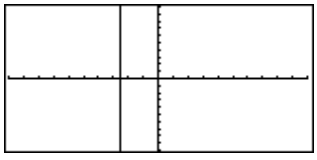
Öffnet den Graphikbildschirm und zeichnet, löscht oder invertiert eine vertikale Gerade durch die Fensterposition *x*.

Wenn *Zeichenmodus* = 1, wird die Gerade gezeichnet (Vorgabe).
Wenn *Zeichenmodus* = 0, wird die Gerade gelöscht.
Wenn *Zeichenmodus* = ⁻1, wird die Gerade invertiert (d. h. auf "ein" gesetzt, wenn sie "aus" ist und umgekehrt).

Hinweis: Beim Neuzeichnen werden alle gezeichneten Elemente gelöscht. Siehe auch **PxlVert**.

In einem ZoomStd-Fenster:

`LineVert -2.5` ENTER



LinReg MATH/Statistics/Regressions-Menü

LinReg *Liste1*, *Liste2*[, [*Liste3*] [, *Liste4*, *Liste5*]]

Berechnet die lineare Regression und aktualisiert alle Statistik-Systemvariablen.

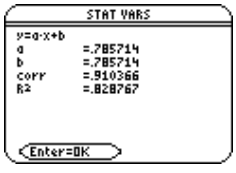
Alle Listen außer *Liste5* müssen die gleiche Dimension besitzen.

Liste1 stellt die Liste der x-Werte dar.
Liste2 stellt die Liste der y-Werte dar.
Liste3 stellt die Angaben für die Häufigkeit dar.
Liste4 stellt die Klassencodes dar.
Liste5 stellt die Klassenliste dar.

Hinweis: *Liste1* bis einschl. *Liste4* müssen Variablenamen oder c1–c99 sein (Spalten in der letzten Datenvariablen, die im Daten/ Matrix-Editor angezeigt wurde). *Liste5* braucht kein Variablenname zu sein und kann keine Spalte c1–c99 sein.

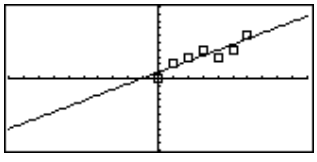
Im Funktions-Graphikmodus:

`{0,1,2,3,4,5,6}→L1` ENTER
`{0 1 2 ...}`
`{0,2,3,4,3,4,6}→L2` ENTER
`{0 2 3 ...}`
`LinReg L1,L2` ENTER `Done`
`ShowStat` ENTER



ENTER
`Regeq(x)→y1(x)` ENTER `Done`
`NewPlot 1,1,L1,L2` ENTER `Done`

◆ [GRAPH]



Δlist() MATH/List menu

list(*list1*) ⇒ *list*

Ergibt eine Liste mit den Differenzen der aufeinanderfolgenden Elemente in *list1*. Jedes Element in *list1* wird vom folgenden Element in *list1* subtrahiert. Die Ergebnisliste enthält stets ein Element weniger als die ursprüngliche *list1*.

`Δlist({20,30,45,70})` ENTER
`{10,15,25}`

list▸mat() MATH/List-Menü

list▸mat(*Liste* [**,** *ElementeProZeile*]**)** ⇒ *Matrix*

Gibt eine Matrix zurück, die Zeile für Zeile mit den Elementen aus *Liste* aufgefüllt wurde.

ElementeProZeile gibt (sofern angegeben) die Anzahl der Elemente pro Zeile an. Vorgabe ist die Anzahl der Elemente in *Liste* (eine Zeile).

Wenn *Liste* die resultierende Matrix nicht vollständig auffüllt, werden Nullen hinzugefügt.

```
list▸mat({1,2,3}) ENTER    [1 2 3]

list▸mat({1,2,3,4,5},2) ENTER
                          [1 2]
                          [3 4]
                          [5 0]
```

ln() 2nd [LN] Taste LN Taste

ln(*Term1***)** ⇒ *Term*
ln(*Liste1***)** ⇒ *Liste*

Gibt den natürlichen Logarithmus des Parameters zurück.

Gibt für eine Liste der natürlichen Logarithmen der einzelnen Elemente zurück.

```
ln(2.0) ENTER                .693...
```

Bei Komplexformatmodus REAL:

```
ln({-3,1.2,5}) ENTER
          Error: Non-real result
```

Bei Komplexformatmodus RECTANGULAR:

```
ln({-3,1.2,5}) ENTER
     {ln(3) + π·i  .182...  ln(5)}
```

---

ln(*quadrat_Matrix1***)** ⇒ *quadrat_Matrix*

Ergibt den natürlichen Matrix-Logarithmus von *quadrat_Matrix1*. Dies ist *nicht* gleichbedeutend mit der Berechnung des natürlichen Logarithmus jedes einzelnen Elements. Näheres zum Berechnungsverfahren finden Sie im Abschnitt **cos()**.

Quadrat_Matrix1 muß diagonalisierbar sein. Das Ergebnis enthält stets Fließkommazahlen.

Im Winkelmodus Radian und im Komplex-Formatmodus "Rectangular":

```
ln([1,5,3;4,2,1;6,-2,1]) ENTER

[1.831...+1.734...·i   .009...-1.490...·i   ...]
[.448...-.725...·i     1.064...+.623·i      ...]
[-.266...-2.083...·i   1.124...+1.790...·i  ...]
```

LnReg MATH/Statistics/Regressions-Menü

LnReg *Liste1*, *Liste2*[, [*Liste3*] [, *Liste4*, *Liste5*]]

Berechnet die logarithmische Regression und aktualisiert alle Statistik-Systemvariablen.

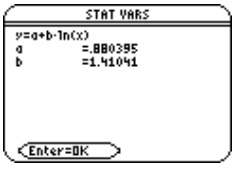
Alle Listen außer *Liste5* müssen die gleiche Dimension besitzen.

Liste1 stellt die Liste der x-Werte dar.
Liste2 stellt die Liste der y-Werte dar.
Liste3 stellt die Angaben für die Häufigkeit dar.
Liste4 stellt die Klassencodes dar.
Liste5 stellt die Klassenliste dar.

Hinweis: *Liste1* bis einschl. *Liste4* müssen Variablenamen oder c1–c99 sein (Spalten in der letzten Datenvariablen, die im Daten/ Matrix-Editor angezeigt wurde). *Liste5* braucht kein Variablenname zu sein und kann keine Spalte c1–c99 sein.

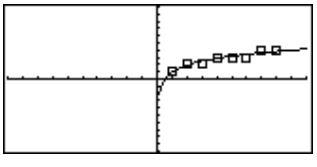
Im Funktions-Graphikmodus:

```
{1,2,3,4,5,6,7,8}→L1 ENTER
                          {1 2 3 ...}
{1,2,2,3,3,3,4,4}→L2 ENTER
                          {1 2 2 ...}
LnReg L1,L2 ENTER              Done
ShowStat ENTER
```




```
ENTER
Regeq(x)→y1(x) ENTER           Done
NewPlot 1,1,L1,L2 ENTER        Done
```

◆ [GRAPH]



Local CATALOG

Local *Var1*[, *Var2*] [, *Var3*] ...

Deklariert die angegebenen Variablen *VarX* als lokale Variablen. Diese Variablen existieren nur für die Dauer der Ausführung eines Programms oder einer Funktion und werden gelöscht, wenn das Programm oder die Funktion beendet wird.

Hinweis: Lokale Variablen sparen Speicherplatz, da sie nur temporär existieren. Außerdem stören sie keine vorhandenen globalen Variablenwerte. Sie müssen lokale Variablen in **For**-Schleifen verwenden und für das temporäre Speichern von Werten in mehrzeiligen Funktionen, da Änderungen globaler Variablen in einer Funktion unzulässig sind.

Programmlisting:

```
:prgmname()
:Prgm
:Local x,y
:Input "Enter x",x
:Input "Enter y",y
:Disp x*y
:EndPrgm
```

Hinweis: *x* und *y* existieren nach Programmausführung nicht mehr.

Lock CATALOG

Lock *Var1*[**,** *Var2*] **...**

Sperrt die angegebenen Variablen. Sie verhindern damit das versehentliche Löschen oder Ändern der Variablen. Löschen oder Ändern können Sie die Variable(n) erst nach Aufheben der Sperrung.

Im rechts gezeigten Beispiel ist die Variable L1 gesperrt und kann weder gelöscht noch geändert werden.

Hinweis: Sie heben die Sperrung mit dem Befehl **unlock** auf**.**

```
{1,2,3,4}→L1 ENTER          {1,2,3,4}
Lock L1 ENTER                    Done
DelVar L1 ENTER
Error: Variable is locked or protected
```

log() CATALOG

log(*Term1***)** ⇒ *Term*
log(*Liste1***)** ⇒ *Liste*

Gibt den Zehnerlogarithmus des Parameters zurück.

Gibt für eine Liste den Zehnerlogarithmus der einzelnen Elemente zurück.

```
log(2.0) ENTER                   .301…
```

Bei Komplexformatmodus REAL:

```
log({-3,1.2,5}) ENTER
          Error: Non-real result
```

Bei Komplexformatmodus RECTANGULAR:

```
log({-3,1.2,5}) ENTER
```

$$\left\{\frac{\ln(3)}{\ln(10)}+\frac{\pi}{\ln(10)}\cdot i \quad .079\ldots \quad \frac{\ln(5)}{\ln(10)}\right\}$$

---

log(*quadrat_Matrix1***)** ⇒ *quadrat_Matrix*

Ergibt den dekadischen Matrix-Logarithmus von *quadrat_Matrix1*. Dies ist *nicht* gleichbedeutend mit der Berechnung des dekadischen Logarithmus jedes einzelnen Elements. Näheres zum Berechnungsverfahren finden Sie im Abschnitt **cos()**.

Quadrat_Matrix1 muß diagonalisierbar sein. Das Ergebnis enthält stets Fließkommazahlen.

Im Winkelmodus Radian und im Komplex-Formatmodus “Rectangular”:

```
log([1,5,3;4,2,1;6,-2,1]) ENTER
```

$$\begin{bmatrix} .795\ldots+.753\ldots\cdot i & .003\ldots-.647\ldots\cdot i & \ldots \\ .194\ldots-.315\ldots\cdot i & .462\ldots+.270\cdot i & \ldots \\ -.115\ldots-.904\ldots\cdot i & .488\ldots+.777\ldots\cdot i & \ldots \end{bmatrix}$$

Logistic MATH/Statistics/Regressions-Menü

Logistic *Liste1*, *Liste2* [, [*Iterationen*], [*Liste3*] [, *Liste4*, *Liste5*]]

Berechnet die logistische Regression und aktualisiert alle Systemstatistik-Variablen.

Alle Listen außer *Liste5* müssen die gleiche Dimension besitzen.

Liste1 stellt die Liste der x-Werte dar.
Liste2 stellt die Liste der y-Werte dar.
Liste3 stellt die Angaben für die Häufigkeit dar.
Liste4 stellt die Klassencodes dar.
Liste5 stellt die Klassenliste dar.

Iterationen gibt an, wie viele Lösungsversuche maximal stattfinden. Bei Auslassung wird 64 verwendet. Größere Werte führen in der Regel zu höherer Genauigkeit aber auch längeren Ausführungszeiten und umgekehrt.

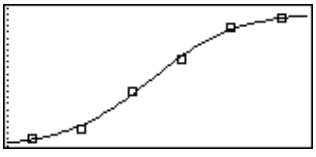
Hinweis: *Liste1* bis einschl. *Liste4* müssen Variablennamen oder c1–c99 sein (Spalten in der letzten Datenvariablen, die im Daten/ Matrix-Editor angezeigt wurde). *Liste5* braucht kein Variablenname zu sein und kann keine Spalte c1–c99 sein.

Im Funktions-Graphikmodus:

```
{1,2,3,4,5,6}→L1 ENTER {1 2 3 …}
{1,1.3,2.5,3.5,4.5,4.8}→L2
ENTER
                          {1 1.3 2.5 …}
Logistic L1,L2 ENTER            Done
ShowStat ENTER
```

```
ENTER
regeq(x)→y1(x) ENTER            Done
NewPlot 1,1,L1,L2 ENTER         Done
```

◆ [GRAPH]
F2 9



Loop CATALOG

Loop
Block
EndLoop

Führt die in *Block* enthaltenen Anweisungen wiederholt aus. Beachten Sie, daß dies eine Endlosschleife ist. Beenden Sie sie, indem Sie eine Anweisung **Goto** oder **Exit** in *Block* ausführen.

Block ist eine Folge von Anweisungen, die durch das Zeichen ":" voneinander getrennt sind.

Programmsegment:

```
  ⋮
:1→i
:Loop
:  Rand(6)→die1
:  Rand(6)→die2
:  If die1=6 and die2=6
:    Goto End
:  i+1→i
:EndLoop
:Lbl End
:Disp "The number of rolls is", i
  ⋮
```

LU MATH/Matrix-Menü

LU *Matrix*, *lMatName*, *rMatName*, *pMatName*[, *Tol*]

Berechnet die Doolittle LR-Zerlegung (links/rechts; im Englischen LU-lower/upper) einer reellen oder komplexen *Matrix*. Die linke Dreiecksmatrix ist in *lMatName* gespeichert, die rechte Dreiecksmatrix in *rMatName* und die Permutationsmatrix (in welcher der bei der Berechnung vorgenommene Zeilentausch dokumentiert ist) in *pMatName*.

lMatName ∗ *rMatName* = *pMatName* ∗ *Matrix*

Sie haben die Option, daß jedes Matrixelement als Null behandelt wird, wenn dessen absoluter Wert geringer als *Tol* ist. Diese Toleranz wird nur dann verwendet, wenn die Matrix Fließkommaelemente enthält und keinerlei symbolische Variablen ohne zugewiesene Werte enthält. Anderenfalls wird *Tol* ignoriert.

- Wenn Sie ◆ ENTER verwenden oder den Modus auf Exact/Approx=APPROXIMATE einstellen, werden Berechnungen im Fließkomma-Rechenmodus durchgeführt.
- Wird *Tol* weggelassen oder nicht verwendet, so wird die Standardtoleranz folgendermaßen berechnet:

  5E⁻14 ∗ **max(dim(***matrix***))** ∗ **rowNorm(***matrix***)**

  Der **LU**-Faktorisierungsalgorithmus verwendet partielle Pivotisierung mit Zeilentausch.

```
[6,12,18;5,14,31;3,8,18] →m1 ENTER
```

$$\begin{bmatrix} 6 & 12 & 18 \\ 5 & 14 & 31 \\ 3 & 8 & 18 \end{bmatrix}$$

```
LU m1,lower,upper,perm ENTER  Done
```

```
lower ENTER
```

$$\begin{bmatrix} 1 & 0 & 0 \\ 5/6 & 1 & 0 \\ 1/2 & 1/2 & 1 \end{bmatrix}$$

```
upper ENTER
```

$$\begin{bmatrix} 6 & 12 & 18 \\ 0 & 4 & 16 \\ 0 & 0 & 1 \end{bmatrix}$$

```
perm ENTER
```

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

```
[m,n;o,p] →m1 ENTER
```

$$\begin{bmatrix} m & n \\ o & p \end{bmatrix}$$

```
LU m1,lower,upper,perm ENTER  Done
```

```
lower ENTER
```

$$\begin{bmatrix} 1 & 0 \\ \frac{m}{o} & 1 \end{bmatrix}$$

```
upper ENTER
```

$$\begin{bmatrix} o & p \\ 0 & n - \frac{m \cdot p}{o} \end{bmatrix}$$

```
perm ENTER
```

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

mat▸list() MATH/List-Menü

mat▸list(*Matrix***)** ⇒ *Liste*

Gibt eine Liste zurück, die mit den Elementen aus *Matrix* gefüllt wurde. Die Elemente werden Zeile für Zeile aus *Matrix* kopiert.

```
mat▸list([1,2,3]) ENTER   {1 2 3}
[1,2,3;4,5,6]→ M1 ENTER
```

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

```
mat▸list(M1) ENTER  {1 2 3 4 5 6}
```

max() MATH/List-Menü

max(*Term1***, *Term2***)** ⇒ *Term*
max(*Liste1***, *Liste2***)** ⇒ *Liste*
max(*Matrix1***, *Matrix2***)** ⇒ *Matrix*

Gibt das Maximum der beiden Parameter zurück. Wenn die Parameter zwei Listen oder Matrizen sind, wird eine Liste bzw. Matrix zurückgegeben, die den Maximalwert für jedes entsprechende Elementpaar enthält.

```
max(2.3,1.4) ENTER                    2.3
max({1,2},{-4,3}) ENTER               {1 3}
```

---

max(*Liste***)** ⇒ *Term*

Gibt das größte Element von *Liste* zurück.

```
max({0,1,-7,1.3,.5}) ENTER            1.3
```

---

max(*Matrix1***)** ⇒ *Matrix*

Gibt einen Zeilenvektor zurück, der das größte Element jeder Spalte von *Matrix1* enthält.

Hinweis: Siehe auch **fMax()** und **min()**.

```
max([1,-3,7;-4,0,.3]) ENTER
                              [1 0 7]
```

mean() MATH/Statistics-Menü

mean(*list*[*, freqlist*]**)** ⇒ *expression*

Ergibt das arithmetische Mittel der Elemente in *list*.

Jedes *freqlist*-Element gewichtet die Elemente von *list* in der gegebenen Reihenfolge entsprechend.

```
mean({.2,0,1,-.3,.4}) ENTER           .26
mean({1,2,3},{3,2,1}) ENTER           5/3
```

---

mean(*matrix1*[*, freqmatrix*]**)** ⇒ *matrix*

Ergibt einen Zeilenvektor aus den arithmetischen Mittelwerten der Spalten in *matrix1.*

Jedes *freqmatrix*-Element gewichtet die Elemente von *matrix1* in der gegebenen Reihenfolge entsprechend.

Im Modus Vektorformat rectangular:

```
mean([.2,0;-1,3;.4,-.5]) ENTER
                        [-.133... .833...]
mean([1/5,0;-1,3;2/5,-1/2])
ENTER
                           [-2/15  5/6]
mean([1,2;3,4;5,6],[5,3;4,1;
6,2]) ENTER          [47/15, 11/3]
```

median() MATH/Statistics-Menü

median(*Liste***)** ⇒ *Term*

Gibt den Medianwert der Elemente von *Liste* zurück.

```
median({.2,0,1,-.3,.4}) ENTER         .2
```

---

median(*Matrix1***)** ⇒ *Matrix*

Gibt einen Zeilenvektor zurück, der die Medianwerte der einzelnen Spalten von *Matrix1* enthält.

Hinweis: Alle Elemente der Liste bzw. der Matrix müssen zu Zahlen vereinfachbar sein.

```
median([.2,0;1,-.3;.4,-.5])
ENTER
                              [.4 -.3]
```

MedMed MATH/Statistics/Regressions-Menü

MedMed *Liste1*, *Liste2*[, [*Liste3*] [, *Liste4*, *Liste5*]]

Berechnet die Median-Median-Gerade und aktualisiert alle Statistik-Systemvariablen.

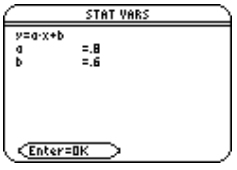
Alle Listen außer *Liste5* müssen die gleiche Dimension besitzen.

Liste1 stellt die Liste der x-Werte dar.
Liste2 stellt die Liste der y-Werte dar.
Liste3 stellt die Angaben für die Häufigkeit dar.
Liste4 stellt die Klassencodes dar.
Liste5 stellt die Klassenliste dar.

Hinweis: *Liste1* bis einschl. *Liste4* müssen Variablenamen oder c1–c99 sein (Spalten in der letzten Datenvariablen, die im Daten/ Matrix-Editor angezeigt wurde). *Liste5* braucht kein Variablenname zu sein und kann keine Spalte c1–c99 sein.

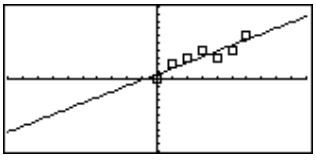
Im Funktions-Graphikmodus:

```
{0,1,2,3,4,5,6}→L1 ENTER {0 1 2 …}
{0,2,3,4,3,4,6}→L2 ENTER {0 2 3 …}
MedMed L1,L2 ENTER              Done
ShowStat ENTER
```




```
ENTER
Regeq(x)→y1(x) ENTER            Done
NewPlot 1,1,L1,L2 ENTER         Done
```

◆ [GRAPH]



mid() MATH/String-Menü

mid(*QuellString*, *Start*[, *Anzahl*]**)** ⇒ *String*

Gibt einen *Anzahl* Zeichen langen *String* aus der Zeichenkette *QuellString* ab dem Zeichen mit der Nummer *Start* zurück

Wird *Anzahl* weggelassen oder ist sie größer als die Länge von *QuellString*, werden alle Zeichen von *QuellString* ab dem Zeichen mit der Nummer *Start* zurückgegeben.

Anzahl muß ≥ 0 sein. Bei *Anzahl* = 0 wird eine leere Zeichenkette zurückgegeben.

```
mid("Hello Sie da",2) ENTER
                    "ello there"
mid("Hello Sie da",7,3) ENTER
                          "the"
mid("Hello Sie da",1,5) ENTER
                        "Hello"
mid("Hello Sie da",1,0) ENTER
                            ""
```

---

mid(*QuellListe*, *Start* [, *Anzahl*]**)** ⇒ *Liste*

Gibt *Anzahl* Elemente aus *QuellListe* ab dem Element mit der Nummer *Start* zurück.

Wird *Anzahl* weggelassen oder ist sie größer als die Dimension von *QuellListe*, werden alle Elemente von *QuellListe* ab dem Element mit der Nummer *Start* zurückgegeben.

Anzahl muß ≥ 0 sein. Bei *Anzahl* = 0 wird eine leere Liste zurückgegeben.

```
mid({9,8,7,6},3) ENTER      {7 6}
mid({9,8,7,6},2,2) ENTER    {8 7}
mid({9,8,7,6},1,2) ENTER    {9 8}
mid({9,8,7,6},1,0) ENTER      {}
```

---

mid(*QuellStringListe*, *Start*[, *Anzahl*]**)** ⇒ *Liste*

Gibt *Anzahl* Elemente aus *QuellStringListe* ab dem Element mit der Nummer *Start* zurück.

```
mid({"A","B","C","D"},2,2) ENTER
                          {"B" "C"}
```

min() MATH/List-Menü

min(*Term1***, *Term2***)** ⇒ *Term*
min(*Liste1, Liste2***)** ⇒ *Liste*
min(*Matrix1, Matrix2***)** ⇒ *Matrix*

Gibt das Minimum der beiden Parameter zurück. Wenn die Parameter zwei Listen oder Matrizen sind, wird eine Liste bzw. Matrix zurückgegeben, die den Minimalwert für jedes entsprechende Elementpaar enthält.

```
min(2.3,1.4) ENTER                    1.4
min({1,2},{-4,3}) ENTER          {-4 2}
```

---

min(*Liste***)** ⇒ *Term*

Gibt das kleinste Element von *Liste* zurück.

```
min({0,1,-7,1.3,.5}) ENTER            -7
```

---

min(*Matrix1***)** ⇒ *Matrix*

Gibt einen Zeilenvektor zurück, der das kleinste Element jeder Spalte von *Matrix1* enthält.

Hinweis: Siehe auch **fMin()** und **max()**.

```
min([1,-3,7;-4,0,.3]) ENTER
                          [-4 -3 .3]
```

mod() MATH/Number-Menü

mod(*Term1***, *Term2***)** ⇒ *Term*
mod(*Liste1***, *Liste2***)** ⇒ *Liste*
mod(*Matrix1***, *Matrix2***)** ⇒ *Matrix*

Gibt erster Parameter modulo zweiter Parameter gemäß der folgenden Identitäten zurück:

mod(x,0) ≡ x
mod(x,y) ≡ x– y floor(x/y)

Ist der zweite Parameter ungleich Null, ist das Ergebnis in diesem Parameter periodisch. Das Ergebnis ist entweder Null oder besitzt das gleiche Vorzeichen wie der zweite Parameter.

Sind die Parameter zwei Listen bzw. zwei Matrizen, wird eine Liste bzw. Matrix zurückgegeben, die den Modulus jedes Elementpaars enthält.

Hinweis: Siehe auch **remain()**.

```
mod(7,0) ENTER                        7
mod(7,3) ENTER                        1
mod(-7,3) ENTER                       2
mod(7,-3) ENTER                      -2
mod(-7,-3) ENTER                     -1
mod({12,-14,16},{9,7,-5}) ENTER
                              {3 0 -4}
```

MoveVar CATALOG

MoveVar *Var*, *VerzAlt*, *VerzNeu*

Verschiebt die Variable *Var* aus *VerzAlt* in *VerzNeu*. Existiert *VerzNeu* noch nicht, wird er von **MoveVar** erstellt.

```
{1,2,3,4}→L1 ENTER          {1 2 3 4}
MoveVar L1,Main,Games ENTER  Done
```

mRow() MATH/Matrix/Row ops-Menü

mRow(*Term***, *Matrix1***, *Index***)** ⇒ *Matrix*

Gibt eine Kopie von *Matrix1* zurück, in der jedes Element der Zeile *Index* von *Matrix1* mit *Term* multipliziert ist.

```
mRow(-1/3,[1,2;3,4],2) ENTER
                          [ 1    2  ]
                          [-1  -4/3 ]
```

mRowAdd() MATH/Matrix/Row ops-Menü

mRowAdd(*Term***, *Matrix1***, *Index1***, *Index2***)** ⇒ *Matrix*

Gibt eine Kopie von *Matrix1* zurück, wobei jedes Element in Zeile *Index2* von *Matrix1* ersetzt wird durch:

Term × Zeile *Index1* + Zeile *Index2*

```
mRowAdd(-3,[1,2;3,4],1,2) ENTER
                          [1   2]
                          [0  -2]

mRowAdd(n,[a,b;c,d],1,2) ENTER
                          [a        b      ]
                          [a·n+c    b·n+d  ]
```

nCr() MATH/Probability-Menü

nCr(*Term1***, *Term2***)** ⇒ *Term*

Für ganzzahlige *Term1* und *Term2* mit *Term1* ≥ *Term2* ≥ 0 ist **nCr()** die Anzahl der Möglichkeiten, *Term2* Elemente aus *Term1* Elementen auszuwählen. (Auch als Binominalkoeffizient bekannt.) Beide Parameter können ganze Zahlen oder symbolische Terme sein.

nCr(*Term***, 0)** ⇒ 1

nCr(*Term, negGanze_Zahl***)** ⇒ 0

nCr(*Term, posGanze_Zahl***)** ⇒ *Term* • (*Term*− 1).. (*Term* − *posGanze_Zahl*+1)/ *posGanze_Zahl*!

nCr(*Term, keineGanze_Zahl***)** ⇒ *Term*!/ ((*Term* − *keineGanze_Zahl*)! • *keineGanze_Zahl*!)

nCr(z,3) $\frac{z\cdot(z-2)\cdot(z-1)}{6}$

ans(1)|z=5 10

nCr(z,c) $\frac{z!}{c!(z-c)!}$

ans(1)/nPr(z,c) $\frac{1}{c!}$

---

nCr(*Liste1***, *Liste2***)** ⇒ *Liste*

Gibt eine Liste von Binomialkoeffizienten auf der Basis der entsprechenden Elementpaare der beiden Listen zurück. Die Parameter müssen Listen gleicher Größe sein.

nCr({5,4,3},{2,4,2}) ENTER

{10 1 3}

---

nCr(*Matrix1***, *Matrix2***)** ⇒ *Matrix*

Gibt eine Matrix von Binomialkoeffizienten auf der Basis der entsprechenden Elementpaare der beiden Matrizen zurück. Die Parameter müssen Matrizen gleicher Größe sein.

nCr([6,5;4,3],[2,2;2,2]) ENTER

$\begin{bmatrix} 15 & 10 \\ 6 & 3 \end{bmatrix}$

nDeriv() MATH/Calculus-Menü

nDeriv(*Term1***, *Var*[, *h*])** ⇒ *Term*
nDeriv(*Term1***, *Var*, *Liste*)** ⇒ *Liste*
nDeriv(*Liste*, *Var*[, *h*])** ⇒ *Liste*
nDeriv(*Matrix*, *Var*[, *h*])** ⇒ *Matrix*

Gibt die numerische Ableitung zurück. Benutzt die Zentraldifferenzenquotientenformel.

h ist die Schrittweite. Wird *h* weggelassen, wird der Vorgabewert 0,001 verwendet.

Wenn Sie *List* oder *Matrix* verwenden, wird die Operation über die Werte in der Liste oder die Matrixelemente abgebildet.

Hinweis: Siehe auch **avgRC()** und ***d*()**.

nDeriv(cos(x),x,h) ENTER

$\frac{-(\cos(x-h)-\cos(x+h))}{2\cdot h}$

limit(nDeriv(cos(x),x,h),h,0) ENTER

$-\sin(x)$

nDeriv(x^3,x,0.01) ENTER

$3.\cdot(x^2+.000033)$

nDeriv(cos(x),x)|x=π/2 ENTER

-1.

nDeriv(x^2,x,{.01,.1}) ENTER

{2.•x 2.•x}

NewData CATALOG

NewData *DatenVar*, *Liste1*[, *Liste2*] [, *Liste3*]...

Erzeugt die Datenvariable *DatenVar,* deren Spalten von den Listen in dieser Reihenfolge gebildet werden.

Mindestens eine Liste ist erforderlich.

Liste1, *Liste2*, ..., *ListeN* können wie abgebildet Listen, zu Listen auflösende Terme oder Listenvariablennamen sein.

NewData macht die neue Variable im Daten/Matrix-Editor zur aktuellen Variablen.

NewData meindat,{1,2,3},{4,5,6} ENTER

Done

(Wechseln Sie zum Daten/Matrix-Editor, und öffnen Sie die Variable *Var* mydata, um die Datenvariable wie abgebildet anzuzeigen.)

DATA				
	c1	c2	c3	
1	1	4		
2	2	5		
3	3	6		
4				

NewData *DatenVar*, *Matrix*

Erzeugt die Datenvariable *DatenVar* auf Grundlage von *Matrix*.

NewData sysData, *Matrix*

Lädt den Inhalt von *Matrix* in die Systemdatenvariable sysData.

NewFold CATALOG

NewFold *Verzeichnisname*

Erzeugt ein benutzerdefiniertes Verzeichnis mit dem Namen *Verzeichnisname* und macht dieses zum aktuellen Verzeichnis. Nachdem diese Anweisung ausgeführt ist, befinden Sie sich im neuen Verzeichnis.

```
NewFold games ENTER          Done
```

newList() CATALOG

newList(*AnzahlElemente***)** ⇒ *Liste*

Gibt eine Liste der Dimension *AnzahlElemente* zurück. Jedes Element ist Null.

```
newList(4) ENTER          {0 0 0 0}
```

newMat() CATALOG/MATH/Matrix-Menü

newMat(*AnzahlZeilen*, *AnzahlSpalten***)** ⇒ *Matrix*

Gibt eine Matrix der Dimension *AnzahlZeilen* mal *AnzahlSpalten* zurück, wobei die Elemente Null sind.

```
newMat(2,3) ENTER          [0 0 0]
                           [0 0 0]
```

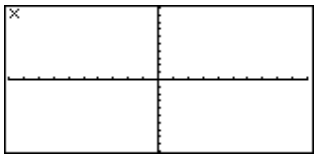
NewPic CATALOG

NewPic *Matrix*, *picVar* [*, maxZeilen*][*, maxSpalten*]

Erzeugt die Bildvariable *picVar* aus *Matrix*. *Matrix* muß eine *nx2*-Matrix sein, in der jede Zeile ein Pixel darstellt. Die Pixel-Koordinaten beginnen mit 0,0. Wenn *picVar* bereits existiert, wird sie von **NewPic** ersetzt.

Die Vorgabe für *picVar* ist die Mindestfläche, die für die Matrixwerte erforderlich ist. Die optionalen Parameter *maxZeilen* und *maxSpalten* legen die maximalen Grenzen für *picVar* fest.

```
NewPic [1,1;2,2;3,3;4,4;5,5;
  5,1;4,2;2,4;1,5],xpic ENTER
                             Done
RclPic xpic ENTER
```



NewPlot CATALOG

NewPlot *n*, *Typ*, *xListe* [,[*yListe*], [*HäufListe*], [*KlassCodeListe*],[*KlassListe*], [*Zeichen*] [, *Stabbreite*]]

Erzeugt eine neue Plot-Definition für Plot Nummer *n*.

Typ legt die Art des Graph-Plots fest.
1 = Streudiagramm
2 = xy-Liniendiagramm
3 = Boxplot
4 = Histogramm
5 = modifizierter Boxplot

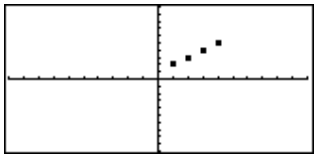
Zeichen legt die Darstellungsform der Werte fest.
1 = ▫ (Kästchen)
2 = × (Kreuz)
3 = + (Pluszeichen)
4 = ▪ (gefülltes Quadrat)
5 = • (Punkt)

Stabbreite ist die Breite eines Histrogrammbalkens (*Typ* = 4) und hängt von den Fenstervariablen xmin und xmax ab. *Stabbreite* muß >0 sein. Vorgabe = 1.

Hinweis: *n* kann im Bereich 1–9 liegen. Die Listen müssen Variablenamen oder c1–c99 sein (Spalten in der letzten Datenvariablen, die im Daten/Matrix-Editor angezeigt wurde). Ausgenommen davon ist *EinbezKatListe*, die kein Variablenname zu sein braucht und keine Spalte c1–c99 sein kann.

```
FnOff ENTER                          Done
PlotsOff ENTER                       Done
{1,2,3,4}→L1 ENTER          {1 2 3 4}
{2,3,4,5}→L2 ENTER          {2 3 4 5}
NewPlot 1,1,L1,L2,,,,4 ENTER         Done
```

Drücken Sie zum Anzeigen ◆ [GRAPH]:



NewProb CATALOG

NewProb

Führt verschiedene Operationen durch, die es Ihnen ermöglichen, eine neue Aufgabe vom gelöschten Zustand aus zu beginnen, ohne dass der Speicher zurückgesetzt werden muß.

- Löscht alle aus einem Zeichen bestehenden Variablennamen (Clear a–z) im aktuellen Verzeichnis, sofern diese nicht geschützt oder archiviert sind.
- Schaltet alle Funktionen und Statplots (**FnOff** und **PlotsOff**) im aktuellen Graphikmodus aus.
- Führt **ClrDraw**, **ClrErr**, **ClrGraph**, **ClrHome**, **ClrIO** und **ClrTable** aus.

```
NewProb ENTER                        Done
```

nInt() MATH/Calculus-Menü

nInt(*Term1, Var, unten, oben***)** ⇒ *Term*

Wenn der Integrand *Term1* außer *Var* keine weiteren Variablen enthält und *unten* und *oben* Konstanten oder positiv ∞ oder negativ ∞ sind, gibt **nInt()** eine Näherung für ∫**(***Term1*, *Var*, *unten*, *oben***)** zurück. Diese Näherung ist der gewichtete Durchschnitt von Stichprobenwerten des Integranden im Intervall *unten*<*Var*<*oben*.

```
nInt(e^(⁻x^2),x,⁻1,1) ENTER
                                     1.493...
```

Das Berechnungsziel sind sechs signifikante Stellen. Der angewendete Algorithmus beendet die Weiterberechnung, wenn das Ziel hinreichend erreicht ist oder wenn weitere Stichproben wahrscheinlich zu keiner sinnvollen Verbesserung führen.

Wenn es scheint, daß das Berechnungsziel nicht erreicht wurde, wird die Meldung "Genauigkeit zweifelhaft" ("Questionable accuracy") angezeigt.

Sie können **nInt()** verschachteln, um mehrere numerische Integrationen durchzuführen. Die Integrationsgrenzen können von außerhalb liegenden Integrationsvariablen abhängen.

Hinweis: Siehe auch ∫**()**.

```
nInt(cos(x),x,⁻π,π+1E⁻12) ENTER
                                  -1.041...E⁻12
∫(cos(x),x,⁻π,π+10^(⁻12)) ENTER
                                  -sin(1/1000000000000)
ans(1)◆ ENTER                     -1.E⁻12
nInt(nInt(e^(⁻x*y)/√(x^2-y^2),
  y,⁻x,x),x,0,1) ENTER            3.304...
```

norm( ) MATH/Matrix/Norms-Menü

norm(*Matrix***)** ⇒ *Term*

Gibt die Frobeniusdeterminante zurück.

```
norm([a,b;c,d]) ENTER
                                  √(a²+b²+c²+d²)
norm([1,2;3,4]) ENTER             √30
```

not MATH/Test-Menü

not *Boolescher Term1* ⇒ *Boolescher Term*

Gibt wahr, falsch oder einen vereinfachten *Booleschen Term1* zurück.

```
not 2>=3                          true
not x<2 ENTER                     x ≥ 2
not not innocent ENTER            innocent
```

---

not *Ganze_Zahl1* ⇒ *Ganze_Zahl*

Gibt das Einerkomplement einer reellen ganzen Zahl zurück. Intern wird *Ganze_Zahl1* in eine 32-Bit-Dualzahl mit Vorzeichen umgewandelt. Für das Einerkomplement werden die Werte aller Bits umgekehrt (so dass 0 zu 1 wird und umgekehrt). Die Ergebnisse werden gemäß dem Base-Modus angezeigt.

Sie können die ganzen Zahlen mit jeder Basis eingeben. Für eine binäre oder hexadezimale Eingabe muß das Präfix 0b bzw. 0h verwendet werden. Ohne Präfix wird die ganze Zahl als dezimal behandelt (Grundzahl 10).

Geben Sie eine dezimale ganze Zahl ein, die für eine 32-Bit-Dualform mit Vorzeichen zu groß ist, dann wird eine symmetrische Modulo-Operation ausgeführt, um den Wert in den erforderlichen Bereich zu bringen.

Im Modus Hex base:

```
not 0h7AC36 ENTER                 0hFFF853C9
```

Wichtig: Null, nicht Buchstabe O.

Im Modus Bin base:

```
0b100101▸dec ENTER                37
not 0b100101 ENTER
  0b11111111111111111111111111011010
ans(1)▸dec ENTER                  -38
```

Hinweis: Eine binäre Eingabe kann bis zu 32 Stellen haben (das Präfix 0b wird nicht mitgezählt). Eine hexadezimale Eingabe kann bis zu 8 Stellen aufweisen.

Hinweis: Drücken Sie ▸ zur Eingabe des Umwandlungs-Operators 2nd [▸]. Sie können auch im Menü MATH/Base Basiskonversionen wählen.

nPr() MATH/Probability-Menü

nPr(*Term1***, *Term2***)** ⇒ *Term*

Für ganzzahlige *Term1* und *Term2* mit *Term1* ≥ *Term2* ≥ 0 ist **nPr()** die Anzahl der Möglichkeiten *Term2* Elemente unter Berücksichtigung der Reihenfolge aus *Term1* gegebenen Elementen auszuwählen. Beide Parameter können ganze Zahlen oder symbolische Terme sein.

```
nPr(z,3) ENTER                    z·(z-2)·(z-1)
ans(1)|z=5 ENTER                             60
nPr(z,-3) ENTER                    1
                                  ─────────────────
                                  (z+1)·(z+2)·(z+3)
nPr(z,c) ENTER                     z!
                                  ─────
                                  (z-c)!
```

nPr(*Term***, 0)** ⇒ 1

nPr(*Term, negGanze_Zahl***)** ⇒ 1/((*Term*+1) · (*Term*+2)... (*Term* – *negGanze_Zahl*))

nPr(*Term, posGanze_Zahl***)** ⇒ *Term* · (*Term* – 1)... (*Term* – *posGanze_Zahl*+1)

nPr(*Term, keineGanze_Zahl***)** ⇒ *Term*!/ (*Term* – *keineGanze_Zahl*)!

```
ans(1)*nPr(z-c,-c) ENTER                     1
```

---

nPr(*Liste1***, *Liste2***)** ⇒ *Liste*

Gibt eine Liste der Permutationen auf der Basis der entsprechenden Elementpaare der beiden Listen zurück. Die Parameter müssen Listen gleicher Größe sein.

```
nPr({5,4,3},{2,4,2}) ENTER
                                  {20 24 6}
```

---

nPr(*Matrix1***, *Matrix2***)** ⇒ *Matrix*

Gibt eine Matrix der Permutationen auf der Basis der entsprechenden Elementpaare der beiden Matrizen zurück. Die Parameter müssen Matrizen gleicher Größe sein.

```
nPr([6,5;4,3],[2,2;2,2]) ENTER
                                  [30  20]
                                  [12   6]
```

nSolve() MATH/Algebra-Menü

nSolve(*Gleichung***, *VarOderSchätzung***)** ⇒ *Zahl der Fehlermeldung*

Ermittelt iterativ eine reelle numerische Näherungslösung von *Gleichung* für deren eine Variable. *VarOderSchätzung* angeben als:

Variable
– oder –
Variable = reelle Zahl

Beispiel: x ist gültig, und x=3 ebenfalls.

nSolve() ist häufig sehr viel schneller als **solve()** oder **zeros()**, insbesondere, wenn zusätzlich der Operator "|" benutzt wird, um die Suche auf ein relativ kleines Intervall zu beschränken, das genau eine einzige Lösung enthält.

```
nSolve(x^2+5x-25=9,x) ENTER
                                      3.844...
nSolve(x^2=4,x=-1) ENTER                 -2.
nSolve(x^2=4,x=1) ENTER                   2.
```

Hinweis: Existieren mehrere Lösungen, können Sie mit Hilfe einer Schätzung eine bestimmte Lösung suchen.

```
nSolve(x^2+5x-25=9,x)|x<0 ENTER
                                     -8.844...
nSolve(((1+r)^24-1)/r=26,r)|r>
0 and r<.25 ENTER                      .0068...
```

nSolve() versucht entweder einen Punkt zu ermitteln, wo der Unterschied zwischen tatsächlichem und erwartetem Wert Null ist oder zwei relativ nahe Punkte, wo der Restfehler entgegengesetzte Vorzeichen besitzt und nicht zu groß ist. Wenn **nSolve()** dies nicht mit einer kleinen Anzahl von Versuchen erreichen kann, wird die Zeichenkette "no solution found" ("keine Lösung gefunden") zurückgegeben.

Wenn Sie **nSolve()** in einem Programm verwenden, können Sie mit **getType()** auf ein numerisches Ergebnis prüfen, bevor Sie es in einem algebraischen Term benutzen.

Hinweis: Siehe auch **cSolve()**, **cZeros()**, **solve()** und **zeros()**.

```
nSolve(x^2=-1,x) ENTER
                 "no solution found"
```

OneVar MATH/Statistics-Menü

OneVar *Liste1* [[, *Liste2*] [, *Liste3*] [, *Liste4*]]

Berechnet die 1-Variablenstatistik und aktualisiert alle Statistiksystemvariablen.

Alle Listen außer *Liste4* müssen die gleiche Dimension besitzen.

Liste1 stellt die Liste der x-Werte dar.
Liste2 stellt die Angaben für die Häufigkeit dar.
Liste3 stellt die Klassencodes dar.
Liste4 stellt die Klassenliste dar.

Hinweis: *Liste1* bis einschl. *Liste3* müssen Variablenamen oder c1–c99 sein (Spalten in der letzten Datenvariablen, die im Daten/ Matrix-Editor angezeigt wurde). *Liste4* braucht kein Variablenname zu sein und kann keine Spalte c1–c99 sein.

```
{0,2,3,4,3,4,6}→L1 ENTER
OneVar L1 ENTER                    Done
ShowStat ENTER
```

or MATH/Test-Menü

Boolescher Term1 **or** *Boolescher Term2* ⇒ *Boolescher Term*

Gibt "wahr" oder "falsch" oder eine vereinfachte Form der ursprünglichen Eingabe zurück.

Gibt "wahr" zurück, wenn ein Term oder beide Terme zu "wahr" ausgewertet werden. Gibt nur dann "falsch" zurück, wenn beide Terme "falsch" ergeben.

Hinweis: Siehe **xor**.

```
x≥3 or x≥4 ENTER                   x ≥ 3
```

Programmsegment:

```
  :
If x<0 or x≥5
  Goto END
  :
If choice=1 or choice=2
  Disp "Wrong choice"
  :
```

Ganze_Zahl1 **or** *Ganze_Zahl2* ⇒ *Ganze_Zahl*

Vergleicht zwei reelle ganze Zahlen mit Hilfe einer **or**-Operation Bit für Bit. Intern werden beide ganzen Zahlen in 32-Bit-Dualzahlen mit Vorzeichen konvertiert. Beim Vergleich der sich entsprechenden Bits ist das Ergebnis dann 1, wenn eines der Bits 1 ist; das Ergebnis ist nur dann 0, wenn beide Bits 0 sind. Der zurückgegebene Wert stellt die Bit-Ergebnisse dar und wird im jeweiligen Base-Modus angezeigt.

Sie können die ganzen Zahlen mit jeder Basis eingeben. Für eine binäre oder hexadezimale Eingabe muß das Präfix 0b bzw. 0h verwendet werden. Ohne Präfix werden die ganzen Zahlen als dezimal behandelt (Grundzahl 10).

Geben Sie eine dezimale ganze Zahl ein, die für eine 32-Bit-Dualform mit Vorzeichen zu groß ist, dann wird eine symmetrische Modulo-Operation ausgeführt, um den Wert in den erforderlichen Bereich zu bringen.

Hinweis: Siehe **xor**.

Im Modus Hex base:

```
0h7AC36 or 0h3D5F ENTER  0h7BD7F
```

Wichtig: Null, nicht Buchstabe O.

Im Modus Bin base:

```
0b100101 or 0b100 ENTER 0b100101
```

Hinweis: Eine binäre Eingabe kann bis zu 32 Stellen haben (das Präfix 0b wird nicht mitgezählt). Eine hexadezimale Eingabe kann bis zu 8 Stellen aufweisen.

ord() MATH/String-Menü

ord(*String***)** ⇒ *Ganze_Zahl*
ord(*Liste1***)** ⇒ *Liste*

Gibt den Zahlenwert (Code) des ersten Zeichens der Zeichenkette *String* zurück. Handelt es sich um eine Liste, wird der Code des ersten Zeichens jedes Listenelements zurückgegeben.

Eine vollständige Aufstellung der Zeichencodes finden Sie in Anhang B.

```
ord("Hello") ENTER                104
char(104) ENTER                   "h"
ord(char(24)) ENTER                24
ord({"alpha","beta"}) ENTER
                              {97  98}
```

Output CATALOG

Output *Zeile*, *Spalte*, *TermOderString*

Zeigt *TermOderString* (einen Term oder eine Zeichenkette) auf dem Programm-I/O-Bildschirm an den Textkoordinaten (*Zeile*, *Spalte*) an.

Ein Term kann Umwandlungsoperationen wie ▸**DD** und **Rect** enthalten. Sie können auch den Operator ▸ benutzen, um Einheiten- und Zahlensystem-Umwandlungen durchzuführen.

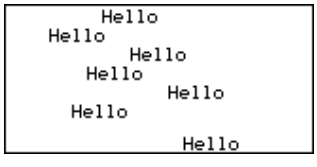
Ist Pretty Print = ON, wird *TermOderString* als "pretty print" angezeigt.

Im Programm- I/O-Bildschirm können Sie F5 drücken, um den Hauptbildschirm anzuzeigen; ein Programm kann **DispHome** verwenden.

Programmsegment:

```
   ⋮
:randseed(1147)
:ClrIO
:For i,1,100,10
:  Output i, rand(200),"Hello"
:EndFor
   ⋮
```

Ergebnis nach Ausführung:



P▸Rx() MATH/Angle-Menü

P▸Rx(*rTerm*, θ*Term*) ⇒ *Term*
P▸Rx(*rListe*, θ*Liste*) ⇒ *Liste*
P▸Rx(*rMatrix*, θ*Matrix*) ⇒ *Matrix*

Gibt die äquivalente x-Koordinate des Paars (r, θ) zurück.

Hinweis: Der θ-Parameter wird gemäß der aktuellen Winkel-Moduseinstellung als Ordnung oder Radiant interpretiert. Ist der Parameter ein Term, können Sie ° oder ʳ benutzen, um die Winkel-Moduseinstellung temporär zu ändern.

Im Radian-Modus:

P▸Rx(r,θ) ENTER → cos(θ)·r

P▸Rx(4,60°) ENTER → 2

P▸Rx({-3,10,1.3},{π/3,-π/4,0}) ENTER → $\{-3/2 \quad 5\cdot\sqrt{2} \quad 1.3\}$

P▸Ry() MATH/Angle-Menü

P▸Ry(*rTerm*, θ*Term*) ⇒ *Term*
P▸Ry(*rListe*, θ*List*) ⇒ *Liste*
P▸Ry(*rMatrix*, θ*Matrix*) ⇒ *Matrix*

Gibt die äquivalente y-Koordinate des Paars (r, θ) zurück.

Hinweis: Der θ-Parameter wird gemäß der aktuellen Winkel-Moduseinstellung als Ordnung oder Radiant interpretiert. Ist der Parameter ein Term, können Sie ° oder ʳ benutzen, um die Winkel-Moduseinstellung temporär zu ändern.

Im Radian-Modus:

P▸Ry(r,θ) ENTER → sin(θ)·r

P▸Ry(4,60°) ENTER → $2\cdot\sqrt{3}$

P▸Ry({-3,10,1.3},{π/3,-π/4,0}) ENTER → $\left\{\frac{-3\cdot\sqrt{3}}{2} \quad -5\cdot\sqrt{2} \quad 0.\right\}$

part() CATALOG

part(*Term1*[,*nicht_negative_Ganzzahl*]**)**

Mit dieser erweiterten Programmierfunktion können Sie alle Unterterme im vereinfachten Ergebnis von *Term1* bestimmen und extrahieren.

Kann beispielsweise *Term1* zu cos(π∗ x+3) vereinfacht werden:

- Hat die Funktion **cos()** ein Argument: (π∗ x+3).
- Hat die Summe von (π∗ x+3) zwei Operanden: π∗ x und 3.
- Hat die Zahl 3 weder Argumente noch Operanden.
- Hat das Produkt π∗ x zwei Operanden: π und x.
- Haben die Variable x und die symbolische Konstante π weder Argumente noch Operanden.

Besitzt x einen numerischen Wert und Sie drücken ◆ ENTER, so wird der numerische Wert von π∗ x berechnet, das Ergebnis zu 3 addiert und dann der Cosinus berechnet. Da **cos() zuletzt** angewendet wird, ist er der **Top-level**-Operator.

---

part(*Term1***)** ⇒ *Ganze_Zahl*

Vereinfacht *Term1* und gibt die Anzahl an Top-level-Argumenten oder -Operanden zurück. Hierbei wird 0 zurückgegeben, wenn *Term1* eine Zahl, Variable oder eine symbolische Konstante wie z.B. π, *e*, ***i*** oder ∞ ist.

part(cos(π∗ x+3)) ENTER → 1

Hinweis: cos(π∗ x+3) hat ein Argument.

part(*Term1***, 0)** ⇒ *String*

Vereinfacht *Term1* und gibt eine Zeichenfolge zurück, welche den Top-level-Funktionsnamen oder -Operator enthält. Hierbei wird **string(***Term1***)** zurückgegeben, wenn *Term1* eine Zahl, Variable oder eine symbolische Konstante wie z.B. π, *e*, ***i*** oder ∞ ist.

```
part(cos(π*x+3),0) ENTER    "cos"
```

part(*Term1***, *n*)** ⇒ *Term*

Vereinfacht *Term1* und gibt das *n*-te Argument bzw. den Operand zurück, wobei *n* > 0 ist und ≤ die Anzahl an Top-level-Argumenten oder -Operanden, die durch **part(***Term1***)** zurückgegeben werden. Anderenfalls wird ein Fehler angezeigt.

```
part(cos(π*x+3),1) ENTER    3+π·x
```

Hinweis: Durch die Vereinfachung wurde die Argumentenreihenfolge verändert.

Durch Kombination der Variationen von **part()** können Sie alle Unterterme im vereinfachten Ergebnis von *Term1* extrahieren. Wie aus nebenstehendem Beispiel hervorgeht, können Sie ein Argument oder einen Operanden speichern und dann weitere Unterterme mit **part()** extrahieren.

Hinweis: Verlassen Sie sich bei der Verwendung von **part()** nicht auf die Reihenfolge in Summen und Produkten.

```
part(cos(π*x+3)) ENTER                1
part(cos(π*x+3),0) ENTER          "cos"
part(cos(π*x+3),1)→temp ENTER
                                  3+π·x
temp ENTER                        π·x+3
part(temp,0) ENTER                  "+"
part(temp) ENTER                      2
part(temp,2) ENTER                    3
part(temp,1)→temp ENTER             π·x
part(temp,0) ENTER                  "*"
part(temp) ENTER                      2
part(temp,1) ENTER                    π
part(temp,2) ENTER                    x
```

Terme wie (x+y+z) und (x−y−z) werden intern als (x+y)+z und (x−y)−z dargestellt. Dies wirkt sich auf die für das erste und zweite Argument zurückgegebenen Werte aus. Aus technischen Gründen gibt **part(**x+y+z,1**)** y+x anstelle von x+y zurück.

```
part(x+y+z) ENTER                     2
part(x+y+z,2) ENTER                   z
part(x+y+z,1) ENTER                 y+x
```

Ebenso wird x*y*z intern als (x*y)*z dargestellt. Auch in diesem Fall wird das erste Argument aus technischen Gründen als y·x anstatt als x·y zurückgegeben.

```
part(x*y*z) ENTER                     2
part(x*y*z,2) ENTER                   z
part(x*y*z,1) ENTER                 y·x
```

Beachten Sie beim Extrahieren von Untertermen aus einer Matrix, daß Matrizen, wie in nebenstehendem Beispiel verdeutlicht, als Listen von Listen gespeichert werden.

```
part([a,b,c;x,y,z],0) ENTER         "{"
part([a,b,c;x,y,z]) ENTER             2
part([a,b,c;x,y,z],2)→temp
ENTER
                                {x  y  z}
part(temp,0) ENTER                  "{"
part(temp) ENTER                      3
part(temp,3) ENTER                    z
delVar temp ENTER                  Done
```

In der nebenstehenden als Beispiel aufgeführten Programm-Editor-Funktion werden zum teilweisen Einsatz symbolischer Differentiation **getType()** und **part()** verwendet. Eine eingehende Betrachtung der Funktion und ihre Vervollständigung kann dabei behilflich sein, das manuelle Differenzieren zu erlernen. Sie könnten sogar Funktionen wie Bessel-Funktionen angeben, die der TI-89 Titanium / Voyage™ 200 nicht differenzieren kann.

```
:d(y,x)
:Func
:Local f
:If getType(y)="VAR"
:  Return when(y=x,1,0,0)
:If part(y)=0
:  Return 0 ◙ y=π,∞,i,numbers
:part(y,0)→f
:If f="-" ◙ if negate
:  Return -d(part(y,1),x)
:If f="–" ◙ if minus
:  Return d(part(y,1),x)
     -d(part(y,2),x)
:If f="+"
:  Return d(part(y,1),x)
     +d(part(y,2),x)
:If f="*"
:  Return
part(y,1)*d(part(y,2),x)
     +part(y,2)*d(part(y,1),x)
:If f="{"
:  Return seq(d(part(y,k),x),
     k,1,part(y))
:Return undef
:EndFunc
```

PassErr CATALOG

PassErr

Gibt einen Fehler an die nächste Programmebene weiter.

Ist "errornum" gleich 0, bleibt **PassErr** ohne Wirkung.

In der Klausel **Else** des Programms sollte **ClrErr** oder **PassErr** benutzt werden. Verwenden Sie **ClrErr**, wenn der Fehler verarbeitet oder ignoriert werden soll. Ist unbekannt, wie der Fehler behandelt werden soll, übergeben Sie ihn mit **PassErr** an die nächste Fehlerbehandlungsroutine. (Siehe auch **ClrErr**).

Siehe **ClrErr** Programmliste

Pause CATALOG

Pause [*Term*]

Hält die Programmausführung vorübergehend an. Wenn Sie *Term* angeben, wird *Term* auf dem Programm-I/O-Bildschirm angezeigt.

Term kann Umwandlungsoperationen wie ▸**DD** und ▸**Rect** enthalten. Sie können auch den Operator ▸ benutzen, um Einheiten- und Zahlensystem-Umwandlungen durchzuführen.

Ergebnisse von *Term*, die für die Anzeige auf dem Bildschirm zu groß sind, können Sie mit dem Cursurfeld scrollen.

Die Programmausführung wird fortgesetzt, wenn Sie ENTER drücken.

Program segment:

```
  ⋮
:ClrIO
:DelVar temp
:1→temp[1]
:1→temp[2]
:Disp temp[2]
:◙ Guess the Pattern
:For i,3,20
:  temp[i-2]+temp[i-1]→temp[i]
:  Disp temp[i]
:  Disp temp,"Can you guess
the
     next","number?"
:  Pause
:EndFor
  ⋮
```

PlotsOff CATALOG

PlotsOff [**1**] [**, 2**] [**, 3**] **...** [**, 9**]

Schaltet die angegebenen Plots für die graphische Darstellung aus. Im 2-Graphikmodus wirkt sich dies nur auf den aktiven Graph aus.

Werden keine Parameter angegeben, werden alle Plots ausgeschaltet.

```
PlotsOff 1,2,5 ENTER          Done
PlotsOff ENTER                Done
```

PlotsOn CATALOG

PlotsOn [**1**] [**, 2**] [**, 3**] **...** [**, 9**]

Schaltet die angegebenen Plots für die graphische Darstellung ein. Im 2-Graph-Modus wirkt sich dies nur auf den aktiven Graph aus.

Wird kein Parameter angegeben, werden alle Plots eingeschaltet.

```
PlotsOn 2,4,5 ENTER           Done
PlotsOn ENTER                 Done
```

▸Polar MATH/Matrix/Vector ops-Menü

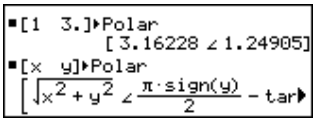
Vektor **▸Polar**

Zeigt *Vektor* in der Polar-Form [r ∠θ] an. Der Vektor muß die Dimension 2 besitzen und kann eine Zeile oder eine Spalte sein.

Hinweis: **▸Polar** ist eine Anzeigeformatanweisung, keine Konvertierungsfunktion. Sie können sie nur am Ende einer Eingabezeile benutzen, und sie nimmt keine Aktualisierung von ans vor.

Hinweis: Siehe auch **▸Rect**.

```
[1,3.]▸Polar ENTER
[x,y]▸Polar ENTER
```



---

komplexer_Wert **▸Polar**

Zeigt *komplexer_Vektor* in Polar-Form an.

- Der Degree-Modus für Winkel gibt (r∠θ) zurück.
- Der Radian-Modus für Winkel gibt $re^{i\theta}$ zurück.

Komplexer_Wert kann jede komplexe Form haben. Eine $re^{i\theta}$-Eingabe verursacht jedoch im Winkelmodus Degree einen Fehler.

Hinweis: Für eine Eingabe in Polar-Form müssen Klammern (r∠θ) verwendet werden.

Im Radian-Modus:

```
3+4i▸Polar ENTER      e^(i·(π/2 − tan⁻¹(3/4)))·5
(4∠π/3)▸Polar ENTER   e^(i·π/3)·4
```

Im Winkelmodus Degree:

```
3+4i▸Polar ENTER (5∠90−tan⁻¹(3/4))
```

polyEval() MATH/List-Menü

polyEval(*Liste1***,** *Term1***)** ⇒ *Term*
polyEval(*Liste1***,** *Liste2***)** ⇒ *Term*

Interpretiert den ersten Parameter als Koeffizienten eines nach fallenden Potenzen geordneten Polynoms und gibt das Polynom bezüglich des zweiten Parameters zurück.

```
polyEval({a,b,c},x) ENTER
a·x²+b·x+c
polyEval({1,2,3,4},2) ENTER      26
polyEval({1,2,3,4},{2,-7})
ENTER                        {26  -262}
```

PopUp CATALOG

PopUp *Menüpunktliste*, *Var*

Zeigt ein Popup-Menü an, das die Zeichenketten aus *Menüpunktliste* enthält, wartet, bis Sie einen Menüpunkt ausgewählt haben und speichert die Nummer der Auswahl in *Var*.

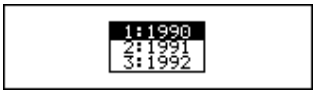
Die Elemente von *Menüpunktliste* müssen Zeichenketten sein: {*MenüPunkt1String*, *MenüPunkt2String*, *MenüPunkt3String*, ...}

Wenn *Var* bereits existiert und eine gültige Menüpunktnummer in ihr gespeichert ist, wird dieser Menüpunkt als Auswahlvorgabe angezeigt.

Menüpunktliste muß mindestens eine Auswahlmöglichkeit (einen Menüpunkt) enthalten.

```
PopUp
{"1990","1991","1992"},var1
```

[ENTER]



PowerReg MATH/Statistics/Regressions-Menü

PowerReg *Liste1*, *Liste2*[, [*Liste3*] [, *Liste4*, *Liste5*]]

Berechnet die Potenz-Regression und aktualisiert alle Statistik-Systemvariablen.

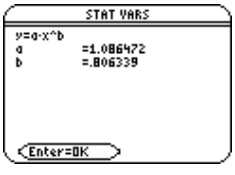
Alle Listen außer *Liste5* müssen die gleiche Dimension besitzen.

Liste1 stellt die Liste der x-Werte dar.
Liste2 stellt die Liste der y-Werte dar.
Liste3 stellt die Angaben für die Häufigkeit dar.
Liste4 stellt die Klassencodes dar.
Liste5 stellt die Klassenliste dar.

Hinweis: *Liste1* bis einschl. *Liste4* müssen Variablenamen oder c1–c99 sein (Spalten in der letzten Datenvariablen, die im Daten/ Matrix-Editor angezeigt wurde). *Liste5* braucht kein Variablenname zu sein und kann keine Spalte c1–c99 sein.

Im Funktions-Graphikmodus:

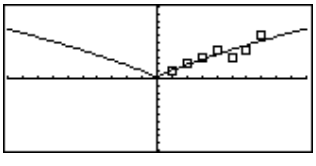
```
{1,2,3,4,5,6,7}→L1 [ENTER]
                          {1 2 3 ...}
{1,2,3,4,3,4,6}→L2 [ENTER]
                          {1 2 3 ...}
PowerReg L1,L2 [ENTER]          Done
ShowStat [ENTER]
```



[ENTER]

```
Regeq(x)→y1(x) [ENTER]          Done
NewPlot 1,1,L1,L2 [ENTER]       Done
```

[◆] [GRAPH]



Prgm CATALOG

Prgm
⋮
EndPrgm

Anweisung zum Identifizieren des Beginns eines Programms. Diese Anweisung ist zwingend erforderlich. Die letzte Programmzeile muß **EndPrgm** sein.

Programmsegment:

```
:prgmname()
:Prgm
:
:EndPrgm
```

product() MATH/List-Menü

product(*list*[, *start*[, *end*]]**)** ⇒ *expression*

Gibt das Produkt der Elemente von *Liste* zurück.

```
product({1,2,3,4}) [ENTER]          24
product({2,x,y}) [ENTER]          2·x·y
product({4,5,8,9},2,3) [ENTER]      40
```

product(*matrix1*[, *start*[, *end*]]) ⇒ *matrix*

Gibt einen Zeilenvektor zurück, der die Produkte der Elemente aus den Spalten von *Matrix1* enthält. *Start* und *end* sind nicht unbedingt erforderlich; sie dienen zur Angabe eines Reihenbereichs.

```
product([1,2,3;4,5,6;7,8,9]) ENTER
                              [28 80 162]
product([1,2,3;4,5,6;7,8,9],
1,2) ENTER                    [4,10,18]
```

Produkt() Siehe Π(), Seite 990.

Prompt CATALOG

Prompt *Var1*[, *Var2*] [, *Var3*] **...**

Gibt auf dem Programm-I/O-Bildschirm eine Eingabeaufforderung für jede Variable der Parameterliste aus, wobei `var1?` als Eingabeaufforderung benutzt wird. Speichert den eingegebenen Term/Ausdruck in der entsprechenden Variablen.

Prompt muß mit mindestens einem Parameter benutzt werden.

Programmsegment:

```
  ⋮
Prompt A,B,C
  ⋮
EndPrgm
```

propFrac() MATH/Algebra-Menü

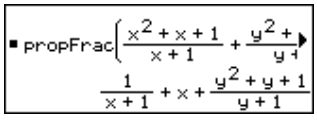
propFrac(*Term1*[, *Var*]**)** ⇒ *Term*

propFrac(*rationale_Zahl***)** gibt *rationale_Zahl* als Summe einer ganzen Zahl und eines Bruchs zurück, der das gleiche Vorzeichen besitzt und dessen Nenner größer ist als der Zähler.

```
propFrac(4/3) ENTER          1 + 1/3
propFrac(⁻4/3) ENTER         ⁻1 - 1/3
```

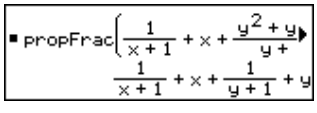
propFrac(*rationaler_Term,Var***)** gibt die Summe der echten Brüche und ein Polynom bezüglich *Var* zurück. Der Ordnung von *Var* im Nenner übersteigt in jedem echten Bruch den Ordnung von *Var* im Zähler. Gleichartige Potenzen von *Var* werden zusammengefaßt. Die Terme und ihre Faktoren werden nach *Var* als der Hauptvariablen sortiert.

```
propFrac((x^2+x+1)/(x+1)+
  (y^2+y+1)/(y+1),x) ENTER
```



Wird *Var* weggelassen, wird eine Entwicklung des echten Bruchs bezüglich der wichtigsten Hauptvariablen vorgenommen. Die Koeffizienten des Polynomteils werden dann zuerst bezüglich der wichtigsten Hauptvariablen entwickelt usw.

```
propFrac(ans(1))
```



Für rationale Terme ist **propFrac()** eine schnellere, aber weniger extreme Alternative zu **expand()**.

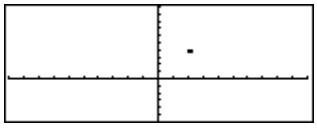
PtChg CATALOG

PtChg *x*, *y*
PtChg *xListe*, *yListe*

Öffnet den Graphikbildschirm und invertiert das Pixel, das den Fensterkoordinaten *x*, *y* am nächsten liegt.

Hinweis: Für die Einträge **PtChg** bis **PtText** werden ähnliche fortlaufende Beispiele benutzt.

```
PtChg 2,4 ENTER
```

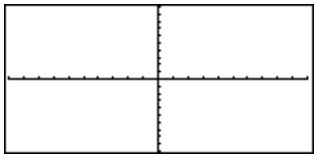


PtOff CATALOG

PtOff *x*, *y*
PtOff *xListe*, *yListe*

Öffnet den Graphikbildschirm und schaltet das Pixel aus, das den Fensterkoordinaten *x*, *y* am nächsten liegt.

```
PtOff 2,4 ENTER
```

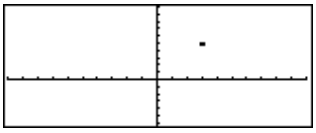


PtOn CATALOG

PtOn *x*, *y*
PtOn *xListe*, *yListe*

Öffnet den Graphikbildschirm und schaltet das Pixel ein, das den Fensterkoordinaten *x*, *y* am nächsten liegt.

```
PtOn 3,5 ENTER
```



ptTest( ) CATALOG

ptTest (*x*, *y***)** ⇒ *Boolescher konstanter Term*
ptTest (*xListe*, *yListe***)** ⇒ *Boolescher konstanter Term*

Gibt "wahr" oder "falsch" zurück. Gibt nur dann "wahr" zurück, wenn das Bildschirmpixel, das den Fensterkoordinaten *x*, *y* am nächsten liegt, eingeschaltet ist.

```
ptTest(3,5) ENTER                    true
```

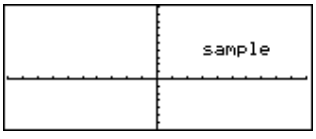
PtText CATALOG

PtText *String*, *x*, *y*

Öffnet den Graphikbildschirm und blendet die Zeichenkette *String* ab dem Pixel ein, das den angegebenen Fensterkoordinaten *x, y* am nächsten liegt.

Die oberste linke Ecke des ersten Zeichens von *String* wird an den Koordinaten plaziert.

```
PtText "sample",3,5 ENTER
```



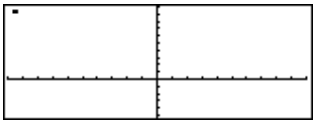
PxlChg CATALOG

PxlChg *Zeile*, *Spalte*
PxlChg *Zeilenliste*, *Spaltenliste*

Öffnet den Graphikbildschirm und invertiert das Pixel an den Pixel-Koordinaten *Zeile*, *Spalte*.

Hinweis: Beim Neuzeichnen werden alle gezeichneten Elemente gelöscht.

```
PxlChg 2,4 ENTER
```



PxlCrcl CATALOG

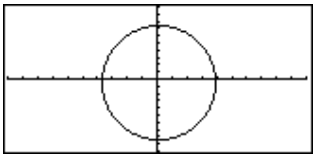
PxlCrcl *Zeile*, *Spalte*, *r* [, *Zeichenmodus*]

Öffnet den Graphikbildschirm und zeichnet einen Kreis mit dem Mittelpunkt an den Pixel-Koordinaten *Zeile*, *Spalte* mit dem Radius *r* Pixel.

Bei *Zeichenmodus*= 1 wird der Kreis gezeichnet (Vorgabe).
Bei *Zeichenmodus*= 0 wird der Kreis gelöscht.
Bei *Zeichenmodus*= -1 werden die Pixel entlang des Kreises invertiert.

Hinweis: Beim Neuzeichnen werden alle gezeichneten Elemente gelöscht. Siehe auch **Circle**.

```
PxlCrcl 40,80,30,1 ENTER
PxlCrcl 50,125,40,1
ENTER
```



PxlHorz CATALOG

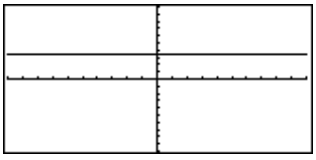
PxlHorz *Zeile* [**,** *Zeichenmodus*]

Öffnet den Graphikbildschirm und zeichnet eine horizontale Gerade durch die Pixel-Position *Zeile*.

Wenn *Zeichenmodus* = 1, wird die Gerade gezeichnet (Vorgabe).
Wenn *Zeichenmodus* = 0, wird die Gerade gelöscht.
Wenn *Zeichenmodus* = ⁻1, wird die Gerade invertiert (d. h. auf "ein" gesetzt, wenn sie "aus" ist und umgekehrt).

Hinweis: Beim Neuzeichnen werden alle gezeichneten Elemente gelöscht. Siehe auch **LineHorz**.

```
PxlHorz 25,1 ENTER
```



PxlLine CATALOG

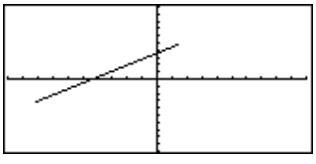
PxlLine *ZeileStart***,** *SpalteStart***,** *ZeileEnd***,** *SpalteEnd* [**,** *Zeichenmodus*]

Öffnet den Graphikbildschirm und zeichnet eine Strecke zwischen den Pixel-Koordinaten *ZeileStart*, *SpalteStart* und *ZeileEnd*, *SpalteEnd*, wobei beide Endpunkte eingeschlossen sind.

Wenn *Zeichenmodus* = 1, wird die Strecke gezeichnet (Vorgabe).
Wenn *Zeichenmodus* = 0, wird die Strecke gelöscht.
Wenn *Zeichenmodus* = ⁻1, wird die Strecke invertiert (d. h. auf “ein” gesetzt, wenn sie "aus" ist und umgekehrt).

Hinweis: Beim Neuzeichnen werden alle gezeichneten Elemente gelöscht. Siehe auch **Line**.

```
PxlLine 50,15,20,90,1 ENTER
PxlLine 80,20,30,150,1
ENTER
```



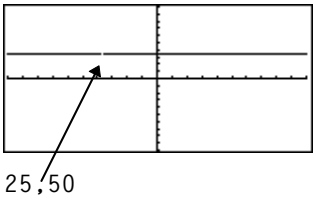
PxlOff CATALOG

PxlOff *Zeile***,** *Spalte*
PxlOff *Zeilenliste***,** *Spaltenliste*

Öffnet den Graphikbildschirm und schaltet das Pixel an den Pixel-Koordinaten *Zeile*, *Spalte* aus.

Hinweis: Beim Neuzeichnen werden alle gezeichneten Elemente gelöscht.

```
PxlHorz 25,1 ENTER
PxlOff 25,50 ENTER
```



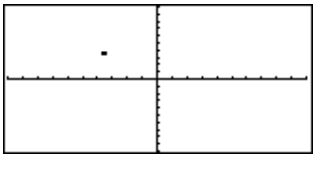
PxlOn CATALOG

PxlOn *Zeile***,** *Spalte*
PxlOn *Zeilenliste***,** *Spaltenliste*

Öffnet den Graphikbildschirm und schaltet das Pixel an den Pixel-Koordinaten *Zeile*, *Spalte* ein.

Hinweis: Beim Neuzeichnen werden alle gezeichneten Elemente gelöscht.

```
PxlOn 25,50 ENTER
```



pxlTest() CATALOG

pxlTest (*Zeile*, *Spalte***)** ⇒ *Boolescher Term*
pxlTest (*Zeilenliste*, *Spaltenliste***)** ⇒ *Boolescher Term*

Gibt "wahr" zurück, wenn das Pixel an den Pixel-Koordinaten *Zeile*, *Spalte* "ein" ist. Gibt "falsch" zurück, wenn das Pixel "aus" ist.

Hinweis: Beim Neuzeichnen werden alle gezeichneten Elemente gelöscht.

```
PxlOn 25,50 ENTER
```

[HOME]

[◆] [CALC HOME]

```
PxlTest(25,50) ENTER          true
PxlOff 25,50 ENTER
```

[HOME]

[◆] [CALC HOME]

```
PxlTest(25,50) ENTER          false
```

PxlText CATALOG

PxlText *String*, *Zeile*, *Spalte*

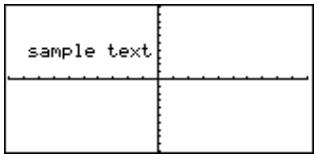
Öffnet den Graphikbildschirm und blendet die Zeichenkette *String* ab den Pixel-Koordinaten *Zeile, Spalte* ein.

Die oberste linke Ecke des ersten Zeichens von *String* wird an den Koordinaten plaziert.

Hinweis: Beim Neuzeichnen werden alle gezeichneten Elemente gelöscht.

```
PxlText
"Beispieltext",20,10 ENTER
```

```
PxlText
"Beispieltext",20,50 ENTER
```



PxlVert CATALOG

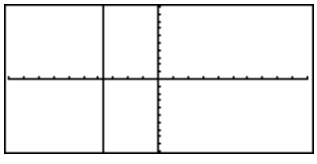
PxlVert *Spalte* [, *Zeichenmodus*]

Zeichnet eine vertikale Gerade durch die Pixel-Position *Spalte*.

Wenn *Zeichenmodus* = 1, wird die Gerade gezeichnet (Vorgabe).
Wenn *Zeichenmodus* = 0, wird die Gerade gelöscht.
Wenn *Zeichenmodus* = ⁻1, wird die Gerade invertiert (d. h. auf "ein" gesetzt, wenn sie "aus" ist und umgekehrt).

Hinweis: Beim Neuzeichnen werden alle gezeichneten Elemente gelöscht. Siehe auch **LineVert**.

```
PxlVert 50,1 ENTER
```



QR MATH/Matrix-Menü

QR *Matrix*, *qMatName*, *rMatName*[, *Tol*]

Berechnet die Householdersche QR-Faktorisierung einer reellen oder komplexen *Matrix*. Die sich ergebenden Q- und R-Matrzen werden in den angegebenen *MatNamen* gespeichert. Die Q-Matrix ist unitär. Bei der R-Matrix handelt es sich um eine obere Dreiecksmatrix.

Sie haben die Option, daß jedes Matrixelement als Null behandelt wird, wenn dessen absoluter Wert geringer als *Tol* ist. Diese Toleranz wird nur dann verwendet, wenn die Matrix Fließkommaelemente aufweist und keinerlei symbolische Variablen ohne zugewiesene Werte enthält. Anderenfalls wird *Tol* ignoriert.

- Wenn Sie ◆ ENTER verwenden oder den Modus auf Exact/Approx=APPROXIMATE einstellen, werden Berechnungen im Fließkomma-Rechenmodus durchgeführt.
- Wird *Tol* weggelassen oder nicht verwendet, so wird die Standardtoleranz folgendermaßen berechnet:

  5E⁻14 ∗ **max(dim(***Matrix***))** ∗**rowNorm(***Matrix***)**

  Die QR-Faktorisierung wird anhand von Householderschen Transformationen numerisch berechnet. Die symbolische Lösung wird mit dem Gram-Schmidt-Verfahren berechnet. Die Spalten in *qMatName* sind die orthonormalen Basisvektoren, die den durch *Matrix* definierten Raum aufspannen.

Die Fließkommazahl (9,) in m1 bewirkt, daß das Ergebnis in Fließkommaform berechnet wird.

```
[1,2,3;4,5,6;7,8,9.] →m1 ENTER
```

$$\begin{bmatrix}1 & 2 & 3\\ 4 & 5 & 6\\ 7 & 8 & 9.\end{bmatrix}$$

```
QR m1,qm,rm ENTER                Done
```

```
qm ENTER
```

$$\begin{bmatrix}.123\ldots & .904\ldots & .408\ldots\\ .492\ldots & .301\ldots & -.816\ldots\\ .861\ldots & -.301\ldots & .408\ldots\end{bmatrix}$$

```
rm ENTER
```

$$\begin{bmatrix}8.124\ldots & 9.601\ldots & 11.078\ldots\\ 0. & .904\ldots & 1.809\ldots\\ 0. & 0. & 0.\end{bmatrix}$$

```
[m,n;o,p] →m1 ENTER
```

$$\begin{bmatrix}m & n\\ o & p\end{bmatrix}$$

```
QR m1,qm,rm ENTER                Done
qm ENTER
```

$$\begin{bmatrix}\frac{m}{\sqrt{m^2+o^2}} & \frac{-sign(m\cdot p-n\cdot o)\cdot o}{\sqrt{m^2+o^2}}\\ \frac{o}{\sqrt{m^2+o^2}} & \frac{m\cdot sign(m\cdot p-n\cdot o)}{\sqrt{m^2+o^2}}\end{bmatrix}$$

```
rm ENTER
```

$$\begin{bmatrix}\sqrt{m^2+o^2} & \frac{m\cdot n+o\cdot p}{\sqrt{m^2+o^2}}\\ 0 & \frac{|m\cdot p-n\cdot o|}{\sqrt{m^2+o^2}}\end{bmatrix}$$

QuadReg MATH/Statistics/Regressions-Menü

QuadReg *Liste1*, *Liste2*[, [*Liste3*] [, *Liste4*, *Liste5*]]

Berechnet die quadratische polynomische Regression und aktualisiert alle Statistik-Systemvariablen.

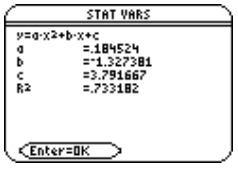
Alle Listen außer *Liste5* müssen die gleiche Dimension besitzen.

Liste1 stellt die Liste der x-Werte dar.
Liste2 stellt die Liste der y-Werte dar.
Liste3 stellt die Angaben für die Häufigkeit dar.
Liste4 stellt die Klassencodes dar.
Liste5 stellt die Klassenliste dar.

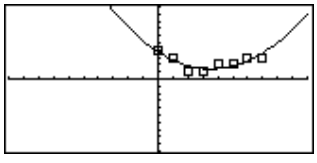
Hinweis: *Liste1* bis einschl. *Liste4* müssen Variablenamen oder c1–c99 sein (Spalten in der letzten Datenvariablen, die im Daten/ Matrix-Editor angezeigt wurde). *Liste5* braucht kein Variablenname zu sein und kann keine Spalte c1–c99 sein.

Im Funktions-Graphikmodus:

```
{0,1,2,3,4,5,6,7}→ L1 ENTER
                          {1 2 3 …}
{4,3,1,1,2,2,3,3}→ L2 ENTER
                          {4 3 1 …}
QuadReg L1,L2 ENTER              Done
ShowStat ENTER
```




```
ENTER
Regeq(x)→y1(x) ENTER             Done
NewPlot 1,1,L1,L2 ENTER          Done
◆ [GRAPH]
```



QuartReg MATH/Statistics/Regressions-Menü

QuartReg *Liste1*, *Liste2*[, [*Liste3*] [, *Liste4*, *Liste5*]]

Berechnet die polynomische Regression vierten Ordnunges und aktualisiert alle Statistik-Systemvariablen.

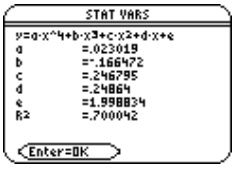
Alle Listen außer *Liste5* müssen die gleiche Dimension besitzen.

Liste1 stellt die Liste der x-Werte dar.
Liste2 stellt die Liste der y-Werte dar.
Liste3 stellt die Angaben für die Häufigkeit dar.
Liste4 stellt die Klassencodes dar.
Liste5 stellt die Klassenliste dar.

Hinweis: *Liste1* bis einschl. *Liste4* müssen Variablenamen oder c1–c99 sein (Spalten in der letzten Datenvariablen, die im Daten/ Matrix-Editor angezeigt wurde). *Liste5* braucht kein Variablenname zu sein und kann keine Spalte c1–c99 sein.

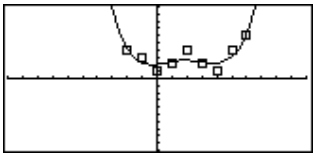
Im Funktions-Graphikmodus:

```
{-2,-1,0,1,2,3,4,5,6}→L1 ENTER
                          {-2 -1 0 ...}
{4,3,1,2,4,2,1,4,6}→L2 ENTER
                            {4 3 1 ...}
QuartReg L1,L2 ENTER          Done
ShowStat ENTER
```




```
ENTER
Regeq(x)→y1(x) ENTER          Done
NewPlot 1,1,L1,L2 ENTER       Done
```

◆ [GRAPH]



R▸Pθ() MATH/Angle-Menü

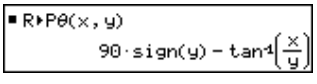
R▸Pθ (*xTerm*, *yTerm*) ⇒ *Term*
R▸Pθ (*xListe*, *yListe*) ⇒ *Liste*
R▸Pθ (*xMatrix*, *yMatrix*) ⇒ *Matrix*

Gibt die äquivalente θ-Koordinate des Parameterpaars (*x*, *y*) zurück.

Hinweis: Das Ergebnis wird entsprechend dem aktuellen Winkelmodus in Ordnung oder im Bogenmaß zurückgegeben.

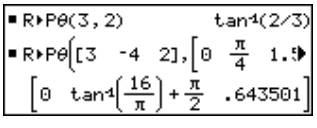
Im Degree-Modus für Winkel:

```
R▶Pθ(x,y) ENTER
```



Im Radian-Modus:

```
R▸Pθ(3,2) ENTER
R▸Pθ([3,-4,2],[0,π/4,1.5]) ENTER
```



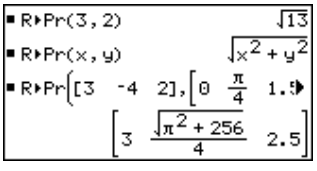
R▸Pr() MATH/Angle-Menü

R▸Pr (*xTerm*, *yTerm*) ⇒ *Term*
R▸Pr (*xListe*, *yListe*) ⇒ *Liste*
R▸Pr (*xMatrix*, *yMatrix*) ⇒ *Matrix*

Gibt die äquivalente r-Koordinate des Parameterpaars (*x*, *y*) zurück.

Im Radian-Modus:

```
R▸Pr(3,2) ENTER
R▸Pr(x,y) ENTER
R▸Pr([3,-4,2],[0,π/4,1.5]) ENTER
```



rand() MATH/Probability-Menü

rand(*n***)** ⇒ *Term*

n ist eine ganze Zahl ≠ Null.

Ohne Parameterangabe wird die nächste Zufallszahl einer Zufallszahlenreihe zwischen 0 und 1 zurückgegeben. Bei positivem Parameter wird eine zufällige ganze Zahl im Intervall [1, *n*] zurückgegeben. Bei negativem Parameter wird eine zufällige ganze Zahl im Intervall [− *n*, − 1] zurückgegeben.

```
RandSeed 1147 ENTER                    Done
```
(Setzt Ausgangsbasis für Zufallszahlengenerierung)

```
rand() ENTER                           .158...
rand(6) ENTER                          5
rand(-100) ENTER                       -49
```

randMat() MATH/Probability-Menü

randMat(*AnzahlZeilen***, *AnzahlSpalten***)** ⇒ *Matrix*

Gibt eine Matrix der angegebenen Dimension mit ganzzahligen Werten zwischen -9 und 9 zurück.

Beide Parameter müssen zu ganzen Zahlen vereinfachbar sein.

```
RandSeed 1147 ENTER                    Done
randMat(3,3) ENTER
```

$$\begin{bmatrix} 8 & -3 & 6 \\ -2 & 3 & -6 \\ 0 & 4 & -6 \end{bmatrix}$$

Hinweis: Die Werte in dieser Matrix ändern sich mit jedem Drücken von ENTER.

randNorm() MATH/Probability-Menü

randNorm(*Mittel***, *sd***)** ⇒ *Term*

Gibt eine Dezimalzahl aus der Gauß'schen Normalverteilung zurück. Dies könnte eine beliebige reelle Zahl sein, die Werte konzentrieren sich jedoch stark in dem Intervall [*Mittel*-3∗ *sd*, *Mittel*+3∗ *sd*].

```
RandSeed 1147 ENTER                    Done
randNorm(0,1) ENTER                    .492...
randNorm(3,4.5) ENTER                  -3.543...
```

randPoly() MATH/Probability-Menü

randPoly(*Var***, *Ordnung***)** ⇒ *Term*

Gibt ein Polynom in *Var* des angegebenen Ordnunges zurück. Die Koeffizienten sind zufällige ganze Zahlen im Bereich − 9 bis 9. Der führende Koeffizient ist nie Null.

Ordnung muß zwischen 0 und 99 betragen.

```
RandSeed 1147 ENTER                    Done
randPoly(x,5) ENTER
```

$$-2 \cdot x^5+3 \cdot x^4-6 \cdot x^3+4 \cdot x-6$$

RandSeed MATH/Probability-Menü

RandSeed *Zahl*

Zahl = 0 setzt die Ausgangsbasis ("seed") für den Zufallszahlengenerator auf die Werkseinstellung zurück. Bei *Zahl* ≠ 0 werden zwei Basen erzeugt, die in den Systemvariablen seed1 und seed2 gespeichert werden.

```
RandSeed 1147 ENTER                    Done
rand() ENTER                           .158...
```

RclGDB CATALOG

RclGDB *GDBVar*

Stellt alle Einstellungen wieder her, die in der Graphik-Einstellungs-Variablen GDBvar gespeichert sind.

Eine Liste der Einstellungen finden Sie bei **StoGDB**.

Hinweis: Die Wiederherstellung von Daten ist nur möglich, wenn sie in GDBvar gespeichert wurden.

```
RclGDB GDBvar ENTER                    Done
```

RclPic CATALOG

RclPic *picVar* [**,** *Zeile***,** *Spalte*]

Öffnet den Graphikbildschirm und überlagert den aktuellen Graphikbildschirm ab den Pixel-Koordinaten *Zeile, Spalte* in einer logischen ODER-Verknüpfung mit dem in *picVar* gespeicherten Bild.

picVar muß ein Bildtyp (PIC) sein.

Vorgabewert der Koordinaten ist (0,0).

real() MATH/Complex-Menü

real(*Term1***)** ⇒ *Term*

Gibt den Realteil des Parameters zurück.

Hinweis: Alle undefinierten Variablen werden als reelle Variablen behandelt. Siehe auch **imag()**.

```
real(2+3i) ENTER                    2
real(z) ENTER                       z
real(x+iy) ENTER                    x
```

---

real(*Liste1***)** ⇒ *Liste*

Gibt für jedes Element den Realteil zurück.

```
real({a+i* b,3,i}) ENTER   {a   3   0}
```

---

real(*Matrix1***)** ⇒ *Matrix*

Gibt für jedes Element den Realteil zurück.

```
real([a+i* b,3;c,i]) ENTER   [a  3]
                             [c  0]
```

▸Rect MATH/Matrix/Vector ops-Menü

Vektor **▸Rect**

Zeigt *Vektor* in der kartesischen Form [x, y, z] an. Der Vektor muß die Dimension 2 oder 3 besitzen und kann eine Zeile oder eine Spalte sein.

Hinweis: **▸Rect** ist eine Anzeigeformatanweisung, keine Konvertierungsfunktion. Sie können sie nur am Ende einer Eingabezeile benutzen, und sie nimmt keine Aktualisierung von ans vor.

Hinweis: Siehe auch **▸Polar**.

```
[3,∠π/4,∠π/6]▸Rect ENTER
                [3·√2/4  3·√2/4  3·√3/2]

[a,∠b,∠c] ENTER [a·cos(b)·sin(c)
                 a·sin(b)·sin(c)  a·cos(c)]
```

---

komplexer_Wert **▸Rect**

Zeigt *komplexer_Wert* in der kartesischen Form a+b*i* an. *Komplexer_Wert* kann jede komplexe Form haben. Eine $re^{i\theta}$-Eingabe verursacht jedoch im Winkelmodus Degree einen Fehler.

Hinweis: Für eine Eingabe in Polar-Form müssen Klammern (r∠θ) verwendet werden.

Im Radian-Modus:

```
44e^(π/3)▸Rect ENTER          4·e^(π/3)

(4∠π/3)▸Rect ENTER            2+2·√3·i
```

In Degree angle mode:

```
(4∠60)▸Rect ENTER             2+2·√3·i
```

Hinweis: Drücken Sie zur Eingabe von **▸Rect** mit der Tastatur 2nd [▸] für den Operator. Um ∠ einzugeben, drücken Sie 2nd [∠].

ref() MATH/Matrix-Menü

ref(*Matrix1*[, *Tol*]**)** ⇒ *Matrix*

Gibt die Zeilenstaffelform von *Matrix1* zurück.

Sie haben die Option, daß jedes Matrixelement als Null behandelt wird, wenn dessen absoluter Wert geringer als *Tol* ist. Diese Toleranz wird nur dann verwendet, wenn die Matrix Fließkommaelemente aufweist und keinerlei symbolische Variablen ohne zugewiesene Werte enthält. Anderenfalls wird *Tol* ignoriert.

- Wenn Sie ◆ ENTER verwenden oder den Modus auf Exact/Approx=APPROXIMATE einstellen, werden Berechnungen im Fließkomma-Rechenmodus durchgeführt.
- Wird *Tol* weggelassen oder nicht verwendet, so wird die Standardtoleranz folgendermaßen berechnet:

  5E⁻14 * **max(dim(***Matrix1***))** * **rowNorm(***Matrix1***)**

Hinweis: Siehe auch **rref()**.

```
ref([-2,-2,0,-6;1,-1,9,-9;-5,
2,4,-4]) ENTER
```

$$\begin{bmatrix} 1 & -2/5 & -4/5 & 4/5 \\ 0 & 1 & 4/7 & 11/7 \\ 0 & 0 & 1 & -62/71 \end{bmatrix}$$

```
[a,b,c;e,f,g]→m1 ENTER
```

$$\begin{bmatrix} a & b & c \\ e & f & g \end{bmatrix}$$

```
ref(m1) ENTER
```

$$\begin{bmatrix} 1 & \frac{f}{e} & \frac{g}{e} \\ 0 & 1 & \frac{a\cdot g - c\cdot e}{a\cdot f - b\cdot e} \end{bmatrix}$$

remain() MATH/Number-Menü

remain(*Term1***,** *Term2***)** ⇒ *Term*
remain(*Liste1***,** *Liste2***)** ⇒ *Liste*
remain(*Matrix1***,** *Matrix2***)** ⇒ *Matrix*

Gibt den Rest des ersten Parameters bezüglich des zweiten Parameters gemäß folgender Definitionen zurück:

remain(x,0) ≡ x
remain(x,y) ≡ x− y*iPart(x/y)

Als Folge daraus ist zu beachten, daß **remain(**− x,y**)** ≡ − **remain(**x,y**)**. Das Ergebnis ist entweder Null oder besitzt das gleiche Vorzeichen wie das erste Argument.

Hinweis: Siehe auch **mod()**.

```
remain(7,0) ENTER                    7
remain(7,3) ENTER                    1
remain(-7,3) ENTER                  -1
remain(7,-3) ENTER                   1
remain(-7,-3) ENTER                 -1
remain({12,-14,16},{9,7,-5})
ENTER
                              {3  0 1}
remain([9,-7;6,4],[4,3;4,-3])
ENTER
```

$$\begin{bmatrix} 1 & -1 \\ 2 & 1 \end{bmatrix}$$

Rename CATALOG

Rename *alterVarName***,** *neuerVarName*

Benennt die Variable *alterVarName* in *neuerVarName* um.

```
{1,2,3,4}→L1 ENTER          {1,2,3,4}
Rename L1, liste1 ENTER          Done
liste1 ENTER                {1,2,3,4}
```

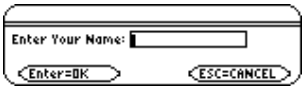
Request CATALOG

Request *AuffordString***,** *Var*

Befindet sich **Request** innerhalb einer Struktur **Dialog**...**EndDlog**, erzeugt es ein Eingabefeld, in das der Benutzer Daten eingibt. Bei Verwendung als eigenständige Anweisung erzeugt **Request** ein Dialogfeld für die Eingabe. Für beide Fälle gilt: enthält *Var* eine Zeichenkette, wird diese im Eingabefeld markiert als Auswahlvorgabe angezeigt. *AuffordString* muß ≤ 20 Zeichen sein.

Diese Anweisung kann eigenständig oder als Teil einer Dialogstruktur benutzt werden.

```
Request "Enter Your Name",str1
ENTER
```



Return CATALOG

Return [*Term***]**

Gibt *Term* als Ergebnis einer Funktion zurück. Verwendbar in einem Block **Func**...**EndFunc** oder einem Block **Prgm**...**EndPrgm**.

Hinweis: Zum Beenden eines Programms können Sie **Return** ohne Parameter benutzen.

Hinweis: Geben Sie den Text als eine zusammenhängende Zeile in Hauptbildschirm ein (ohne Zeilenumbrüche).

```
Define factoral(nn)=Func
 :local answer,count:1→answer
 :For count,1,nn
 :answer*count→answer:EndFor
 :Return answer:EndFunc ENTER Done

factoral(3) ENTER                6
```

right() MATH/List-Menü

right(*Liste1***[, *Anzahl*]) ⇒ *Liste***

Gibt *Anzahl* Elemente zurück, die rechts in *Liste1* enthalten sind.

Wenn Sie *Anzahl* weglassen, wird die gesamte *Liste1* zurückgegeben.

```
right({1,3,-2,4},3) ENTER
                              {3 -2 4}
```

---

right(*QuellString***[, *Anzahl*]) ⇒ *string***

Gibt *Anzahl* Zeichen zurück, die rechts in der Zeichenkette *QuellString* enthalten sind.

Wenn Sie *Anzahl* weglassen, wird der gesamte *QuellString* zurückgegeben.

```
right("Hello",2) ENTER          "lo"
```

---

right(*Vergleich***) ⇒ *Term***

Gibt die rechte Seite einer Gleichung oder Ungleichung zurück.

```
right(x<3) ENTER                 3
```

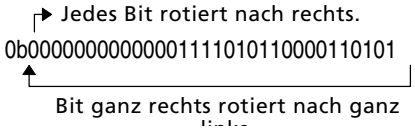
rotate() MATH/Base-Menü

rotate(*Ganze_Zahl1***[,*Anzahl_Rotationen*]) ⇒ *Ganze_Zahl***

Rotiert die Bits in einer binären ganzen Zahl. *Ganze_Zahl1* kann mit jeder Basis eingegeben werden und wird stets automatisch in eine 32-Bit-Dualform konvertiert. Ist der Absolutwert von *Ganze_Zahl1* für diese Form zu groß, so wird eine symmetrische Modulo-Operation ausgeführt, um sie in den erforderlichen Bereich zu bringen.

Ist *Anzahl_Rotationen* positiv, erfolgt eine Rotation nach links; ist *Anzahl_Rotationen* negativ, erfolgt eine Rotation nach rechts. Vorgabe ist -1 (ein Bit nach rechts rotieren).

Aus beispielsweise folgender Rechtsrotation:



ergibt sich:

0b10000000000000011110101100001101O

Das Ergebnis wird gemäß dem jeweiligen Base-Modus angezeigt.

Im Modus Bin base:

```
rotate(0b1111010110000110101)
ENTER
0b10000000000000011110101100001101O
rotate(256,1) ENTER 0b1000000000
```

Im Modus Hex base:

```
rotate(0h78E) ENTER          0h3C7
rotate(0h78E,-2) ENTER 0h800001E3
rotate(0h78E,2) ENTER        0h1E38
```

Wichtig: Geben Sie eine Dual- oder Hexadezimalzahl stets mit dem Präfix 0b bzw. 0h ein (Null, nicht der Buchstabe O).

rotate(*Liste1*[**,***Anzahl_Rotationen*]**)** ⇒ *Liste*

Gibt eine um *Anzahl_Rotationen* Elemente nach rechts oder links rotierte Kopie von *Liste1* zurück. Verändert *Liste1* nicht.

Ist *Anzahl_Rotationen* positiv, erfolgt eine Rotation nach links; ist *Anzahl_Rotationen* negativ, erfolgt eine Rotation nach rechts. Vorgabe ist −1 (ein Bit nach rechts rotieren).

Im Modus Dec base:

```
rotate({1,2,3,4}) ENTER
                        {4 1 2 3}
rotate({1,2,3,4},-2) ENTER
                        {3 4 1 2}
rotate({1,2,3,4},1) ENTER
                        {2 3 4 1}
```

rotate(*String1*[**,***Anzahl_Rotationen*]**)** ⇒ *String*

Gibt eine um *Anzahl_Rotationen* Zeichen nach rechts oder links rotierte Kopie von *String1* zurück. Verändert *String1* nicht.

Ist *Anzahl_Rotationen* positiv, erfolgt eine Rotation nach links; ist *Anzahl_Rotationen* negativ, erfolgt eine Rotation nach rechts. Vorgabe ist −1 (ein Bit nach rechts rotieren).

```
rotate("abcd") ENTER      "dabc"
rotate("abcd",-2) ENTER   "cdab"
rotate("abcd",1) ENTER    "bcda"
```

round() MATH/Number-Menü

round(*Term1*[**,** *Stellen*]**)** ⇒ *Term*

Gibt den Parameter gerundet auf die angegebene Anzahl von Stellen nach dem Dezimaltrennzeichen zurück.

Stellen muß eine ganze Zahl im Bereich 0–12 sein. Wird *Stellen* weggelassen, wird der Parameter auf 12 signifikante Stellen gerundet.

Hinweis: Die Anzeige des Ergebnisses kann von der Einstellung Anzeige Digits (Anzeigestellen) beeinflußt werden.

```
round(1.234567,3) ENTER     1.235
```

round(*Liste1*[**,** *Stellen*]**)** ⇒ *Liste*

Gibt eine Liste von Elementen zurück, die auf die angegebene Stellenzahl gerundet wurden.

```
round({π,√(2),ln(2)},4) ENTER
                  {3.1416 1.4142 .6931}
```

round(*Matrix1*[**,** *Stellen*]**)** ⇒ *Matrix*

Gibt eine Matrix von Elementen zurück, die auf die angegebene Stellenzahl gerundet wurden.

```
round([ln(5),ln(3);π,e^(1)],1)
ENTER
                          [1.6  1.1]
                          [3.1  2.7]
```

rowAdd() MATH/Matrix/Row ops-Menü

rowAdd(*Matrix1***,** *rIndex1***,** *rIndex2***)** ⇒ *Matrix*

Gibt eine Kopie von *Matrix1* zurück, in der die Zeile *rIndex2* durch die Summe der Zeilen *rIndex1* und *rIndex2* ersetzt ist.

```
rowAdd([3,4;-3,-2],1,2) ENTER
                          [3  4]
                          [0  2]
rowAdd([a,b;c,d],1,2) ENTER
                          [a    b  ]
                          [a+c  b+d]
```

rowDim() MATH/Matrix/Dimensions-Menü

rowDim(*Matrix***)** ⇒ *Term*

Gibt die Anzahl der Zeilen von *Matrix* zurück.

Hinweis: Siehe auch **colDim()**.

```
[1,2;3,4;5,6]→M1 ENTER      [1  2]
                            [3  4]
                            [5  6]
rowdim(M1) ENTER              3
```

rowNorm() MATH/Matrix/Norms-Menü

rowNorm(*Matrix***)** ⇒ *Term*

Gibt das Maximum der Summen der Absolutwerte der Elemente der Zeilen von *Matrix* zurück.

Hinweis: Alle Matrixelemente müssen zu Zahlen vereinfachbar sein. Siehe auch **colNorm()**.

```
rowNorm([-5,6,-7;3,4,9;9,-9,
-7]) ENTER                    25
```

rowSwap() MATH/Matrix/Row ops-Menü

rowSwap(*Matrix1***, *rIndex1*, *rIndex2*)** ⇒ *Matrix*

Gibt eine Matrix zurück, in der die Zeilen *rIndex1* und *rIndex2* von *Matrix1* vertauscht sind.

```
[1,2;3,4;5,6]→Mat ENTER
```

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$$

```
rowSwap(Mat,1,3) ENTER
```

$$\begin{bmatrix} 5 & 6 \\ 3 & 4 \\ 1 & 2 \end{bmatrix}$$

RplcPic CATALOG

RplcPic *picVar*[**,** *Zeile*][**,** *Spalte*]

Löscht den Graphikbildschirm und setzt das Bild *picVar* an den Pixel-Koordinaten *Zeile*, *Spalte* ein. Soll der Bildschirm nicht gelöscht werden, benutzen Sie **RclPic**.

picVar muß eine Variable des Datentyps “picture” (Bild) sein. *Zeile* und *Spalte* legen (sofern angegeben) die Pixel-Koordinaten der linken oberen Ecke des Bilds fest. Vorgegeben sind die Koordinaten (0, 0).

Hinweis: Bei Bildern, die nicht bildschirmfüllend sind, wird nur der Bereich gelöscht, der für das neue Bild benötigt wird.

rref() MATH/Matrix-Menü

rref(*Matrix1*[**,** *Tol*]**)** ⇒ *Matrix*

Gibt die reduzierte Zeilenstaffelform von *Matrix1* zurück.

Sie haben die Option, daß jedes Matrixelement als Null behandelt wird, wenn dessen absoluter Wert geringer als *Tol* ist. Diese Toleranz wird nur dann verwendet, wenn die Matrix Fließkommaeinträge aufweist und keinerlei symbolische Variablen ohne zugeteilte Werte enthält. Anderenfalls wird *Tol* ignoriert.

- Wenn Sie ◆ ENTER verwenden oder den Modus auf Exact/Approx=APPROXIMATE einstellen, werden Berechnungen im Fließkomma-Rechenmodus durchgeführt.
- Wird *Tol* weggelassen oder nicht verwendet, so wird die Standardtoleranz folgendermaßen berechnet:

  5E⁻14 * **max(dim(***Matrix1***))** * **rowNorm(***Matrix1***)**

Hinweis: Siehe auch **ref()**.

```
rref([-2,-2,0,-6;1,-1,9,-9;
-5,2,4,-4]) ENTER
```

$$\begin{bmatrix} 1 & 0 & 0 & 66/71 \\ 0 & 1 & 0 & \frac{147}{71} \\ 0 & 0 & 1 & -62/71 \end{bmatrix}$$

```
rref([a,b,x;c,d,y]) ENTER
```

$$\begin{bmatrix} 1 & 0 & \frac{d \cdot x - b \cdot y}{a \cdot d - b \cdot c} \\ 0 & 1 & \frac{-(c \cdot x - a \cdot y)}{a \cdot d - b \cdot c} \end{bmatrix}$$

sec() Menü MATH/Trig (MATHEMATIK/Trigonometrie)

sec(*Ausdruck1***)** ⇒ *Ausdruck*
sec(*Liste1***)** ⇒ *Liste*

Gibt den Sekans von *Ausdruck1* oder eine Liste der Sekans aller Elemente in *Liste1* zurück.

Hinweis: Das Argument wird entsprechend dem aktiven Winkelmodus als Winkel in Grad oder im Bogenmaß (Radian) interpretiert.

Im Grad-Modus:

sec(45) ENTER $\sqrt{(2)}$

sec({1,2.3,4}) ENTER

$\frac{1}{\cos(1)}$ 1.000… $\frac{1}{\cos(4)}$

$\sec^{-1}()$ Menü MATH/Trig (MATHEMATIK/Trigonometrie)

sec⁻¹(*Ausdruck1***)** ⇒ *Ausdruck*
sec⁻¹(*Liste1***)** ⇒ *Liste*

Gibt entweder den Winkel, dessen Sekans *Ausdruck1* entspricht, oder eine Liste der inversen Sekans aller Elemente in *Liste1* zurück.

Hinweis: Das Ergebnis wird entsprechend dem aktiven Winkelmodus als Winkel in Grad oder im Bogenmaß (Radian) interpretiert.

Im Grad-Modus:

$\sec^{-1}(1)$ ENTER 0

Im Radian-Modus:

$\sec^{-1}(\{1,2,5\})$ ENTER

0 $\frac{\pi}{3}$ $\cos^{-1}(1/5)$

sech() Menü MATH/Hyperbolic (MATHEMATIK/Hyperbolisch)

sech(*Ausdruck1***)** ⇒ *Ausdruck*
sech(*Liste1***)** ⇒ *Liste*

Gibt den hyperbolischen Sekans von *Ausdruck1* oder eine Liste der hyperbolischen Sekans aller Elemente in *Liste1* zurück.

sech(3) ENTER $\frac{1}{\cosh(3)}$

sech({1,2.3,4}) ENTER

$\frac{1}{\cosh(1)}$ .198… $\frac{1}{\cosh(4)}$

$\text{sech}^{-1}()$ Menü MATH/Hyperbolic (MATHEMATIK/Hyperbolisch)

sech⁻¹(*Ausdruck1***)** ⇒ *Ausdruck*
sech⁻¹(*Liste1***)** ⇒ *Liste*

Gibt den inversen hyperbolischen Sekans von *Ausdruck1* oder eine Liste der inversen hyperbolischen Sekans aller Elemente in *Liste1* zurück.

Im Radian-Modus und Komplex-Modus rechtwinklig:

$\text{sech}^{-1}(1)$ ENTER 0

$\text{sech}^{-1}(\{1,-2,2.1\})$ ENTER

0 $(\frac{2 \cdot \pi}{3}) \cdot i$ 1.074… $\cdot i$

Send CATALOG

Send *Liste*

CBL™-(Calculator-Based Laboratory™) oder CBR™-(Calculator-Based Ranger™) -Anweisung Sendet *Liste* zum Kommunikationsanschluß.

Programmsegment:

```
   ⋮
:Send {1,0}
:Send {1,2,1}
   ⋮
```

SendCalc CATALOG

SendCalc *Var*

Sendet die Variable *Var* zur Verbindungsschnittstelle, über die ein weiteres an die Schnittstelle angeschlossenes Gerät den Variablenwert empfangen kann. Auf dem empfangenden Gerät muß der Hauptbildschirm geöffnet sein oder **GetCalc** in einem Programm ausgeführt werden.

Wenn Sie von einem TI-89 Titanium, TI-92 Plus oder Voyage™ 200 an einen TI–92 senden und der TI–92 **GetCalc** in einem Programm ausführt, tritt ein Fehler auf. In diesem Fall muß auf dem sendenden Gerät **SendChat** verwendet werden.

SendCalc *Var[,port]*

Sendet den Wert von *var* von einem TI-89 Titanium zu einem anderen TI-89 Titanium.

Ist kein Anschluss angegeben oder ist *port = 0*, so sendet der TI-89 Titanium die Daten über den USB-Anschluss, sofern angeschlossen. Trifft dies nicht zu, werden die Daten über den E/A-Anschluss gesendet.

Bei *port = 1* sendet der TI-89 Titanium Daten ausschließlich über den USB-Anschluss.

Bei *port = 2* sendet der TI-89 Titanium Daten ausschließlich über den E/A-Anschluss.

Programmausschnitt:

```
   ⋮
:a+b→x
:SendCalc x
   ⋮
```

SendChat CATALOG

SendChat *Var*

Eine allgemeine Alternative zu **SendCalc**, die sich anbietet, wenn es sich beim empfangenden Gerät um einen TI–92 handelt (oder bei einem allgemeinen "Chat"-Programm, für welches entweder ein TI–92, TI-92 Plus oder ein Voyage 200 verwendet werden kann). Nähere Informationen finden Sie unter **SendCalc**.

SendChat sendet nur dann eine Variable, wenn diese Variable mit dem TI–92 kompatibel ist; dies ist typischerweise der Fall bei "Chat"-Programmen. **SendChat** sendet jedoch keine archivierten Variablen oder TI–89-Graphik-Einstellungen etc.

Programmausschnitt:

```
   ⋮
:a+b→x
:SendChat x
   ⋮
```

seq() MATH/List-Menü

seq(*Term*, *Var*, *unten*, *oben*[, *Schritt*]**)** ⇒ *Liste*

Erhöht die Variable *Var* in Stufen von *Schritt* ab dem Ausgangswert *unten* bis zum Wert *oben,* wertet *Term* aus und gibt die Ergebnisse als Liste zurück. Der ursprüngliche Inhalt von *Var* ist nach Beendigung von **seq()** weiterhin vorhanden.

Var darf keine Systemvariable sein.

Der Vorgabewert für *Schritt* ist 1.

```
seq(n^2,n,1,6) ENTER
                {1 4 9 16 25 36}

seq(1/n,n,1,10,2) ENTER
              {1 1/3 1/5 1/7 1/9}

sum(seq(1/n^2,n,1,10,1)) ENTER
                              196...
                              127...
```

oder drücken Sie ◆ ENTER,
um diesen Wert zu erhalten: 1.549...

setDate() CATALOG

setDate(*year,month,day***)** ⇒ *listold*

Stellt die Uhr auf das durch das Argument festgelegte Datum und gibt eine Liste zurück. (**Hinweis:** Das Jahr muss im Bereich 1997 – 2132 liegen.) Die zurückgegebene Liste hat das Format {*Altes Jahr,Alter Monat,Alter Tag*}. Das zurückgegebene Datum entspricht dem vorherigen Wert der Uhrzeit.

Geben Sie das Jahr als vierstelligen Integer an. Monat und Tag können ein- oder zweistellige Integer sein.

```
setDate(2001,10,31) ENTER
                    {2001  11  1}
```

setDtFmt() CATALOG

setDtFmt(*integer***)** ⇒ *integerold*

Stellt das Datumsformat für den Arbeitsplatz dem Argument entsprechend und gibt den Formatwert des vorher eingestellten Datumsformats zurück.

Integer-Werte:

```
1 = MM/TT/JJ
2 = TT/MM/JJ
3 = MM.TT.JJ
4 = TT.MM.JJ
5 = JJ.MM.TT
6 = MM-TT-JJ
7 = TT-MM-JJ
8 = JJ-MM-TT
```

setFold() CATALOG

setFold(*neuerVerzName***)** ⇒ *alterVerzString*

Gibt den Namen des aktuellen Verzeichnisses als Zeichenkette zurück und macht *neuerVerzName* zum aktuellen Verzeichnis.

Das Verzeichnis *neuerVerzName* muß bereits vorhanden sein.

```
newFold chris ENTER          Done
setFold(main) ENTER       "chris"
setFold(chris)→oldfoldr ENTER
                          "main"
1→a ENTER                      1
setFold(#oldfoldr) ENTER  "chris"
a ENTER                        a
chris\a ENTER                  1
```

setGraph() CATALOG

setGraph(*ModusNameString***,** *EinstellString***)** ⇒ *String*

Setzt den Graph-Modus *ModusNameString* auf *EinstellungString* und gibt die vorherige Einstellung des Modus zurück. Wenn Sie die vorherige Einstellung abspeichern, können Sie sie später wiederherstellen.

ModusNameString ist eine Zeichenkette, die den gewünschten neuen Modus angibt. Es muß sich um einen der Modusnamen aus der nachfolgenden Tabelle handeln.

EinstellString ist eine Zeichenkette, die die gewünschte Einstellung für den Modus angibt. Es muß sich um eine der ebenfalls nachfolgend aufgeführten Einstellungen für den jeweiligen Modus handeln.

```
setGraph("Graph Order","Seq")
ENTER                      "SEQ"
setGraph("Coordinates","Off")
ENTER                      "RECT"
```

Hinweis: Großbuchstaben und Leerzeichen sind beim Eingeben von Modusbezeichnungen optional.

Modusname	Einstellungen
"Coordinates"	"Rect", "Polar", "Off"

"Graph Order"	"Seq", "Simul" [1]	
"Grid"	"Off", "On" [2]	
"Axes"	"Off", "On" "Off", "Axes", "Box"	(nicht im 3D Graph-Modus) (im 3D Graph-Modus)
"Leading Cursor"	"Off", "On" [2]	
"Labels"	"Off", "On"	
"Style"	"Wire Frame", "Hidden Surface", "Contour Levels", "Wire and Contour", "Implicit Plot" [3]	
"Seq Axes"	"Time", "Web", "U1-vs-U2" [4]	
"DE Axes"	"Time", "t-vs-y' ", "y-vs-y' ", "y1-vs-y2", "y1-vs-y2' ", "y1'-vs-y2' " [5] **Tipp:** Zur Eingabe eines Strichs (') drücken Sie 2nd [′].	
"Solution Method"	"RK", "Euler" [5]	
"Fields"	"SlpFld", "DirFld", "FldOff" [5]	

[1]Nicht verfügbar in den Modi Sequence, 3D oder Diff Equations Graph.
[2]Nicht verfügbar im Modus 3D Graph.
[3]Gilt nur für den Modus 3D Graph.
[4]Gilt nur für den Modus Sequence Graph.
[5]Gilt nur für den Modus Diff Equations Graph.

setMode() CATALOG

setMode(*ModusNameString***,** *EinstellString***)** ⇒ *String*
setMode(*Liste***)** ⇒ *StringListe*

Setzt den Modus *ModusNameString* auf die Einstellung *EinstellString* und gibt die aktuelle Einstellung dieses Modus zurück.

ModusNameString ist eine Zeichenkette, die den gewünschten Modus angibt. Es muß sich um einen der Modusnamen aus der nachfolgenden Tabelle handeln.

EinstellString ist eine Zeichenkette, die die gewünschte Einstellung für den Modus angibt. Es muß sich um eine der ebenfalls nachfolgend aufgeführten Einstellungen für den jeweiligen Modus handeln.

Sie können Modi und Einstellungen paarweise als *Liste* angeben. Diese Vorgehensweise wird empfohlen, wenn Sie mehrere Modi/Einstellungen gleichzeitig ändern möchten. Wenn Sie die Listenpaare der abgebildeten Beispiele in dieser Reihenfolge separat mit **setMode()** einzustellen versuchen, kann es sein, daß Sie nicht das gleiche Resultat erhalten wie bei der gleichzeitigen Änderung.

Einstellungen, die Sie mit **getMode("ALL")**→*Var* gespeichert haben, können Sie mit **setMode(***Var***)** wiederherstellen.

Hinweis: Zum Einstellen des Modus Unit System oder um Informationen über diesen abzurufen, verwenden Sie **setUnits()** bzw. **getUnits()** anstelle von **setMode()** oder **getMode()**.

```
setMode("Angle","Degree")
ENTER                          "RADIAN"
sin(45) ENTER                  √2/2

setMode("Angle","Radian")
ENTER                          "DEGREE"
sin(π/4) ENTER                 √2/2

setMode("Display Digits",
"Fix 2") ENTER                 "FLOAT"
π ◆ ENTER                      3.14

setMode ("Display Digits",
"Float") ENTER                 "FIX 2"
π ◆ ENTER                      3.141…

setMode ({"Split Screen",
"Left-Right","Split 1 App",
"Graph","Split 2 App","Table"})
ENTER
          {"Split 2 App" "Graph"
            "Split 1 App" "Home"
          "Split Screen" "FULL"}
```

Hinweis: Großbuchstaben und Leerzeichen sind beim Eingeben von Modusbezeichnungen optional. Die Ergebnisse dieser Beispiele weichen möglicherweise von den Resultaten auf Ihrem Gerät ab.

Modusname	**Einstellungen**
"Graph"	"Function", "Parametric", "Polar", "Sequence", "3D", "Diff Equations"
"Anzeige Digits"	"Fix 0", "Fix 1", ..., "Fix 12", "Float", "Float 1", ..., "Float 12"
"Angle"	"Radian", "Degree"
"Exponential Format"	"Normal", "Scientific", "Engineering"

"Complex Format"	"Real", "Rectangular", "Polar"
"Vector Format"	"Rectangular", "Cylindrical", "Spherical"
"Pretty Print"	"Off", "On"
"Split Screen"	"Full", "Top-Bottom", "Left-Right"
"Split 1 App"	"Home", "Y= Editor", "Window Editor", "Graph", "Table", "Data/Matrix Editor", "Program Editor", "Text Editor", "Numeric Solver", "*Flash-Anwendung*"
"Split 2 App"	"Home", "Y= Editor", "Window Editor", "Graph", "Table", "Data/Matrix Editor", "Program Editor", "Text Editor", "Numeric Solver", "*Flash-Anwendung*"
"Number of Graphs"	"1", "2"
"Graph2"	"Function", "Parametric", "Polar", "Sequence", "3D", "Diff Equations"
"Exact/Approx"	"Auto", "Exact", "Approximate"
"Base"	"Dec", "Hex", "Bin"
"Sprache"	"English", "*Andere Sprache*"

setTable() CATALOG

setTable(*ModusNameString***,** *EinstellString***)** ⇒ *String*

Setzt den Tabellenparameter *ModusNameString* auf die Einstellung *EinstellString* und gibt die vorherige Einstellung des Parameters zurück. Wenn Sie die vorherige Einstellung speichern, können Sie sie später wiederherstellen.

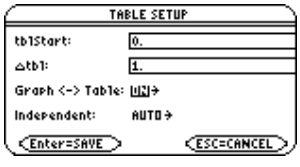
ModusNameString ist eine Zeichenkette, die den gewünschten Parameter angibt. Es muß sich um einen der Parameter aus der nachfolgenden Tabelle handeln.

EinstellString ist eine Zeichenkette, die die gewünschte neue Einstellung für den Parameter angibt. Es muß sich um eine der ebenfalls nachfolgend aufgeführten Einstellungen für den jeweiligen Parameter handeln.

```
setTable("Graph <-> Table","ON")
ENTER                              "OFF"

setTable("Independent","AUTO")
ENTER                              "ASK"
```

◆ [TblSet]



Hinweis: Großbuchstaben und Leerzeichen sind beim Eingeben von Parametern optional.

Parametername	**Einstellungen**
"Graph <-> Table"	"Off", "On"
"Independent"	"Auto", "Ask"

setTime() CATALOG

setTime(*hour,minute,second***)** ⇒ *listold*

Stellt die Uhr auf die durch das Argument vorgegebene Zeit und gibt eine Liste zurück. Die Liste hat das Format {*Alte Stunde,Alte Minute,Alte Sekunde*}. Die zurückgegebene Zeit stellt den vorher eingestellten Wert dar.

Geben Sie die Stunde im 24-Stunden-Format ein.

```
setTime(11,32,50)
                              {10  44  49}
```

setTmFmt() CATALOG

setTmFmt(*integer***)** ⇒ *integerold*

Stellt das Zeitformat dem Argument entsprechend ein und gibt das vorher eingestellte Zeitformat zurück.

Integer-Werte:

```
12 = 12 -Stunden-Uhr

24 = 24 -Stunden-Uhr
```

setTmZn() CATALOG

setTmZn(*integer***)** ⇒ *integerold*

Stellt die Zeitzone dem Argument entsprechend ein und gibt die vorher eingestellte Zeitzone zurück.

Der zurückgegebene Integer entspricht der Anzahl der Minuten, um welche die Zeitzone relativ zur Greenwich Mean Time (GMT) verschoben ist, die in Greenwich, England vorgegeben wird. Wenn die Zeitzone beispielsweise zur GMT um zwei Stunden verschoben ist, gibt das Gerät 120 (Minuten) zurück.

Integer für Zeitzonen westlich der GMT sind negativ.

Integer für Zeitzonen östlich der GMT sind positiv.

Wenn die Greenwich Mean Time 14:07:07 ist, ist es:

in Denver, Colorado (Mountain Daylight Time) 8:07:07 a.m.
(–360 Minuten zur GMT)

in Brüssel, Belgien (Central European Standard Time) 16:07:07 p.m.
(+120 Minuten zur GMT)

setUnits() CATALOG

setUnits(*Liste1***)** ⇒ *Liste*

Dient zum Einstellen der Standardeinheiten für die in *Liste1* angegebenen Werte und gibt eine Liste der vorherigen Standardeinstellungen zurück.

- Zur Angabe der vorhandenen Systeme SI (metrisch) oder ENG/US verwendet *Liste1* die Form:

  {"SI"} oder {"ENG/US"}

- Zur Angabe eines Satzes benutzerspezifischer Standardeinheiten verwendet *Liste1* die Form:

  {"CUSTOM", "*Kat1*", "*Einheit1*" [, "*Kat2*", "*Einheit2*", …]}

  wobei jedes *Kat*-/*Einheit*-Paar eine Kategorie und deren Standardeinheit angibt. (Sie können nur integrierte Einheiten angeben; die Angabe von benutzerdefinierten Einheiten ist nicht möglich.) Für jede nicht angegebene Kategorie wird ihre vorherige benutzerspezifische Einheit verwendet.

- Um zu den vorherigen benutzerspezifischen Einheiten zurückzukehren, verwendet *Liste1* die Form:

  {"CUSTOM"}

Möchten Sie, daß in unterschiedlichen Situationen unterschiedliche Standardeinheiten verwendet werden, erstellen Sie separate Listen, und speichern Sie diese mit eindeutigen Listennamen. Zur Verwendung eines Standardsatzes geben Sie dann den entsprechenden Listennamen in **setUnits()** an.

Einstellungen, die Sie mit **setUnits()**→ *Var* oder mit **getUnits()**→ *Var* gespeichert haben, können Sie mit **setUnits()** wiederherstellen.

Namen von Einheiten müssen mit einem Unterstrich (_) beginnen.

[◆] [_]
[2nd] [_]

Auch mit folgenden Tasten können Maßeinheiten aus einem Menü ausgewählt werden:

[2nd] [UNITS]
[◆] [UNITS]

```
setUnits({"SI"}) ENTER
          {"SI"  "Area"  "NONE"
      "Capacitance"  "_F"  ...}


setUnits({"CUSTOM","Length",
"_cm","Mass","_gm"}) ENTER
          {"SI"  "Length"  "_m"
            "Mass"  "_kg"  ...}
```

Hinweis: Auf Ihrem Bildschirm werden möglicherweise andere Einheiten angezeigt.

Shade CATALOG

Shade *Term1*, *Term2*, [*xUnten*], [*xOben*], [*Muster*], [*Musterdichte*]

Öffnet den Graphikbildschirm, zeichnet die Graphen für *Term1* und *Term2* und schraffiert die Bereiche, in denen *Term1* < *Term2*. (*Term1* und *Term2* müssen x als unabhängige Variable benutzen).

xUnten und *xOben* legen (sofern angegeben) die linke und die rechte Grenze für die Schraffur fest. Gültig sind Werte zwischen xmin und xmax. Vorgegeben sind xmin und xmax.

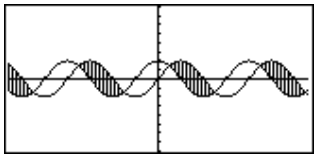
Muster legt eines der vier Schraffurmuster fest:
1 = vertikal (Vorgabe)
2 = horizontal
3 = negativ geneigt mit 45°
4 = positiv geneigt mit 45°

Musterdichte gibt den Abstand der Schraffurlinien an:
1= kein Abstand
2= 1 Pixel (Vorgabe)
3= 2 Pixel
⋮
10= 9 Pixel

Hinweis: Interaktiv können Sie eine Schraffur auf dem Graphikbildschirm mit der Anweisung **Shade** vornehmen. Das automatische Schraffieren einer bestimmten Funktion ist mit der Anweisung **Style** möglich. **Shade** ist nicht im Modus 3D Graph zulässig.

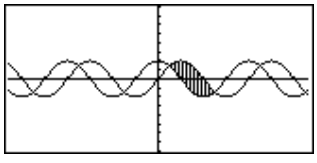
Im ZoomTrig-Ansichtfenster:

```
Shade cos(x),sin(x) ENTER
```



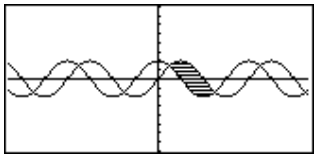
HOME
◆ [CALC HOME]

```
ClrDraw ENTER                    Done
Shade cos(x),sin(x),0,5 ENTER
```



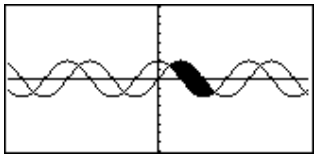
HOME
◆ [CALC HOME]

```
ClrDraw ENTER                    Done
Shade cos(x),sin(x),0,5,2 ENTER
```



HOME
◆ [CALC HOME]

```
ClrDraw ENTER                    Done
Shade cos(x),sin(x),0,5,2,1
ENTER
```



shift() CATALOG

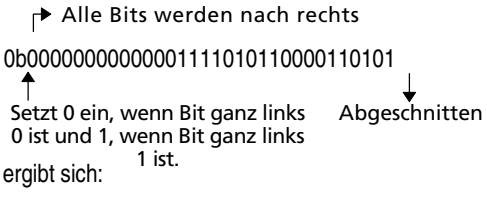
shift(*Ganze_Zahl1*[**,***Anzahl_Verschiebungen*]**)** ⇒ *Ganze_Zahl*

Verschiebt die Bits in einer binären ganzen Zahl. *Ganze_Zahl1* kann in jeder Basis eingegeben werden und wird stets automatisch in eine 32-Bit-Dualform konvertiert. Ist der Absolutwert von *Ganze_Zahl1* für diese Form zu groß, so wird eine symmetrische Modulo-Operation ausgeführt, um sie in den erforderlichen Bereich zu bringen.

Wenn *Anzahl_Verschiebungen* positiv ist, erfolgt die Verschiebung nach links.
Wenn *Anzahl_Verschiebungen* negativ ist, erfolgt die Verschiebung nach rechts.
Die Vorgabe ist ⁻1 (Verschiebung um ein Bit nach rechts).

In einer Rechtsverschiebung wird das ganz rechts stehende Bit abgeschnitten und als ganz links stehendes Bit eine 0 oder 1 eingesetzt. Bei einer Linksverschiebung wird das Bit ganz links abgeschnitten und 0 als letztes Bit rechts eingesetzt.

Aus beispielsweise folgender Rechtsverschiebung:



ergibt sich:

0b00000000000000011110101100001101O

Das Ergebnis wird gemäß dem jeweiligen Base-Modus angezeigt. Führende Nullen werden nicht angezeigt.

Im Modus Bin base:

```
shift(0b11110101100001101O1) ENTER
                0b11110101100001101O
shift(256,1) ENTER
                          0b1000000000
```

In Hex base mode:

```
shift(0h78E) ENTER              0h3C7
shift(0h78E,⁻2) ENTER           0h1E3
shift(0h78E,2) ENTER            0h1E38
```

Wichtig: Geben Sie eine Dual- oder Hexadezimalzahl stets mit dem Präfix 0b bzw. 0h ein (Null, nicht der Buchstabe O).

---

shift(*Liste1* [**,***Anzahl_Verschiebungen*]**)** ⇒ *List*

Gibt eine Kopie von *Liste1* zurück, in der die Elemente um *Anzahl_Verschiebungen* Elemente nach links oder rechts verschoben sind. *Liste1* bleibt unverändert.

Wenn *Anzahl_Verschiebungen* positiv ist, erfolgt die Verschiebung nach links.
Wenn *Anzahl_Verschiebungen* negativ ist, erfolgt die Verschiebung nach rechts.
Die Vorgabe ist ⁻1 (Verschiebung um ein Element nach rechts).

Dadurch eingeführte neue Elemente am Anfang bzw. am Ende von *Liste* werden auf "undefiniert" gesetzt.

Im Modus Dec base:

```
shift({1,2,3,4}) ENTER
                    {undef 1 2 3}
shift({1,2,3,4},⁻2) ENTER
                  {undef undef 1 2}
shift({1,2,3,4},1) ENTER
                    {2 3 4 undef}
```

shift(*String1* [**,***Anzahl_Verschiebungen*]**)** ⇒ *String*

Gibt eine um *Anzahl_Verschiebungen* Zeichen nach rechts oder links verschobene Kopie von *String1* zurück. Verändert *String1* nicht.

Wenn *Anzahl_Verschiebungen* positiv ist, erfolgt die Verschiebung nach links.
Wenn *Anzahl_Verschiebungen* negativ ist, erfolgt die Verschiebung nach rechts.
Die Vorgabe ist ⁻1 (Verschiebung um ein Zeichen nach rechts).

Dadurch eingeführte neue Zeichen am Anfang bzw. am Ende von *String* werden auf ein Leerzeichen gesetzt.

```
shift("abcd") ENTER          " abc"
shift("abcd",-2) ENTER       "  ab"
shift("abcd",1) ENTER        "bcd "
```

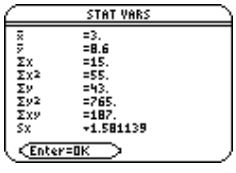
ShowStat CATALOG

ShowStat

Zeigt ein Dialogfeld mit den Ergebnissen der letzten statistischen Berechnung an, sofern diese noch gültig sind. Statistikergebnisse werden automatisch gelöscht, wenn sich die Daten geändert haben, die den Berechnungen zugrunde liegen.

Benutzen Sie diese Anweisung nach statistischen Berechnungen wie etwa **LinReg**.

```
{1,2,3,4,5}→L1 ENTER  {1 2 3 4 5}
{0,2,6,10,25}→L2 ENTER
                      {0 2 6 10 25}
TwoVar L1,L2 ENTER
ShowStat ENTER
```



sign() MATH/Number-Menü

sign(*Term1***)** ⇒ *Term*
sign(*Liste1***)** ⇒ *Liste*
sign(*Matrix1***)** ⇒ *Matrix*

Gibt *Term1*/**abs(***Term1***)** für reelle und komplexe *Term1* zurück, wenn *Term1*≠ 0.

Gibt 1 zurück, wenn *Term1* positiv ist.
Gibt -1 zurück, wenn *Term1* negativ ist.
sign(0**)** gibt ±1 zurück, wenn als Komplex-Formatmodus REAL eingestellt ist; anderenfalls gibt es sich selbst zurück.
sign(0**)** stellt im komplexen Bereich den Einheitskreis dar.

Gibt für jedes Element einer Liste bzw. Matrix das Vorzeichen zurück.

```
sign(-3.2) ENTER                -1.
sign({2,3,4,-5}) ENTER
                        {1  1  1 -1}
sign(1+abs(x)) ENTER             1
```

Beim Komplex-Formatmodus REAL:

```
sign([-3,0,3]) ENTER    [-1 ±1 1]
```

simult() MATH/Matrix-Menü

simult(*KoeffMatrix***,** *KonstVektor***[,** *Tol***])** ⇒ *Matrix*

Ergibt einen Spaltenvektor, der die Lösungen für ein lineares Gleichungssystem enthält.

KoeffMatrix muß eine quadratische Matrix sein, welche die Koeffizienten der Gleichung enthält.

KonstVektor muß die gleiche Zeilenanzahl (gleiche Dimension) besitzen wie *KoeffMatrix* und die Konstanten enthalten.

Sie haben die Option, daß jedes Matrixelement als Null behandelt wird, wenn dessen absoluter Wert geringer als *Tol* ist. Diese Toleranz wird nur dann verwendet, wenn die Matrix Fließkommaelemente aufweist und keinerlei symbolische Variablen ohne zugewiesene Werte enthält. Anderenfalls wird *Tol* ignoriert.

- Wenn Sie ◆ ENTER verwenden oder den Modus auf Exact/Approx=APPROXIMATE einstellen, werden Berechnungen im Fließkomma-Rechenmodus durchgeführt.
- Wird *Tol* weggelassen oder nicht verwendet, so wird die Standardtoleranz folgendermaßen berechnet:

  5E⁻14 * **max(dim(***KoeffMatrix***))** * **rowNorm(***KoeffMatrix***)**

Auflösen nach x und y: x + 2y = 1, 3x + 4y = ⁻1

```
simult([1,2;3,4],[1;⁻1]) ENTER
```

$$\begin{bmatrix} -3 \\ 2 \end{bmatrix}$$

Die Lösung ist x= ⁻3 und y=2.

Auflösen: ax + by = 1, cx + dy = 2

```
[a,b;c,d]→matx1 ENTER
```

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

```
simult(matx1,[1;2]) ENTER
```

$$\begin{bmatrix} \frac{-(2\cdot b-d)}{a\cdot d-b\cdot c} \\ \frac{2\cdot a-c}{a\cdot d-b\cdot c} \end{bmatrix}$$

---

simult(*KoeffMatrix***,** *KonstMatrix***[,** *Tol***])** ⇒ *Matrix*

Löst mehrere lineare Gleichungssysteme, welche alle dieselben Gleichungskoeffizienten aber unterschiedliche Konstanten haben.

Jede Spalte in *KonstMatrix* muß die Konstanten für ein Gleichungssystem enthalten. Jede Spalte in der sich ergebenden Matrix enthält die Lösung für das entsprechende System.

Auflösen x + 2y = 1, 3x + 4y = ⁻1 und x + 2y = 2, 3x + 4y = ⁻3

```
simult([1,2;3,4],[1,2;⁻1,⁻3])
ENTER
```

$$\begin{bmatrix} -3 & -7 \\ 2 & 9/2 \end{bmatrix}$$

Für das erste System ist x= ⁻3 und y=2. Für das zweite System ist x= ⁻7 und y=9/2.

sin() 2nd [SIN] Taste SIN Taste

sin(*Term1***)** ⇒ *Term*
sin(*Liste1***)** ⇒ *Liste*

sin(*Term1***)** gibt den Sinus des Parameters als Term zurück.

sin(*Liste1***)** gibt eine Liste zurück, die für jedes Element von *Liste1* den Sinus enthält.

Hinweis: Der Parameter wird gemäß der aktuellen Winkel-Moduseinstellung als Ordnung oder Radiant interpretiert. Sie können ° oder ʳ benutzen, um die Winkel-Moduseinstellung temporär zu ändern.

Im Degree-Modus für Winkel:

```
sin((π/4)ʳ) ENTER
```

$$\frac{\sqrt{2}}{2}$$

```
sin(45) ENTER
```

$$\frac{\sqrt{2}}{2}$$

```
sin({0,60,90}) ENTER
```

$$\{0 \quad \frac{\sqrt{3}}{2} \quad 1\}$$

Im Radian-Modus:

```
sin(π/4) ENTER
```

$$\frac{\sqrt{2}}{2}$$

```
sin(45°) ENTER
```

$$\frac{\sqrt{2}}{2}$$

sin(*quadrat_Matrix1***)** ⇒ *quadrat_Matrix*

Ergibt den Matrix-Sinus von *quadrat_Matrix1*. Dies ist *nicht* gleichbedeutend mit der Berechnung des Sinus jedes einzelnen Elements. Näheres über das Berechnungsverfahren finden Sie im Abschnitt **cos()**.

Quadrat_Matrix1 muß diagonalisierbar sein. Das Ergebnis enthält stets Fließkommazahlen.

Im Radian-Modus:

```
sin([1,5,3;4,2,1;6, -2,1]) ENTER
```

$$\begin{bmatrix} .942\ldots & -.045\ldots & -.031\ldots \\ -.045\ldots & .949\ldots & -.020\ldots \\ -.048\ldots & -.005\ldots & .961\ldots \end{bmatrix}$$

$\sin^{-1}()$ ▦ ◆ [SIN⁻¹] Taste 📟 2nd [SIN⁻¹] Taste

sin⁻¹(*Term1***)** ⇒ *Term*
sin⁻¹(*Liste1***)** ⇒ *Liste*

sin⁻¹(*Term1***)** gibt als Term den Winkel zurück, dessen Sinus *Term1* ist.

sin⁻¹(*Liste1***)** gibt eine Liste zurück, die für jedes Element von *Liste1* den inversen Sinus enthält.

Hinweis: Das Ergebnis wird gemäß der aktuellen Winkel-Moduseinstellung entweder in Ordnung oder im Bogenmaß zurückgegeben.

Im Degree-Modus für Winkel:

```
sin⁻¹(1) ENTER                                  90
```

Im Radian-Modus:

```
sin⁻¹({0,.2,.5}) ENTER
                          {0   .201…   .523…}
```

---

sin⁻¹(*quadrat_Matrix1***)** ⇒ *quadrat_Matrix*

Ergibt den inversen Matrix-Sinus von *quadrat_Matrix1*. Dies ist *nicht* gleichbedeutend mit der Berechnung des inversen Sinus jedes einzelnen Elements. Näheres zum Berechnungsverfahren finden Sie im Abschnitt **cos()**.

Quadrat_Matrix1 muß diagonalisierbar sein. Das Ergebnis enthält stets Fließkommazahlen.

Im Winkelmodus Radian und im Komplex-Formatmodus "Rectangular":

```
sin⁻¹ ([1,5,3;4,2,1;6,-2,1]) ENTER
```

$$\begin{bmatrix} -.164\ldots-.064\ldots\cdot i & 1.490\ldots-2.105\ldots\cdot i & \ldots \\ .725\ldots-1.515\ldots\cdot i & .947\ldots-.778\ldots\cdot i & \ldots \\ 2.083\ldots-2.632\ldots\cdot i & -1.790\ldots+1.271\ldots\cdot i & \ldots \end{bmatrix}$$

sinh() MATH/Hyperbolic-Menü

sinh(*Term1***)** ⇒ *Term*
sinh(*Liste1***)** ⇒ *Liste*

sinh (*Term1***)** gibt den Sinus hyperbolicus des Parameters als Term zurück.

sinh (*Liste***)** gibt in Form einer Liste für jedes Element aus *Liste1* den Sinus hyperbolicus zurück.

```
sinh(1.2) ENTER                                 1.509…
sinh({0,1.2,3.}) ENTER
                          {0   1.509…   10.017…}
```

---

sinh(*quadrat_Matrix1***)** ⇒ *quadrat_Matrix*

Ergibt den Matrix-Sinus hyperbolicus von *quadrat_Matrix1*. Dies ist *nicht* gleichbedeutend mit der Berechnung des Sinus hyperbolicus jedes einzelnen Elements. Näheres zur Berechnungsmethode finden Sie im Abschnitt **cos()**.

Quadrat_Matrix1 muß diagonalisierbar sein. Das Ergebnis enthält immer Fließkommazahlen.

Im Radian-Modus:

```
sinh([1,5,3;4,2,1;6,-2,1]) ENTER
```

$$\begin{bmatrix} 360.954 & 305.708 & 239.604 \\ 352.912 & 233.495 & 193.564 \\ 298.632 & 154.599 & 140.251 \end{bmatrix}$$

$\sinh^{-1}()$ MATH/Hyperbolic-Menü

sinh⁻¹(*Term1***)** ⇒ *Term*
sinh⁻¹(*Liste1***)** ⇒ *Liste*

sinh⁻¹(*Term1***)** gibt den inversen Sinus hyperbolicus des Parameters als Term zurück.

sinh⁻¹(*Liste1***)** gibt in Form einer Liste für jedes Element aus *Liste1* den inversen Sinus hyperbolicus zurück.

```
sinh⁻¹(0) ENTER                                 0
sinh⁻¹({0,2.1,3}) ENTER
                          {0   1.487…   sinh⁻¹(3)}
```

sinh^{-1}(*quadrat_Matrix1***)** ⇒ *quadrat_Matrix*

Ergibt den inversen Matrix-Sinus hyperbolicus von *quadrat_Matrix1*. Dies ist *nicht* gleichbedeutend mit der Berechnung des inversen Sinus hyperbolicus jedes einzelnen Elements. Näheres zur Berechnungsmethode finden Sie im Abschnitt **cos()**.

Quadrat_Matrix1 muß diagonalisierbar sein. Das Ergebnis enthält immer Fließkommazahlen.

Im Radian-Modus:

sinh^{-1}([1,5,3;4,2,1;6,⁻2,1]) ENTER

$$\begin{bmatrix} .041\ldots & 2.155\ldots & 1.158\ldots \\ 1.463\ldots & .926\ldots & .112\ldots \\ 2.750\ldots & -1.528\ldots & .572\ldots \end{bmatrix}$$

SinReg MATH/Statistics/Regressions-Menü

SinReg *Liste1*, *Liste2* [, [*Iterationen*], [*Periode*] [, *Liste3*, *Liste4*]]

Berechnet die Sinusregression und aktualisiert alle Systemstatistik-Variablen.

Alle Listen außer *Liste4* müssen die gleiche Dimension besitzen.

Liste1 stellt die Liste der x-Werte dar.
Liste2 stellt die Liste der y-Werte dar.
Liste3 stellt die Klassencodes dar.
Liste4 stellt die Klassenliste dar.

Iterationen gibt an, wie viele Lösungsversuche (1 bis 16) maximal stattfinden. Bei Auslassung wird 8 verwendet. Größere Werte führen in der Regel zu höherer Genauigkeit aber auch längeren Ausführungszeiten und umgekehrt.

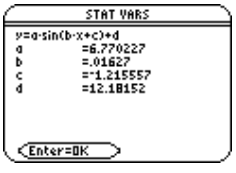
Periode gibt eine geschätzte Periode an. Bei Auslassung sollten die Werte in *Liste1* sequentiell angeordnet und die Differenzen zwischen ihnen gleichmäßig sein. Wenn Sie *Periode* jedoch angeben, können die Differenzen zwischen den einzelnen x-Werten ungleichmäßig sein.

Hinweis: *Liste1* bis einschl. *Liste3* müssen Variablennamen oder c1–c99 sein (Spalten in der letzten Datenvariablen, die im Daten/ Matrix-Editor angezeigt wurde). *Liste4* braucht kein Variablenname zu sein und kann keine Spalte c1–c99 sein.

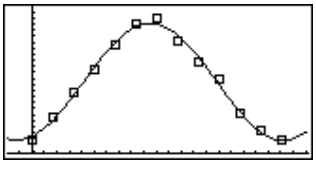
SinReg wird unabhängig von der Winkelmodus-Einstellung stets in Radian ausgegeben.

Im Funktions-Graphikmodus:

seq(x,x,1,361,30)→L1 ENTER
{1 31 61 …}
{5.5,8,11,13.5,16.5,19,19.5,17,14.5,12.5,8.5,6.5,5.5}→L2 ENTER
{5.5 8 11 …}
SinReg L1,L2 ENTER Done
ShowStat ENTER



ENTER
regeq(x)→y1(x) ENTER Done
NewPlot 1,1,L1,L2 ENTER Done
◆ [GRAPH]
F2 9



solve() MATH/Algebra-Menü

solve(*Gleichung***,** *Var***)** ⇒ *Boolescher Term*
solve(*Ungleichung***,** *Var***)** ⇒ *Boolescher Term*

Gibt mögliche reelle Lösungen einer Gleichung oder Ungleichung für *Var* zurück. Das Ziel ist, Kandidaten für alle Lösungen zu erhalten. Es kann jedoch Gleichungen oder Ungleichungen geben, für die es eine unendliche Anzahl Lösungen gibt.

Für manche Wertekombinationen undefinieter Variablen kann es sein, dass mögliche Lösungen nicht reell und endlich sind.

Ist der Modus Exact/Approx auf AUTO eingestellt, ist das Ziel die Ermittlung exakter kompakter Lösungen, wobei ergänzend eine iterative Suche mit Näherungslösungen benutzt wird, wenn exakte Lösungen sich als unpraktisch erweisen.

solve(a*x^2+b*x+c=0,x) ENTER

$$x = \frac{\sqrt{b^2-4\cdot a\cdot c}-b}{2\cdot a}$$

$$\text{or } x = \frac{-\sqrt{b^2-4\cdot a\cdot c}+b}{2\cdot a}$$

ans(1)| a=1 and b=1 and c=1 ENTER
Error: Non-real result

solve((x-a)e^(x)=⁻x*(x-a),x) ENTER
x=a or x=⁻.567…

Da Quotienten standarmäßig mit dem ggT von Zähler und Nenner gekürzt werden, kann es sein, dass Lösungen nur in den Grenzwerten von einer oder beiden Seiten liegen.

```
(x+1)(x-1)/(x-1)+x-3 ENTER   2·x-2
solve(entry(1)=0,x) ENTER      x=1
entry(2)|ans(1) ENTER        undef
limit(entry(3),x,1) ENTER        0
```

Für Ungleichungen der Typen ≥, ≤, < oder > sind explizite Lösungen unwahrscheinlich, es sei denn, die Ungleichung ist linear und enthält nur *Var*.

```
solve(5x-2≥2x,x) ENTER      x≥2/3
```

Ist der Modus Exact/Approx auf EXACT eingestellt, werden nicht lösbare Teile als implizite Gleichung oder Ungleichung zurückgegeben.

```
exact(solve((x-a)e^(x)=⁻x*
  (x-a),x)) ENTER
                  e^x+x=0 or x=a
```

Verwenden Sie den Operator "|" zur Einschränkung des Lösungsintervalls und/oder zur Einschränkung anderer Variablen, die in der Gleichung bzw. Ungleichung vorkommen. Wenn Sie eine Lösung in einem Intervall gefunden haben, können Sie die Ungleichungsoperatoren benutzen, um dieses Intervall aus nachfolgenden Suchläufen auszuschließen.

Im Radian-Modus:

```
solve(tan(x)=1/x,x)|x>0 and x<1
ENTER                       x=.860...
```

Wenn keine reellen Lösungen ermittelt werden können, wird "falsch" zurückgegeben. "wahr" wird zurückgegeben, wenn **solve()** feststellt, dass jeder endliche reelle Wert von *Var* die Gleichung bzw. Ungleichung erfüllt.

```
solve(x=x+1,x) ENTER          false
solve(x=x,x) ENTER             true
```

Da **solve()** stets ein Boolesches Ergebnis liefert, können Sie "and", "or" und "not" verwenden, um Ergebnisse von **solve()** miteinander oder mit anderen Booleschen Termen zu verknüpfen.

```
2x-1≤1 and solve(x^2≠9,x) ENTER
                  x≤1 and x≠⁻3
```

Lösungen können eine neue unbestimmte Variable der Form @n*j* enthalten, wobei *j* eine ganze Zahl im Bereich 1–255 ist. Eine solche Variable steht für eine "beliebige ganze Zahl".

Im Radian-Modus:

```
solve(sin(x)=0,x) ENTER   x=@n1·π
```

Im reellen Modus bezeichnen Bruchpotenzen mit ungeradem Nenner nur das reelle Intervall. Ansonsten bezeichnen zusammengesetzte Terme wie Bruchpotenzen, Logarithmen und inverse trigonometrische Funktionen nur das Hauptintervall. Demzufolge liefert **solve()** nur Lösungen, die diesem einen reellen oder Hauptintervall entsprechen.

```
solve(x^(1/3)=⁻1,x) ENTER    x=⁻1
solve(√(x)=⁻2,x) ENTER       false
solve(⁻√(x)=⁻2,x) ENTER       x=4
```

Hinweis: Siehe auch **cSolve()**, **cZeros()**, **nSolve()** und **zeros()**.

---

solve(*Gleichung1* **and** *Gleichung2* [**and** *...*], **{***VarOderSchätzung1***,** *VarOderSchätzung2* [**,** *...*]**})** ⇒ *Boolescher Term*

Gibt mögliche reelle Lösungen für das Gleichungssystem zurück, wobei jedes *VarOderSchätzwert* eine Variable darstellt, nach der Sie die Gleichungen auflösen möchten.

Sie haben die Option, einen Ausgangsschätzwert für eine Variable anzugeben. *VarOderSchätzwert* muß immer folgende Form haben:

Variable
– oder –
Variable = reelle oder nicht-reelle Zahl

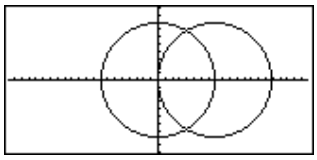
Beispiel: x ist gültig, und x=3 ebenfalls.

Wenn alle Gleichungen Polynome sind und Sie KEINE Anfangsschätzwerte angeben, dann verwendet **Solve()** das lexikalische Gröbner/Buchbergersche Eliminationsverfahren

```
solve(y=x^2-2 and
x+2y=⁻1,{x,y}) ENTER
                      x=1 and y=⁻1
              or x=⁻3/2 and y=1/4
```

beim Versuch, **alle** reellen Lösungen zu bestimmen.

Betrachten wir z.B. einen Kreis mit dem Radius r und dem Ursprung als Mittelpunkt und einen weiteren Kreis mit Radius r und dem Schnittpunkt des ersten Kreises mit der positiven X-Achse als Mittelpunkt. Verwenden Sie **solve()** zur Bestimmung der Schnittpunkte.



Wie in nebenstehendem Beispiel durch r demonstriert, können *Gleichungssysteme* zusätzliche Variablen ohne Wert aufweisen, die aber für numerische Werte stehen, welche später eingesetzt werden können.

```
solve(x^2+y^2=r^2 and
(x-r)^2+y^2=r^2,{x,y}) ENTER
```

$$x=\frac{r}{2} \text{ and } y=\frac{\sqrt{3}\cdot r}{2}$$

$$\text{or } x=\frac{r}{2} \text{ and } y=\frac{-\sqrt{3}\cdot r}{2}$$

Sie können auch (oder statt dessen) Lösungsvariablen angeben, die in den Gleichungen nicht erscheinen. Geben Sie zum Beispiel z als eine Lösungsvariable an, um das vorangehende Beispiel auf zwei parallele sich schneidende Zylinder mit dem Radius r auszudehnen.

```
solve(x^2+y^2=r^2 and
(x-r)^2+y^2=r^2,{x,y,z}) ENTER
```

$$x=\frac{r}{2} \text{ and } y=\frac{\sqrt{3}\cdot r}{2} \text{ and } z=@1$$

$$\text{or } x=\frac{r}{2} \text{ and } y=\frac{-\sqrt{3}\cdot r}{2} \text{ and } z=@1$$

Die Zylinder-Lösungen verdeutlichen, daß Lösungsfamilien "beliebige" Konstanten der Form *@k* enthalten können, wobei *k* ein ganzzahliger Index im Bereich 1 bis 255 ist. Der Index wird wieder auf 1 zurückgesetzt, wenn Sie **ClrHome** oder F1 8:Clear Home verwenden.

Bei Gleichungssystemen kann die Berechnungsdauer oder Speicherbelastung stark von der Reihenfolge abhängen, in welcher Sie die Lösungsvariablen auflisten. Übersteigt Ihre erste Wahl die Speicherkapazität oder Ihre Geduld, versuchen Sie, die Variablen in den Gleichungen und/oder der *VarOderSchätzung*-Liste umzuordnen.

Wenn Sie keine Schätzungswerte angeben und eine Gleichung in einer Variablen kein Polynom ist, aber alle Gleichungen in den Lösungsvariablen linear sind, so verwendet **solve()** das Gaußsche Eliminationsverfahren beim Versuch, alle reellen Lösungen zu bestimmen.

```
solve(x+e^(z)*y=1 and
x-y=sin(z),{x,y}) ENTER
```

$$x=\frac{e^{z}\cdot\sin(z)+1}{e^{z}+1} \text{ and } y=\frac{-(\sin(z)-1)}{e^{z}+1}$$

Wenn ein System weder in all seinen Variablen ein Polynom noch in seinen Lösungsvariablen linear ist, dann bestimmt **solve()** mindestens eine Lösung anhand eines iterativen Näherungsverfahrens. Hierzu muß die Anzahl der Lösungsvariablen gleich der Gleichungsanzahl sein, und alle anderen Variablen in den Gleichungen müssen zu Zahlen vereinfachbar sein.

```
solve(e^(z)*y=1 and
-y=sin(z),{y,z}) ENTER
```

y=.041… and z=3.183…

Jede Lösungsvariable beginnt bei dem entsprechenden geschätzten Wert, falls vorhanden; ansonsten beginnt sie bei 0,0.

Suchen Sie anhand von Schätzwerten nach einzelnen zusätzlichen Lösungen. Für Konvergenz sollte eine Schätzung ziemlich nahe bei einer Lösung liegen.

```
solve(e^(z)*y=1 and
-y=sin(z),{y,z=2π}) ENTER
```

y=.001… and z=6.281…

SortA MATH/List-Menü

SortA *ListeName1*[, *ListeName2*] [, *ListeName3*] **...**
SortA *VektorName1*[, *VektorName2*] [, *VektorName3*] **...**

Sortiert die Elemente des ersten Parameters in aufsteigender Reihenfolge.

Bei Angabe von mehr als einem Parameter werden die Elemente der zusätzlichen Parameter so sortiert, daß ihre neue Position mit der neuen Position der Elemente des ersten Parameters übereinstimmt.

Alle Parameter müssen Listen- oder Vektornamen sein. Alle Parameter müssen die gleiche Dimension besitzen.

```
{2,1,4,3}→list1 ENTER    {2,1,4,3}
SortA list1 ENTER              Done

list1 ENTER              {1 2 3 4}
{4,3,2,1}→list2 ENTER    {4 3 2 1}
SortA list2,list1 ENTER        Done

list2 ENTER              {1 2 3 4}
list1 ENTER              {4 3 2 1}
```

SortD MATH/List-Menü

SortD *ListeName1*[, *ListeName2*] [, *ListeName3*] **...**
SortD *VektorName1*[,*VektorName 2*] [,*VektorName 3*] **...**

Identisch mit **SortA** mit dem Unterschied, daß **SortD** die Elemente in absteigender Reihenfolge sortiert.

```
{2,1,4,3}→list1 ENTER    {2 1 4 3}
{1,2,3,4}→list2 ENTER    {1 2 3 4}
SortD list1,list2 ENTER        Done
list1 ENTER              {4 3 2 1}
list2 ENTER              {3 4 1 2}
```

▸Sphere MATH/Matrix/Vector ops-Menü

Vektor **▸Sphere**

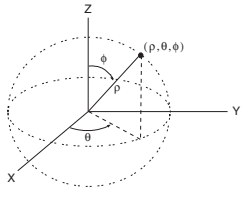
Zeigt den Zeilen- oder Spaltenvektor in Kugelkoordinaten [ρ ∠θ ∠ϕ] an.

Vektor muß die Dimension 3 besitzen und kann ein Zeilen- oder ein Spaltenvektor sein.

Hinweis: **▸Sphere** ist eine Anzeigeformat-anweisung, keine Konvertierungsfunktion. Sie können sie nur am Ende einer Eingabezeile benutzen.

```
[1,2,3]▸Sphere
◆ ENTER [3.741… ∠1.107… ∠.640…]

[2,∠π/4,3]▸Sphere
◆ ENTER  [3.605… ∠.785… ∠.588…]
ENTER  [√13  ∠π/4  ∠cos⁻¹((3·√13)/13)]
```



startTmr() CATALOG

startTmr() ⇒ *integer*

Gibt den aktuellen Wert der Uhr im Integerformat (ganze Zahl) zurück, wodurch die Startzeit (engl. *starttime)* für einen Timer vorgegeben wird. Sie können die *starttime* als Argument von **checkTmr()** angeben, um festzustellen, wieviele Sekunden vergangen sind.

Sie können mehrere Timer gleichzeitig laufen lassen.

Hinweis: Siehe auch **checkTmr()** und **timeCnv()**.

```
startTmr() ENTER          148083315

checkTmr(148083315)            34


startTmr()→Timer1
⋮
startTmr()→Timer2
⋮
checkTmr(Timer1)→Timer1Wert
⋮
checkTmr(Timer2)→Timer2Wert
```

stdDev() MATH/Statistics-Menü

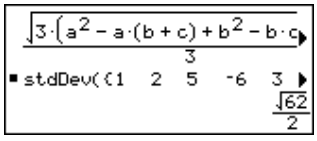
stdDev(*list*[, *freqlist*]**)** ⇒ *expression*

Ergibt die Standardabweichung der Elemente in *list*.

Jedes *freqlist*-Element gibt die Häufigkeit der entsprechenden Elemente in *list* an.

Hinweis: *list* muß mindestens zwei Elemente enthalten.

```
stdDev({a,b,c}) ENTER
stdDev({1,2,5,-6,3,-2}) ENTER
```




```
stdDev({1.3,2.5,-6.4},{3,2,5})
ENTER                          4.33345
```

---

stdDev(*matrix1*[, *freqmatrix*]**)** ⇒ *matrix*

Ergibt einen Reihenvektor der Standardabweichungen der Spalten in *matrix1*.

Jedes *freqmatrix*-Element gibt die Häufigkeit der zugehörigen Elemente in *matrix1* an.

Hinweis: *matrix1* muß mindestens zwei Zeilen enthalten.

```
stdDev([1,2,5;-3,0,1;.5,.7,3])
ENTER
              [2.179...  1.014...  2]
stdDev([-1.2,5.3;2.5,7.3;6,-4],
[4,2;3,3;1,7]) ENTER
                  [2.7005,5.44695]
```

StoGDB CATALOG

StoGDB *GDBVar*

Erzeugt eine Graphik-Einstellungs-Variable (GDB), die folgende aktuelle Angaben enthält:

* Graphikmodus
* Y= Funktionen
* Fenstervariablen
* Graphikformateinstellungen
  1- oder 2-Graph-Einstellung (Split-Screen und Verhältniseinstellungen bei 2-Graph- Modus)
  Winkelmodus
  Modus Real/Complex (reell/komplex)
* Anfangsbedingungen, sofern Modus Sequence oder Diff Equations
* Tabellen-Flags
* tblStart, Δtbl, tblInput

Mit **RclGDB** *GDBVar* können Sie die Umgebung für die graphische Darstellung wiederherstellen.

*__Hinweis:__ Diese Angaben werden im 2-Graph-Modus für beide Graphen gespeichert.

Stop CATALOG

Stop

Programmieranweisung, die die Programmausführung stoppt.

Programmsegment:

```
  ⋮
For i,1,10,1
  If i=5
  Stop
EndFor
  ⋮
```

StoPic CATALOG

StoPic *picVar* [**,** *pxlZeile***,** *pxlSpalte*] [**,** *Breite***,** *Höhe*]

Öffnet den Graphikbildschirm und kopiert einen rechteckigen Bereich in die Variable *picVar*.

PxlZeile und *pxlSpalte*(wahlfrei) geben die linke obere Ecke des zu kopierenden Bereichs an (Vorgabe ist 0, 0).

Breite und *Höhe* (wahlfrei) geben die Abmessung des Bereichs in Pixeln an. Vorgabe sind die Höhe und die Breite des aktuellen Graphikbildschirms (in Pixel).

Store

Siehe → (store), Seite 995.

string() MATH/String-Menü

string(*Term***)** ⇒ *String*

Vereinfacht *Term* und gibt das Ergebnis als Zeichenkette zurück.

```
string(1.2345) ENTER          "1.2345"
string(1+2) ENTER                 "3"
string(cos(x)+√(3)) ENTER
                          "cos(x)+√(3)"
```

Style CATALOG

Style *Gleichung_Nr, StilString*

Setzt die Y = Funktion *Gleichung_Nr* im aktuellen Graphikmodus auf den Graphdarstellungsstil *StilString*.

Gleichung_Nr muß eine ganze Zahl im Bereich 1–99 sein, und die Gleichung muß bereits existieren.

StilString muß eine der folgenden Angaben sein: "Line", "Dot", "Square", "Thick", "Animate", "Path", "Above" oder "Below".

Beachten Sie, daß bei Parameterdarstellung nur die *xt*-Hälfte die Stilinformationen enthält.

Gültige Stilnamen im Graphikmodus:

Function:	alle Stile
Parametric/Polar:	line, dot, square, thick, animate, path
Sequence:	line, dot, square, thick
3D:	keine
Diff Equations:	line, dot, square, thick, animate, path

Hinweis: Großschreibung und Leerzeichen sind bei der Eingabe von *StilString* optional.

```
Style 1,"thick" ENTER          Done
Style 10,"path" ENTER          Done
```

Hinweis: Im Funktions-Graphikmodus wird mit diesen Beispielen der Stil für y1(x) auf "Thick" und für y10(x) auf "Path" gesetzt.

subMat() CATALOG

subMat(*Matrix1***[,** *Startzeile***] [,** *Startspalte***] [,** *Endzeile***]**
[, *Endspalte***])** ⇒ *Matrix*

Gibt die angegebene Untermatrix von *Matrix1* zurück.

Vorgaben: *Startzeile*=1, *Startspalte*=1, *Endzeile*=letzte Zeile, *Endspalte*=letzte Spalte.

```
[1,2,3;4,5,6;7,8,9]→m1 ENTER
                          [1 2 3]
                          [4 5 6]
                          [7 8 9]
subMat(m1,2,1,3,2) ENTER
                            [4 5]
                            [7 8]
subMat(m1,2,2) ENTER
                            [5 6]
                            [8 9]
```

sum() MATH/List-Menü

sum(*list***[**, *start***[**, *end***]])** ⇒ *expression*

Ergibt die Summe der Elemente in in *list*.

Start und *end* sind nicht unbedingt erforderlich; sie dienen zur Angabe eines Elementbereichs.

```
sum({1,2,3,4,5}) ENTER          15
sum({a,2a,3a}) ENTER          6·a
sum(seq(n,n,1,10)) ENTER        55
sum({1,3,5,7,9},3) ENTER        21
```

---

sum(*matrix1***[**, *start***[**, *end***]])** ⇒ *matrix*

Ergibt einen Zeilenvektor mit den Summen der Elemente in den Spalten von *matrix1*.

Start und *end* sind nicht unbedingt erforderlich; sie dienen zur Angabe eines Zeilenbereichs.

```
sum([1,2,3;4,5,6]) ENTER [5  7  9]
sum([1,2,3;4,5,6;7,8,9]) ENTER
                          [12 15 18]
sum([1,2,3;4,5,6;7,8,9],2,3)
ENTER
                          [11,13,15]
```

Summe() Siehe **Σ()**, Seite 990.

switch() CATALOG

switch([*Ganze_Zahl1***])** ⇒ *Ganze_Zahl*

Gibt die Nummer des aktiven Fensters zurück oder setzt das aktive Fenster.

Hinweis: Window 1 ist das linke bzw. das obere; Window 2 ist das rechte bzw. das untere.

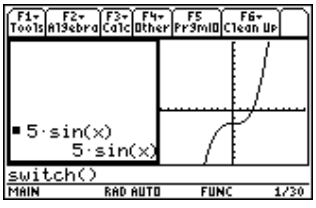
Mit *Ganze_Zahl1* = 0 wird die Nummer des aktiven Fensters zurückgegeben.

Mit *Ganze_Zahl1* = 1 wird Fenster 1 aktiviert und die Nummer des zuvor aktiven Fensters zurückgegeben.

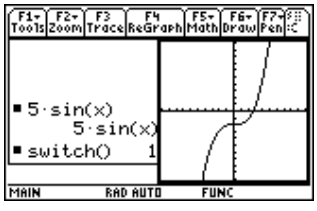
Mit *Ganze_Zahl1* = 2 wird Fenster 2 aktiviert und die Nummer des zuvor aktiven Fensters zurückgegeben.

Wird keine Angabe für *Ganze_Zahl1* gemacht, wird das aktive Fenster gewechselt und die Nummer des zuvor aktiven Fensters zurückgegeben.

Ganze_Zahl1 wird ignoriert, wenn der TI-89 Titanium / Voyage™ 200 zu diesem Zeitpunkt keinen geteilten Bildschirm anzeigt.




```
switch() ENTER
```



T (Transpon.) **MATH/Matrix-Menü**

Matrix1^{T} ⇒ *Matrix*

Gibt die Konjugiert-Komplexe transponierte Matrix von *Matrix1* zurück.

`[1,2,3;4,5,6;7,8,9]→mat1` ENTER

$$\begin{bmatrix}1 & 2 & 3\\4 & 5 & 6\\7 & 8 & 9\end{bmatrix}$$

`mat1`^{T} ENTER

$$\begin{bmatrix}1 & 4 & 7\\2 & 5 & 8\\3 & 6 & 9\end{bmatrix}$$

`[a,b;c,d]→mat2` ENTER $\begin{bmatrix}a & b\\c & d\end{bmatrix}$

`mat2`^{T} ENTER $\begin{bmatrix}a & c\\b & d\end{bmatrix}$

`[1+`i`,2+`i`;3+`i`,4+`i`]→mat3` ENTER

$$\begin{bmatrix}1+i & 2+i\\3+i & 4+i\end{bmatrix}$$

`mat3`^{T} ENTER $\begin{bmatrix}1-i & 3-i\\2-i & 4-i\end{bmatrix}$

Table **CATALOG**

Table *Term1*[**,** *Term2*] [**,** *Var1*]

Erstellt aus den angegebenen Termen oder Funktionen eine Tabelle.

Die Terme der Tabelle können auch graphisch dargestellt werden. Terme, die mit den Befehlen **Table** oder **Graph** eingegeben werden, erhalten fortlaufende Funktionsnummern zugewiesen. Die Numerierung beginnt mit 1. Die Terme können verändert oder einzeln gelöscht werden, wenn man die Editierfunktionen verwendet, die bei angezeigter Tabelle mit F4 Header verfügbar sind. Die derzeit im Y= Editor ausgewählten Funktionen werden vorübergehend ignoriert.

Zum Löschen der Bildschirmanzeige von Funktionen, die mit **Table** oder **Graph** erzeugt wurden, können Sie den Befehl **ClrGraph** ausführen oder den Y= Editor aufrufen.

Wird der Parameter *Var* nicht angegeben, wird die im aktuellen Graphikmodus gültige unabhängige Variable benutzt. Einige gültige Varianten dieser Anweisung sind:

Funktions-Graphikmodus: **Table** *Term***,** *x*

Parameterdarstellung: **Table** *xTerm***,** *yTerm***,** *t*

Polar-Graphikmodus: **Table** *Term***,** θ

Hinweis: Der Befehl **Table** ist für 3D-, Sequence- und Diff Equations-Graphdarstellung nicht zulässig. Als Alternative können Sie **BldData** verwenden.

Im Funktions-Graphikmodus.

`Table 1.25x*cos(x)` ENTER

x	1		
0.	0.		
1.	.67538		
2.	-1.04		
3.	-3.712		
4.	-3.268		

`Table cos(time),time` ENTER

x	1	2	3
0.	0.	1.	
1.	.67538	.5403	
2.	-1.04	-.4161	
3.	-3.712	-.99	
4.	-3.268	-.6536	

tan()

[2nd] [TAN] **Taste** [TAN] **Taste**

tan(*Term1***)** ⇒ *Term*
tan(*Liste1***)** ⇒ *Liste*

tan(*Term1***)** gibt den Tangens des Parameters als Term zurück.

tan(*Liste1***)** gibt eine Liste zurück, die für jedes Element von *Liste1* den Tangens enthält.

Hinweis: Der Parameter wird gemäß der aktuellen Winkel-Moduseinstellung als Ordnung oder Radiant interpretiert. Sie können ° oder ʳ benutzen, um die Winkel-Moduseinstellung temporär zu ändern.

Im Degree-Modus für Winkel:

```
tan((π/4)ʳ) ENTER                    1
tan(45) ENTER                        1
tan({0,60,90}) ENTER
                          {0   √3   undef}
```

Im Radian-Modus:

```
tan(π/4) ENTER                       1
tan(45°) ENTER                       1
tan({π,π/3,-π,π/4}) ENTER
                          {0  √3  0  1}
```

---

tan(*quadrat_Matrix1***)** ⇒ *quadrat_Matrix*

Ergibt den Matrix-Tangens von *quadrat_Matrix1*. Dies ist *nicht* gleichbedeutend mit der Berechnung des Tangens jedes einzelnen Elements. Näheres zum Berechnungsverfahren finden Sie im Abschnitt **cos()**.

Quadrat_Matrix1 muß diagonalisierbar sein. Das Ergebnis enthält stets Fließkommazahlen.

Im Radian-Modus:

```
tan([1,5,3;4,2,1;6,-2,1]) ENTER
```

$$\begin{bmatrix} -28.291\ldots & 26.088\ldots & 11.114\ldots \\ 12.117\ldots & -7.835\ldots & -5.481\ldots \\ 36.818\ldots & -32.806\ldots & -10.459\ldots \end{bmatrix}$$

tan^{-1}()

[◆] [TAN^{-1}] **Taste** [2nd] [TAN^{-1}] **Taste**

tan^{-1}(*Term1***)** ⇒ *Term*
tan^{-1}(*Liste1***)** ⇒ *Liste*

tan^{-1}(*Term1***)** gibt als Term den Winkel zurück, dessen Tangens *Term1* ist.

tan^{-1}(*Liste1***)** gibt eine Liste zurück, die für jedes Element von *Liste1* den inversen Tangens enthält.

Hinweis: Das Ergebnis wird gemäß der aktuellen Winkel-Moduseinstellung entweder in Ordnung oder im Bogenmaß zurückgegeben.

Im Degree-Modus für Winkel:

```
tan⁻¹(1) ENTER                       45
```

Im Radian-Modus:

```
tan⁻¹({0,.2,.5}) ENTER
                          {0   .197…   .463…}
```

---

tan^{-1}(*quadrat_Matrix1***)** ⇒ *quadrat_Matrix*

Ergibt den inversen Matrix-Tangens von *quadrat_Matrix1*. Dies ist *nicht* gleichbedeutend mit der Berechnung des inversen Tangens jedes einzelnen Elements. Näheres zum Berechnungsverfahren finden Sie im Abschnitt **cos()**.

Quadrat_Matrix1 muß diagonalisierbar sein. Das Ergebnis enthält stets Fließkommazahlen.

Im Radian-Modus:

```
tan⁻¹([1,5,3;4,2,1;6,-2,1])
ENTER
```

$$\begin{bmatrix} -.083\ldots & 1.266\ldots & .622\ldots \\ .748\ldots & .630\ldots & -.070\ldots \\ 1.686\ldots & -1.182\ldots & .455\ldots \end{bmatrix}$$

tanh()

MATH/Hyperbolic-Menü

tanh(*Term1***)** ⇒ *Term*
tanh(*Liste1***)** ⇒ *Liste*

tanh(*Term1***)** gibt den Tangens hyperbolicus des Parameters als Term zurück.

tanh(*Liste***)** gibt eine Liste zurück, die für jedes Element von *Liste1* den Tangens hyperbolicus enthält.

```
tanh(1.2) ENTER                      .833…
tanh({0,1}) ENTER          {0   tanh(1)}
```

tanh(*Quadrat_Matrix1***)** ⇒ *Quadrat_Matrix*

Ergibt den Matrix-Tangens hyperbolicus von *quadrat_Matrix1*. Dies ist *nicht* gleichbedeutend mit der Berechnung des Tangens hyperbolicus jedes einzelnen Elements. Näheres zur Berechnungsmethode finden Sie im Abschnitt **cos()**.

Quadrat_Matrix1 muß diagonalisierbar sein. Das Ergebnis enthält stets Fließkommazahlen.

Im Radian-Modus:

```
tanh([1,5,3;4,2,1;6,⁻2,1])
ENTER
```

$$\begin{bmatrix} -.097\ldots & .933\ldots & .425\ldots \\ .488\ldots & .538\ldots & -.129\ldots \\ 1.282\ldots & -1.034\ldots & .428\ldots \end{bmatrix}$$

tanh⁻¹() MATH/Hyperbolic-Menü

tanh⁻¹(*Term1***)** ⇒ *Term*
tanh⁻¹(*Liste1***)** ⇒ *Liste*

tanh⁻¹(*Term1***)** gibt den inversen Tangens hyperbolicus des Parameters als Term zurück.

tanh⁻¹(*Liste1***)** gibt eine Liste zurück, die für jedes Element von *Liste1* den inversen Tangens hyperbolicus enthält.

Im Komplex-Formatmodus "Rectangular":

```
tanh⁻¹(0) ENTER                                    0
tanh⁻¹({1,2.1,3}) ENTER
```

$$\{\infty \quad .518\ldots - 1.570\ldots \cdot i \quad \frac{\ln(2)}{2} - \frac{\pi}{2} \cdot i\}$$

tanh⁻¹(*quadrat_Matrix1***)** ⇒ *quadrat_Matrix*

Ergibt den inversen Matrix-Tangens hyperbolicus von *quadrat_Matrix1*. Dies ist *nicht* gleichbedeutend mit der Berechnung des inversen Tangens hyperbolicus jedes einzelnen Elements. Näheres zur Berechnungsmethode finden Sie im Abschnitt **cos()**.

Quadrat_Matrix1 muß diagonalisierbar sein. Das Ergebnis enthält stets Fließkommazahlen.

Im Winkelmodus Radian und im Komplex-Formatmodus "Rectangular":

```
tanh⁻¹([1,5,3;4,2,1;6,⁻2,1])
ENTER
```

$$\begin{bmatrix} -.099\ldots+.164\ldots\cdot i & .267\ldots-1.490\ldots\cdot i & \ldots \\ -.087\ldots-.725\ldots\cdot i & .479\ldots-.947\ldots\cdot i & \ldots \\ .511\ldots-2.083\ldots\cdot i & -.878\ldots+1.790\ldots\cdot i & \ldots \end{bmatrix}$$

taylor() MATH/Calculus-Menü

taylor(*Term1***, *Var*, *Ordnung*[, *Punkt*])** ⇒ *Term*

Gibt ein Taylorpolynom zurück. Das Polynom enthält alle ganzzahligen Potenzen von (*var* - *point*) mit nichtverschwindenden Koeffizienten von *zero* bis *order*. **taylor()** gibt sich selbst zurück, wenn es keine endliche Potenzreihe dieses Ordnunges gibt oder negative oder Bruchexponenten erforderlich wären. Benutzen Sie Substitution und/oder die temporäre Multiplikation mit einer Potenz (*Var* - *Punkt*), um allgemeinere Potenzreihen zu ermitteln.

Punkt ist vorgegeben als Null und ist der Entwicklungspunkt.

```
taylor(e^(√(x)),x,2) ENTER
taylor(e^(t),t,4)|t=√(x) ENTER
```

$$\blacksquare\ \text{taylor}\left(e^{\sqrt{x}}, x, 2\right)$$
$$\text{taylor}\left(e^{\sqrt{x}}, x, 2, 0\right)$$
$$\blacksquare\ \text{taylor}\left(e^{t}, t, 4\right) \mid t = \sqrt{x}$$
$$\frac{x^2}{24} + \frac{x^{3/2}}{6} + \frac{x}{2} + \sqrt{x} + 1$$

```
taylor(1/(x*(x-1)),x,3) ENTER
```

$$\blacksquare\ \text{taylor}\left(\frac{1}{x\cdot(x-1)}, x, 3\right)$$
$$\text{taylor}\left(\frac{1}{x\cdot(x-1)}, x, 3, 0\right)$$

```
expand(taylor(x/(x*(x-1)),
x,4)/x,x)
ENTER
```

$$\blacksquare\ \text{expand}\left(\frac{\text{taylor}\left(\frac{x}{x\cdot(x-1)}, x\right.}{x}\right.$$
$$-x^3 - x^2 - x - \frac{1}{x} - 1$$

tCollect() MATH\Algebra\Trig-Menü

tCollect(*Term1***)** ⇒ *Term*

Gibt einen Term zurück, in dem Produkte und ganzzahlige Potenzen von Sinus und Cosinus in eine lineare Kombination von Sinus und Cosinus von Winkelsummen, Vielfachen von Winkeln und Winkeldifferenzen umgewandelt sind. Diese Transformation wandelt trigonometrische Polynome in eine lineare Kombination um.

In manchen Fällen führt **tCollect()** zum Erfolg, wo die vorgegebene trigonometrische Vereinfachung nicht zum Erfolg führt. **tCollect()** bewirkt in beinahe allen Fällen eine Umkehrung von Transformationen, die mit **tExpand()** vorgenommen wurden. Manchmal läßt sich ein Term vereinfachen, wenn man in getrennten Schritten **tExpand()** auf ein Ergebnis von **tCollect()** anwendet (oder umgekehrt).

```
tCollect((cos(α))^2) ENTER
                    cos(2·α) + 1
                    ------------
                          2

tCollect(sin(α)cos(β)) ENTER
                  sin(α−β)+sin(α+β)
                  -----------------
                          2
```

tExpand() MATH\Algebra\Trig-Menü

tExpand(*Term1***)** ⇒ *Term*

Gibt einen Term zurück, in dem Sinus und Cosinus von ganzzahligen Winkelvielfachen, Winkelsummen und Winkeldifferenzen entwickelt sind. Aufgrund der Identität $(\sin(x))^2+(\cos(x))^2=1$ sind viele äquivalente Ergebnisse möglich. Ein Ergebnis kann sich daher von einem in anderen Publikationen angegebenen unterscheiden.

In manchen Fällen führt **tExpand()** zum Erfolg, wo die vorgegebene trigonometrische Vereinfachung nicht zum Erfolg führt. **tExpand()** bewirkt in beinahe allen Fällen eine Umkehrung von Transformationen, die mit **tCollect()** vorgenommen wurden. Manchmal läßt sich ein Term vereinfachen, wenn man in getrennten Schritten **tCollect()** auf ein Ergebnis von **tExpand()** anwendet (oder umgekehrt).

Hinweis: Die Skalierung von π/180 im Winkelmodus "Ordnung" behindert die Erkennung entwickelbarer Formen durch **tExpand()**. Die besten Ergebnisse werden bei Benutzung von **tExpand()** im Radian-Modus erzielt.

```
tExpand(sin(3ϕ)) ENTER
      4·sin(ϕ)·(cos(ϕ))^2 − sin(ϕ)

tExpand(cos(α−β)) ENTER
    cos(α)·cos(β)+sin(α)·sin(β)
```

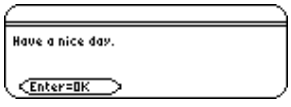
Text CATALOG

Text *AuffordString*

Zeigt das Dialogfeld *AuffordString* an.

Wird diese Anweisung in einem Block **Dialog...EndDlog** benutzt, wird *AuffordString* in diesem Dialogfeld angezeigt. Wird **Text** als eigenständige Anweisung benutzt, erzeugt sie ein Dialogfeld, um die Zeichenkette anzuzeigen.

```
Text "Have a nice day." ENTER
                          Done
```



Then

Siehe **If,** Seite 911.

timeCnv() CATALOG

timeCnv(*seconds***)** ⇒ *list*

Wandelt Sekunden in Zeiteinheiten um, die für Berechnungen leichter zu verwerten sind. Die Liste hat das Format {*Tage,Stunden,Minuten,Sekunden*}.

Hinweis: Siehe auch **checkTmr()** und **startTmr()**.

```
timeCnv(152442117)
                    {1764  9  1  57}
```

Title CATALOG

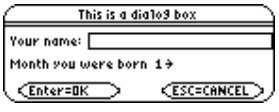
Title *TitelString*, [*Lbl*]

Erzeugt bei Verwendung eines **Toolbar-** oder **Custom**-Konstrukts oder in einem **Dialog...EndDlog**-Block den Titel eines Pulldown-Menüs oder eines Dialogfelds.

Hinweis: *Lbl* kann nur in einer **Toolbar-**Konstruktion benutzt werden. *Lbl* erlaubt dort die Verzweigung von einem Menüauswahlpunkt zu einer im Programm definierten Marke.

Programmsegment:

```
  ⋮
:Dialog
:Title    "This is a dialog
box"
:Request  "Your name",Str1
:Dropdown "Month you were
born",
  seq(string(i),i,1,12),Var1
:EndDlog
  ⋮
```



tmpCnv() CATALOG

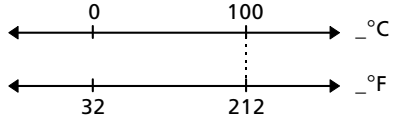
tmpCnv(*Term1*_ °*TempEinheit1*, _ °*TempEinheit2***)**
⇒ *Term* _ °*TempEinheit2*

Konvertiert einen durch *Term1* definierten Temperaturwert von einer Einheit in eine andere. Folgende Temperatureinheiten sind gültig:

_°C	Celsius
_°F	Fahrenheit
_°K	Kelvin
_°R	Rankine

└ ° wird mit 2nd [°] erzeugt.
wird mit ◆ [_] erzeugt.
_ wird mit 2nd [_]

100_°C wird zum Beispiel in 212_°F konvertiert:



Zur Konvertierung eines Temperaturbereichs verwenden Sie hingegen **ΔtmpCnv()**.

```
tmpCnv(100_°c,_°f) ENTER 212.•_°F

tmpCnv(32_°f,_°c) ENTER    0.•_°C

tmpCnv(0_°c,_°k) ENTER 273.15•_°K

tmpCnv(0_°f,_°r) ENTER 459.67•_°R
```

Hinweis: Drücken Sie,
2nd [UNITS]
◆ [UNITS]
um Temperatureinheiten aus einem Menü zu wählen.

Δ**tmpCnv()** **CATALOG**

Δ**tmpCnv(***Term1_ °TempEinheit1***, _ °***TempEinheit2***)**
⇒ *Term _ °TempEinheit2*

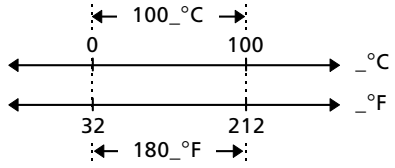
Konvertiert einen durch *Term1* definierten Temperaturbereich (Differenz zwischen zwei Temperaturwerten) von einer Einheit in eine andere. Folgende Temperatureinheiten sind gültig:

_°C	Celsius
_°F	Fahrenheit
_°K	Kelvin
_°R	Rankine

° wird mit [2nd] [°] erzeugt.
wird mit [◆] [_] erzeugt.
_ wird mit [2nd] [_]

1_°C und 1_°K haben denselben Absolutwert, ebenso wie 1_°F und 1_°R. 1_°C ist allerdings 9/5 so groß wie 1_°F.

Ein Bereich 100_°C (von 0_°C bis 100_°C) ist beispielsweise einem Bereich 180_°F äquivalent:



Zur Konvertierung eines Temperaturbereichs verwenden Sie hingegen Δ**tmpCnv()**.

Um Δ zu erhalten, drücken Sie [◆] [(] und dann [↑] [D] (oder [2nd] [CHAR] 1 5).

```
ΔtmpCnv(100_°c,_°f) ENTER
                      180.•_°F
ΔtmpCnv(180_°f,_°c) ENTER
                      100.•_°C
ΔtmpCnv(100_°c,_°k) ENTER
                      100.•_°K
ΔtmpCnv(100_°f,_°r) ENTER
                      100.•_°R
ΔtmpCnv(1_°c,_°f) ENTER   1.8•_°F
```

Hinweis: Drücken Sie
[2nd] [UNITS]
[◆] [UNITS]
um Temperatureinheiten aus einem Menü zu wählen.

Toolbar **CATALOG**

Toolbar
Block
EndTBar

Erzeugt eine Menüleiste.

Block kann eine einzelne Anweisung oder eine Serie von Anweisungen sein, die durch ":" getrennt sind. Als Anweisungen in *Block* sind Title und Item zulässig.

Zu jeden *item* muß gehört eine Marke. Title muß eine Marke haben, wenn der Block keine *item*-Anweisung enthält.

Programmsegment:

```
   ⋮
:Toolbar
:  Title "Examples"
:  Item "Trig", t
:  Item "Calc", c
:  Item "Stop", Pexit
:EndTbar
   ⋮
```

Hinweis: Bei Ausführung als Teil eines Programms erzeugt dieses Segment ein Menü mit drei Auswahlangeboten, von denen aus zu drei Stellen im Programm verzweigt werden kann.

Trace CATALOG

Trace

Zeichnet einen Smart Graph und setzt den Trace-Cursor auf die erste definierte Y= Funktion, an die vorab definierte Cursorposition oder an die Reset-Position, wenn Neuzeichnen erforderlich war.

Gestattet die Benutzung des Cursors sowie der meisten Tasten zum Ändern von Koordinatenwerten. Manche Tasten, etwa die Funktionstasten, APPS und MODE, sind während des Tracens nicht aktiv.

Hinweis: Drücken Sie zum Fortsetzen ENTER.

Try CATALOG

Try
Block1
Else
Block2
EndTry

Führt *Block1* aus, wenn kein Fehler auftritt. Bei Auftreten eines Fehlers in *Block1* wird die Programmausführung an *Block2* übergeben. Die Nummer des Fehlers wird in der Variablen errornum gespeichert, damit das Programm sie zur Fehlerbehandlung und ggf. zur Fehlerbehebung nutzen kann.

Block1 und *Block2* können einzelne Anweisungen oder Folgen von Anweisungen sein, die durch ":" getrennt sind.

Programmsegment:

```
  ⋮
:Try
:  NewFold(temp)
:  Else
:  ●Already exists
:  ClrErr
:EndTry
  ⋮
```

Hinweis: Siehe **ClrErr** und **PassErr**.

TwoVar MATH/Statistics-Menü

TwoVar *Liste1*, *Liste2*[, [*Liste3*] [, *Liste4*, *Liste5*]]

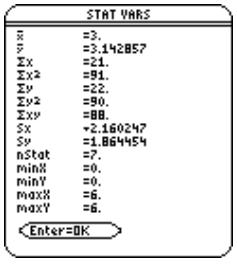
Berechnet die 2-Variablenstatistik und aktualisiert alle Statistiksystemvariablen.

Alle Listen außer *Liste5* müssen die gleiche Dimension besitzen.

Liste1 stellt die Liste der x-Werte dar.
Liste2 stellt die Liste der y-Werte dar.
Liste3 stellt die Angaben für die Häufigkeit dar.
Liste4 stellt die Klassencodes dar.
Liste5 stellt die Klassenliste dar.

Hinweis: *Liste1* bis einschl. *Liste4* müssen Variablenamen oder c1–c99 sein (Spalten in der letzten Datenvariablen, die im Daten/ Matrix-Editor angezeigt wurde). *Liste5* braucht kein Variablenname zu sein und kann keine Spalte c1–c99 sein.

```
{0,1,2,3,4,5,6}→L1 ENTER
                          {0 1 2 …}
{0,2,3,4,3,4,6}→L2 ENTER
                          {0 2 3 …}
TwoVar L1,L2 ENTER          Done
ShowStat ENTER
```



Umleitung

Siehe **#()**, Seite 990.

Unarchiv CATALOG

Unarchiv *Var1* [, *Var2*] [, *Var3*] ...

Überträgt die angegebenen Variablen vom Benutzer-Archivspeicher in den RAM-Speicher.

Auf eine archivierte Variable kann ebenso wie auf eine im RAM-Speicher vorliegende Variable zugegriffen werden. Sie können eine archivierte Variable allerdings weder löschen, umbenennen noch einen neuen Wert speichern, da sie automatisch gesperrt wird.

Zum Archivieren von Variablen dient **Archive**.

```
10→arctest ENTER                    10
Archive arctest ENTER               Done
5*arctest ENTER                     50
15→arctest ENTER
```

```
ESC
Unarchiv arctest ENTER              Done
15→arctest ENTER                    15
```

unitV() MATH/Matrix/Vector ops-Menü

unitV(*Vektor1***)** ⇒ *Vektor*

Gibt je nach der Form von *Vektor1* entweder einen Zeilen- oder einen Spalteneinheitsvektor zurück.

Vektor1 muß eine einzeilige oder eine einspaltige Matrix sein.

```
unitV([a,b,c]) ENTER
```

$$\left[\frac{a}{\sqrt{a^2+b^2+c^2}}\ \frac{b}{\sqrt{a^2+b^2+c^2}}\ \frac{c}{\sqrt{a^2+b^2+c^2}}\right]$$

```
unitV([1,2,1]) ENTER
```

$$\left[\frac{\sqrt{6}}{6}\ \frac{\sqrt{6}}{3}\ \frac{\sqrt{6}}{6}\right]$$

```
unitV([1;2;3]) ENTER
```

$$\begin{bmatrix}\frac{\sqrt{14}}{14}\\ \frac{\sqrt{14}}{7}\\ \frac{3\cdot\sqrt{14}}{14}\end{bmatrix}$$

Unlock CATALOG

Unlock *Var1*[, *Var2*][, *Var3*]...

Hebt die Sperrung der angegebenen Variablen auf.

Hinweis: Sie sperren Variablen mit dem Befehl **Lock**.

variance() MATH/Statistics-Menü

variance(*list*[, *freqlist*]**)** ⇒ *expression*

Ergibt die Varianz von *list*.

Jedes *freqlist*-Element gibt die Häufigkeit des entsprechenden Elements in *list* an.

Hinweis: *list* muß mindestens zwei Elemente enthalten.

```
variance({a,b,c}) ENTER
```

$$\frac{a^2-a\cdot(b+c)+b^2-b\cdot c+c^2}{3}$$

```
variance({1,2,5,-6,3,-2}) ENTER
                                    31/2
variance({1,3,5},{4,6,2}) ENTER
                                    68/33
```

---

variance(*matrix1*[, *freqmatrix*]**)** ⇒ *matrix*

Ergibt einen Reihenvektor mit der Varianz der einzelnen Spalten in *matrix1*.

Jedes *freqmatrix*-Element gibt die Häufigkeit des entsprechenden Elements in *matrix1* an.

Hinweis: *matrix1* muß mindestens zwei Zeilen enthalten.

```
variance([1,2,5;-3,0,1;
.5,.7,3]) ENTER   [4.75  1.03  4]
variance([-1.1,2.2;3.4,5.1;
-2.3,4.3],[6,3;2,4;5,1]) ENTER
                  [3.91731,2.08411]
```

when() CATALOG

when(*Bedingung***,** *true_Ergeb* [, *false_Ergeb*] [, *unbekannt_Ergeb*]**)** ⇒ *Term*

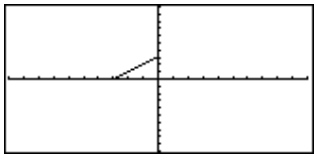
Gibt *true_Ergeb*, *false_Ergeb* oder *unbekannt_Ergeb* zurück, wenn *Bedingung* wahr, falsch bzw. unbekannt ist. Gibt die Eingabe zurück, wenn zu wenige Parameter angegeben werden.

Lassen Sie sowohl *false_Ergeb* als auch *unbekannt_Ergeb* weg, um einen Term nur für den Bereich zu bestimmen, in dem *Bedingung* wahr ist.

```
when(x<0,x+3)|x=5 ENTER
                  when(x<0,3+x)
```

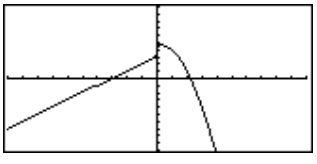
Geben Sie undef für *false_Ergeb* an, um einen Term zu bestimmen, der nur in einem Intervall graphisch dargestellt werden soll.

```
ClrGraph ENTER
Graph when(x≥-π and
x<0,x+3,undef) ENTER
```



Lassen Sie *unbekannt_Ergeb* weg, um einen Term aus zwei Stücken zu definieren.

```
Graph when(x<0,x+3,5-x^2) ENTER
```

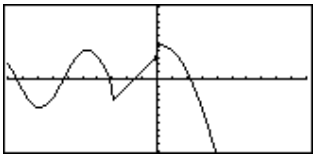


Benutzen Sie verschachtelte **when()** zum Definieren von Termen, die aus mehr als zwei Stücken bestehen.

HOME

◆ [CALC HOME]

```
ClrGraph ENTER                Done
Graph when(x<0,when(x<-π,
4*sin(x),2x+3),5-x^2) ENTER
```



when() ist auch zum Definieren rekursiver Funktionen nützlich.

```
when(n>0,n*factoral(n-1),1)
→factoral(n) ENTER            Done
factoral(3) ENTER               6
3! ENTER                        6
```

While CATALOG

While *Bedingung*
Block
EndWhile

Führt die in *Block* enthaltenen Anweisungen solange aus, wie *Bedingung* wahr ist.

Block kann eine einzelne Anweisung oder eine Serie von Anweisungen sein, die durch ":" getrennt sind.

Programmsegment:

```
  ⋮
:1→i
:0→temp
:While i<=20
:  temp+1/i→temp
:  i+1→i
:EndWhile
:Disp "sum of reciprocals up
to                      20",temp
  ⋮
```

"With"

Siehe |, Seite 994.

xor — MATH/Test-Menü

Boolescher Term1 **xor** *Boolescher Term2* ⇒ *Boolescher Term*

Gibt "true" zurück, wenn *Boolescher Term1* true und *Boolescher Term2* "false" ergeben oder umgekehrt. Gibt "false" zurück, wenn *Boolescher Term1* und *Boolescher Term2* beide true oder beide falsch sind. Gibt einen vereinfachten Booleschen Term zurück, wenn einer der beiden Booleschen Ausgangsterme nicht zu wahr oder falsch ausgewertet werden kann.

Hinweis: Siehe **or**.

```
true xor true ENTER          false
(5>3) xor (3>5) ENTER        true
```

Ganze_Zahl1 **xor** *Ganze_Zahl2* ⇒ *Ganze_Zahl*

Vergleicht zwei reelle ganze Zahlen mit Hilfe einer **xor**-Operation Bit für Bit. Intern werden beide ganzen Zahlen in 32-Bit-Dualzahlen mit Vorzeichen konvertiert. Beim Vergleich der sich entsprechenden Bits ist das Ergebnis 1, wenn eines der Bits (nicht aber beide) 1 ist; das Ergebnis ist 0, wenn entweder beide Bits 0 oder beide Bits 1 sind. Der zurückgegebene Wert stellt die Bit-Ergebnisse dar und wird im jeweiligen Base-Modus angezeigt.

Sie können die ganzen Zahlen mit jeder Basis eingeben. Für eine binäre oder hexadezimale Eingabe muß das Präfix 0b bzw. 0h verwendet werden. Ohne Präfix werden die ganzen Zahlen als dezimal behandelt (Grundzahl 10).

Geben Sie eine dezimale ganze Zahl ein, die für eine 32-Bit-Dualform mit Vorzeichen zu groß ist, dann wird eine eine symmetrische Modulo-Operation ausgeführt, um den Wert in den erforderlichen Bereich zu bringen.

Hinweis: Siehe **or**.

Im Hex Modus:

```
0h7AC36 xor 0h3D5F ENTER 0h79169
```

└ **Wichtig:** Null, nicht Buchstabe O.

Im Modus Bin base:

```
0b100101 xor 0b100 ENTER 0b100001
```

Hinweis: Eine binäre Eingabe kann bis zu 32 Stellen haben (das Präfix 0b wird nicht mitgezählt). Eine hexadezimale Eingabe kann bis zu 8 Stellen aufweisen.

XorPic — CATALOG

XorPic *picVar*[, *Zeile*] [, *Spalte*]

Zeigt das in *picVar* gespeicherte Bild auf dem aktuellen Graphikbildschirm an.

Dabei wird für jedes Pixel eine **xor**-Operation ausgeführt. Nur Pixel, die nur auf dem Bildschirm oder nur im neuen Bild "ein" sind, werden eingeschaltet. Pixel, die in beiden Bildern "ein" sind, werden mit dieser Anweisung ausgeschaltet.

picVar muß vom Datentyp Bild (picture) sein.

Zeile und *Spalte* geben (wahlfrei) die Pixel-Koordinaten für die linke obere Ecke des Bilds an. Vorgegeben sind 0, 0.

zeros() — MATH/Algebra-Menü

zeros(*Term***,** *Var***)** ⇒ *Liste*

Gibt eine Liste möglicher reeller Werte für *Var* zurück, die *Term*=0 ergeben. **zeros()** tut dies durch Berechnung von **exp▸list(solve(***Term*=0,*Var***))**.

```
zeros(a*x^2+b*x+c,x) ENTER
```

$$\left\{\frac{-\left(\sqrt{b^2-4\cdot a\cdot c}+b\right)}{2\cdot a},\ \frac{\sqrt{b^2-4\cdot a\cdot c}-b}{2\cdot a}\right\}$$

```
a*x^2+b*x+c|x=ans(1)[2] ENTER   0
```

Für manche Zwecke ist die Ergebnisform von **zeros()** günstiger als die von **solve()**. Allerdings kann die Ergebnisform von **zeros()** folgende Lösungen nicht ausdrücken: implizite Lösungen; Lösungen, für die Ungleichungen erforderlich sind sowie Lösungen, die nicht *Var* betreffen.

Hinweis: Siehe auch **cSolve()**, **cZeros()** und **solve()**.

```
exact(zeros(a*(e^(x)+x)
  (sign (x)-1),x)) ENTER          {}

exact(solve(a*(e^(x)+x)
  (sign (x)-1)=0,x)) ENTER
          e^x+x=0 or x>0 or a=0
```

---

zeros({*Term1***, *Term2***}, {***VarOderSchätzwert1***, *VarOderSchätzwert2* [, …]}) ⇒ *Matrix***

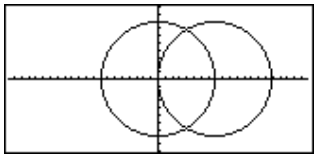
Gibt mögliche reelle Nullen für die simultanen algebraischen *Terme* zurück, wobei jeder *VarOderSchätzwert* einen gesuchten unbekannten Wert angibt.

Sie haben die Option, einen Ausgangsschätzwert für eine Variable anzugeben. *VarOderSchätzwert* muß immer folgende Form haben:

Variable
– oder –
Variable = reelle oder nicht-reelle Zahl

Beispiel: x ist gültig, und x=3 ebenfalls.

Wenn alle Terme Polynome sind und Sie KEINE Anfangsschätzwerte angeben, dann verwendet **zeros()** das lexikalische Gröbner/Buchbergersche Eliminationsverfahren beim Versuch, **alle** reellen Nullstellen zu bestimmen.



Betrachten wir z.B. einen Kreis mit dem Radius r und dem Ursprung als Mittelpunkt und einen weiteren Kreis mit Radius r und dem Schnittpunkt des ersten Kreises mit der positiven X-Achse als Mittelpunkt. Verwenden Sie **zeros()** zur Bestimmung der Schnittpunkte.

```
zeros({x^2+y^2-r^2,
(x-r)^2+y^2-r^2},{x,y}) ENTER
```

$$\begin{bmatrix} \frac{r}{2} & \frac{\sqrt{3}\cdot r}{2} \\ \frac{r}{2} & \frac{-\sqrt{3}\cdot r}{2} \end{bmatrix}$$

Wie in nebenstehendem Beispiel (durch r) demonstriert, können simultane *polynomische* Terme zusätzliche Variablen ohne Wert aufweisen, die aber gegebene numerische Werte darstellen, welche später eingesetzt werden können.

Jede Zeile in der sich ergebenden Matrix stellt eine alternative Nullstelle dar, wobei die Komponenten in derselben Reihenfolge wie in der *VarOderSchätzwert*-Liste angeordnet sind. Um eine Zeile zu extrahieren ist die Matrix nach [*Zeile*] zu indexieren.

Zeile 2 extrahieren:

```
ans(1)[2] ENTER
```

$$\begin{bmatrix} \frac{r}{2} & \frac{-\sqrt{3}\cdot r}{2} \end{bmatrix}$$

Sie können auch (oder statt dessen) Unbekannte angeben, die in den Termen nicht erscheinen. Geben Sie zum Beispiel z als eine Unbekannte an, um das vorangehende Beispiel auf zwei parallele sich schneidende Zylinder mit dem Radius r auszudehnen. Die Zylinder-Nullstellen verdeutlichen, daß Nullstellenfamilien "beliebige" Konstanten der Form *@k* enthalten können, wobei *k* ein ganzzahliger Index im Bereich 1 bis 255 ist. Der Index wird wieder auf 1 zurückgesetzt, wenn Sie **ClrHome** oder F1 8:Clear Home verwenden.

```
zeros({x^2+y^2-r^2,
(x-r)^2+y^2-r^2},{x,y,z})
ENTER
```

$$\begin{bmatrix} \frac{r}{2} & \frac{\sqrt{3}\cdot r}{2} & @1 \\ \frac{r}{2} & \frac{-\sqrt{3}\cdot r}{2} & @1 \end{bmatrix}$$

Bei Systemen von Polynomen kann die Berechnungsdauer oder Speicherbelastung stark von der Reihenfolge abhängen, in welcher Sie die Unbekannten auflisten. Übersteigt Ihre erste Wahl

die Speicherkapazität oder Ihre Geduld, versuchen Sie, die Variablen in den Termen und/oder der *VarOderSchätzwert*-Liste umzuordnen.

Wenn Sie keine Schätzwerte angeben und ein Term in einer Variablen kein Polynom ist, aber alle Terme in ihren Unbekannten linear sind, so verwendet **zeros()** das Gaußsche Eliminationsverfahren beim Versuch, alle reellen Nullstellen zu bestimmen.

```
zeros({x+e^(z)*y-1,x-y-sin(z)}
,{x,y}) ENTER
```

$$\left[\frac{e^z \cdot \sin(z)+1}{e^z+1} \quad \frac{-(\sin(z)-1)}{e^z+1}\right]$$

Wenn ein System weder in all seinen Variablen polynomisch noch in seinen Unbekannten linear ist, dann bestimmt **zeros()** mindestens eine Nullstelle anhand eines iterativen näherungsweisen Verfahrens. Hierzu muß die Anzahl der Unbekannten gleich der Anzahl der Terme sein, und alle anderen Variablen in den Termen müssen zu Zahlen vereinfachbar sein.

```
zeros({e^(z)*y-1,-y-sin(z)},
{y,z}) ENTER
```

```
[.041…  3.183…]
```

Jede Unbekannte beginnt bei dem entsprechenden geschätzten Wert, falls vorhanden; ansonsten beginnt sie bei 0,0.

Suchen Sie anhand von Schätzwerten nach einzelnen zusätzlichen Nullstellen. Für Konvergenz sollte eine Schätzung ziemlich nahe bei einer Nullstelle liegen.

```
zeros({e^(z)*y-1,-y-sin(z)},
{y,z=2π}) ENTER
```

```
[.001…  6.281…]
```

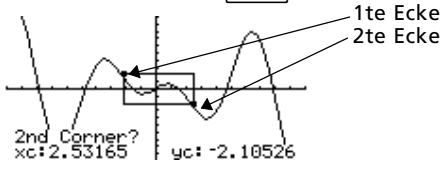
ZoomBox CATALOG

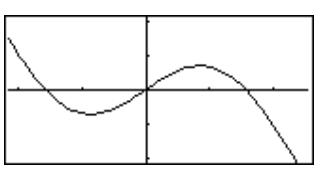
ZoomBox

Öffnet den Graphikbildschirm, damit Sie ein neues Ansichtfenster definieren können. Anschließend wird das Ansichtfenster aktualisiert.

Im Funktions-Graphikmodus:

```
1.25x*cos(x)→y1(x) ENTER      Done
ZoomStd:ZoomBox ENTER
```





Das Anzeige nachdem Sie das neue Fenster eingerichtet und durch das zweite Drücken von ENTER aktiviert haben.

ZoomData CATALOG

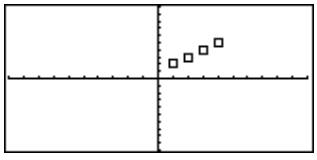
ZoomData

Paßt die Fenstereinstellungen auf der Grundlage der derzeit definierten Plots (und der Daten) so an, daß alle Statistikdatenpunkte erfaßt werden und zeigt den Graphikbildschirm an.

Hinweis: Paßt ymin und ymax nicht für Histogramme an.

Im Funktions-Graphikmodus:

```
{1,2,3,4}→L1 ENTER          {1 2 3 4}
{2,3,4,5}→L2 ENTER          {2 3 4 5}
newPlot 1,1,L1,L2 ENTER          Done
ZoomStd ENTER
```



HOME

◆ [CALC HOME]

```
ZoomData ENTER
```

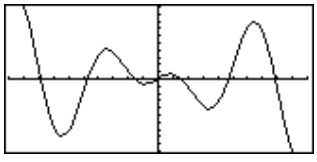
ZoomDec CATALOG

ZoomDec

Paßt das Ansichtfenster so an, daß Δx und $\Delta y = 0{,}1$ und zeigt dann den Graphikbildschirm mit dem Ursprung in Bildschirmmitte an.

Im Funktions-Graphikmodus:

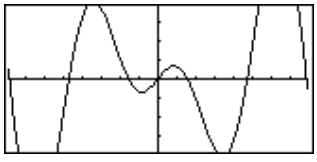
```
1.25x*cos(x)→y1(x) ENTER         Done
ZoomStd ENTER
```



HOME

◆ [CALC HOME]

```
ZoomDec ENTER
```



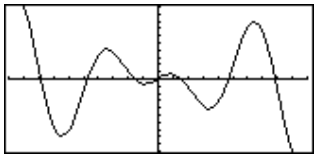
ZoomFit CATALOG

ZoomFit

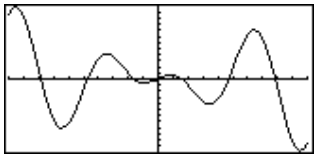
Öffnet den Graphikbildschirm und berechnet die erforderliche Fenstergröße für die abhängigen Variablen so, daß für die aktuellen Einstellungen der unabhängigen Variablen das gesamte Bild angezeigt wird.

Im Funktions-Graphikmodus:

`1.25x*cos(x)→y1(x)` ENTER Done
`ZoomStd` ENTER



HOME
[◆] [CALC HOME]
`ZoomFit` ENTER



ZoomIn CATALOG

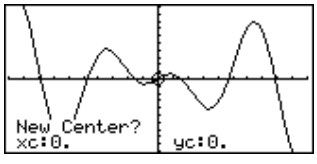
ZoomIn

Öffnet den Graphikbildschirm, läßt Sie ein Zentrum für eine Vergrößerung einstellen und aktualisiert dann das Ansichtfenster.

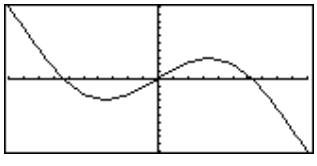
Der VergrößerungsOrdnung hängt von den Zoom-Faktoren xFact und yFact ab. Im 3D-Graphikmodus hängt der Vergrößerungs-Ordnung von xFact, yFact und zFact ab.

Im Funktions-Graphikmodus:

`1.25x*cos(x)→y1(x)` ENTER Done
`ZoomStd:ZoomIn` ENTER



ENTER



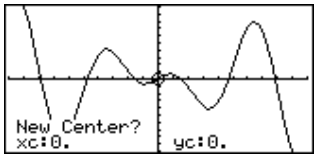
ZoomInt CATALOG

ZoomInt

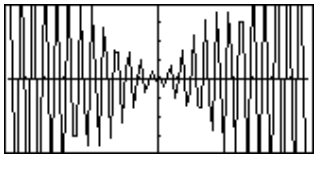
Öffnet den Graphikbildschirm, läßt Sie ein Zentrum für eine Vergrößerung einstellen und paßt die Fenstereinstellungen so an, daß jedes Pixel in alle Richtungen eine ganze Zahl ist.

Im Funktions-Graphikmodus:

`1.25x*cos(x)→y1(x)` ENTER Done
`ZoomStd:ZoomInt` ENTER



ENTER



ZoomOut CATALOG

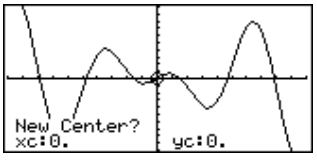
ZoomOut

Öffnet den Graphikbildschirm, läßt Sie ein Zentrum für eine Verkleinerung einstellen und aktualisiert das Ansichtfenster.

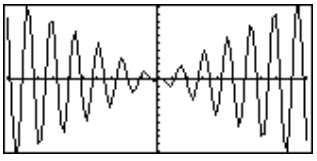
Der VerkleinerungsOrdnung hängt von den Zoom-Faktoren xFact und yFact ab. Im 3D-Graphikmodus hängt der VerkleinerungsOrdnung von xFact, yFact und zFact ab.

Im Funktions-Graphikmodus:

`1.25x*cos(x)→y1(x)` ENTER `Done`
`ZoomStd:ZoomOut` ENTER



ENTER



ZoomPrev CATALOG

ZoomPrev

Öffnet den Graphikbildschirm und aktualisiert das Ansichtfenster mit den Einstellungen, die vor der letzten Vergrößerung/Verkleinerung aktiv waren.

ZoomRcl CATALOG

ZoomRcl

Öffnet den Graphikbildschirm und aktualisiert das Ansichtfenster mit den Einstellungen, die mit der Anweisung **ZoomSto** gespeichert wurden.

ZoomSqr CATALOG

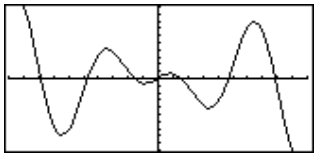
ZoomSqr

Öffnet den Graphikbildschirm, aktualisiert die x- oder y-Einstellungen des Fensters so, daß jedes Pixel eine gleiche Breite und Höhe im Koordinatensystem darstellt, und aktualisiert anschließend das Ansichtfenster selbst.

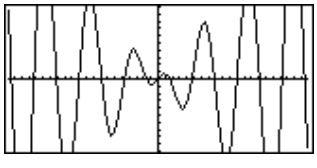
Im Modus 3D Graph verlängert **ZoomSqr** die kürzere der beiden Achsen auf die gleiche Länge wie die längere Achse.

Im Funktions-Graphikmodus:

`1.25x*cos(x)→y1(x)` ENTER `Done`
`ZoomStd` ENTER



HOME
`ZoomSqr` ENTER



ZoomStd CATALOG

ZoomStd

Setzt die Fenstervariablen auf die folgenden Standardwerte und aktualisiert anschließend das Ansichtfenster.

Funktionsgraphen:
x: [⁻10, 10, 1], y: [⁻10, 10, 1] and xres=2

Parameterdarstellung:
t: [0, 2π, π/24], x: [⁻10, 10, 1], y:[⁻10, 10, 1]

Polar-Graphen:
θ: [0, 2π, π/24], x: [⁻10, 10, 1], y: [⁻10, 10, 1]

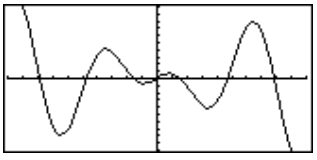
Folgen-Graphen:
nmin=1, nmax=10, plotStrt=1, plotStep=1,
x: [⁻10, 10, 1], y: [⁻10, 10, 1]

3D-Graphen:
eyeθ°=20, eyeϕ°=70, eyeψ°=0
x: [⁻10, 10, 14], y: [⁻10, 10, 14],
z: [⁻10, 10], ncontour=5

DGL-Graphen:
t: [0, 10, .1, 0], x: [⁻1, 10, 1], y: [⁻10, 10, 1],
ncurves=0, Estep=1, diftol=.001, fldres=20,
dtime=0

Im Funktions-Graphikmodus:

`1.25x*cos(x)→y1(x)` ENTER Done
`ZoomStd` ENTER



ZoomSto CATALOG

ZoomSto

Speichert die aktuellen Fenstereinstellungen im Zoom-Speicher. Sie können die Einstellungen mit **ZoomRcl** wiederherstellen.

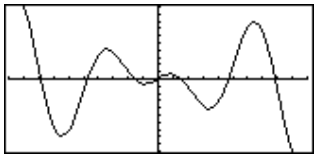
ZoomTrig CATALOG

ZoomTrig

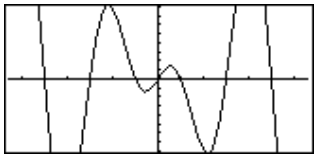
Öffnet den Graphikbildschirm, setzt Δx auf π/24 und xscl auf π/2, zentriert den Ursprung, setzt die y-Einstellungen auf [⁻4, 4, .5] und aktualisiert das Ansichtfenster.

Im Funktions-Graphikmodus:

`1.25x*cos(x)→y1(x)` ENTER Done
`ZoomStd` ENTER



HOME
◆ [CALC HOME]
`ZoomTrig` ENTER



+ (Addition) + -Taste

Term1 + *Term2* ⇒ *Term*

Gibt die Summe von *Term1* und *Term2* zurück.

`56` ENTER 56
`ans(1)+4` ENTER 60
`ans(1)+4` ENTER 64
`ans(1)+4` ENTER 68
`ans(1)+4` ENTER 72

Liste1 + *Liste2* ⇒ *Liste*
Matrix1 + *Matrix2* ⇒ *Matrix*

Gibt eine Liste (bzw. eine Matrix) zurück, die die Summen der entsprechenden Elemente von *Liste1* plus *Liste2* (bzw. *Matrix1* plus *Matrix2*) enthält.

Die Parameter müssen die gleiche Dimension besitzen.

```
{22,π,π/2}→L1 ENTER        {22 π π/2}
{10,5,π/2}→L2 ENTER        {10 5 π/2}
L1+L2 ENTER                {32 π+5 π}
ans(1)+{π,-5,-π} ENTER
                           {π+32  π  0}
[a,b;c,d]+[1,0;0,1] ENTER
                           [ a+1  b  ]
                           [ c  d+1  ]
```

Term + *Liste1* ⇒ *Liste*
Liste1 + *Term* ⇒ *Liste*

Gibt eine Liste zurück, die die Summen von *Term* plus jedem Element der *Liste1* enthält.

```
15+{10,15,20} ENTER        {25 30 35}
{10,15,20}+15 ENTER        {25 30 35}
```

Term + *Matrix1* ⇒ *Matrix*
Matrix1 + *Term* ⇒ *Matrix*

Gibt eine Matrix zurück, in der *Term* zu jedem Element der Diagonalen von *Matrix1* addiert ist. *Matrix1* muß eine quadratische Matrix sein.

Hinweis: Verwenden Sie **.+** (Skalar-Plus, Punkt-Plus) zum Addieren eines Terms zu jedem Element.

```
20+[1,2;3,4] ENTER
                           [21  2 ]
                           [3  24 ]
```

− (Subtrak.) **−-Taste**

Term1 − *Term2* ⇒ *Term*

Gibt *Term1* minus *Term2* zurück.

```
6−2 ENTER                  4
π−π/6 ENTER                5·π
                           ―――
                            6
```

Liste1 − *Liste2* ⇒ *Liste*
Matrix1 − *Matrix2* ⇒ *Matrix*

Subtrahiert die einzelnen Elemente aus *Liste2* (oder *Matrix2*) von denen in *Liste1* (oder *Matrix1*) und gibt die Ergebnisse zurück.

Die Parameter müssen die gleiche Dimension besitzen.

```
{22,π,π/2}−{10,5,π/2} ENTER
                           {12 π−5 0}
[3,4]−[1,2] ENTER          [2 2]
```

Term − *Liste1* ⇒ *Liste*
Liste1 − *Term* ⇒ *Liste*

Subtrahiert jedes Element der *Liste1* von *Term* oder subtrahiert *Term* von jedem Element der *Liste1* und gibt eine Liste der Ergebnisse zurück.

```
15−{10,15,20} ENTER        {5 0 -5}
{10,15,20}−15 ENTER        {-5 0 5}
```

Term − *Matrix1* ⇒ *Matrix*
Matrix1 − *Term* ⇒ *Matrix*

Term − *Matrix1* gibt eine Matrix zurück, die *Term* multipliziert mit der Einheitsmatrix minus *Matrix1* ist. *Matrix1* muß eine quadratische Matrix sein.

Matrix1 − *Term* gibt eine Matrix zurück, die *Term* multipliziert mit der Einheitsmatrix subtrahiert von *Matrix1* ist. *Matrix1* muß eine quadratische Matrix sein.

Hinweis: Verwenden Sie **.−** (Skalar-Minus, Punkt-Minus) zum Subtrahieren eines Terms von jedem Element.

```
20−[1,2;3,4] ENTER
                           [19  -2 ]
                           [-3  16 ]
```

∗ (Multiplik.) ⊠-Taste

Term1 ∗ *Term2* ⇒ *Term*

Gibt das Produkt von *Term1* und *Term2* zurück.

2∗3.45 ENTER → 6.9

x∗y∗x ENTER → $x^2 \cdot y$

Liste1 ∗ *Liste2* ⇒ *Liste*

Gibt eine Liste zurück, die die Produkte der entsprechenden Elemente aus *Liste1* und *Liste2* enthält.

Die Listen müssen die gleiche Dimension besitzen.

{1.0,2,3}∗{4,5,6} ENTER → {4. 10 18}

{2/a,3/2}∗{a²,b/3} ENTER → $\{2 \cdot a \quad \frac{b}{2}\}$

Matrix1 ∗ *Matrix2* ⇒ *Matrix*

Gibt das Matrizenprodukt von *Matrix1* und *Matrix2* zurück.

Die Zeilenanzahl von *Matrix1* muß gleich der Spaltenanzahl von *Matrix2* sein.

[1,2,3;4,5,6]∗[a,d;b,e;c,f] ENTER

$$\begin{bmatrix}1 & 2 & 3\\4 & 5 & 6\end{bmatrix}\cdot\begin{bmatrix}a & d\\b & e\\c & f\end{bmatrix}$$
$$\begin{bmatrix}a+2\cdot b+3\cdot c & d+2\cdot e+3\\4\cdot a+5\cdot b+6\cdot c & 4\cdot d+5\cdot e\end{bmatrix}$$

Term ∗ *Liste1* ⇒ *Liste*
Liste1 ∗ *Term* ⇒ *Liste*

Gibt eine Liste zurück, die die Produkte von *Term* und jedem Element der *Liste1* enthält.

π∗{4,5,6} ENTER → {4·π 5·π 6·π}

Term ∗ *Matrix1* ⇒ *Matrix*
Matrix1 ∗ *Term* ⇒ *Matrix*

Gibt eine Matrix zurück, die die Produkte von *Term* und jedem Element der *Matrix1* enthält.

Hinweis: Verwenden Sie **.∗** (Skalar-Multiplikation, Punkt-Multiplikation) zum Multiplizieren eines Terms mit jedem Element.

[1,2;3,4]∗.01 ENTER → $\begin{bmatrix}.01 & .02\\.03 & .04\end{bmatrix}$

λ∗identity(3) ENTER → $\begin{bmatrix}\lambda & 0 & 0\\0 & \lambda & 0\\0 & 0 & \lambda\end{bmatrix}$

/ (Division) ⊡-Taste

Term1 / *Term2* ⇒ *Term*

Gibt *Term1* dividiert durch *Term2* zurück.

2/3.45 ENTER → .57971

x^3/x ENTER → x^2

Liste1 / *Liste2* ⇒ *Liste*

Gibt eine Liste der Elemente von *Liste1* dividiert durch *Liste2* zurück.

Die Listen müssen die gleiche Dimension besitzen.

{1.0,2,3}/{4,5,6} ENTER → {.25 2/5 1/2}

Term / *Liste1* ⇒ *Liste*
Liste1 / *Term* ⇒ *Liste*

Gibt eine Liste zurück, die *Term* dividiert durch die Elemente von *Liste1* bzw. *Liste1* dividiert durch *Term* enthält.

a/{3,a,√(a)} ENTER → $\left\{\frac{a}{3} \quad 1 \quad \sqrt{a}\right\}$

{a,b,c}/(a∗b∗c) ENTER → $\{\frac{1}{b\cdot c} \quad \frac{1}{a\cdot c} \quad \frac{1}{a\cdot b}\}$

Matrix1 / *Term* ⇒ *Matrix*

Gibt eine Liste zurück, die die Quotienten *Matrix1*/*Term* enthält.

Hinweis: Verwenden Sie **./** (Skalar-Division, Punkt-Division) zum Dividieren eines Terms durch jedes Element.

[a,b,c]/(a∗b∗c) ENTER → $[\frac{1}{b\cdot c} \quad \frac{1}{a\cdot c} \quad \frac{1}{a\cdot b}]$

^ (Potenz) **-Taste**

Term1 **^** *Term2* ⇒ *Term*
Liste1 **^** *Liste2* ⇒ *Liste*

Gibt den ersten Parameter hoch dem zweiten Parameter zurück.

Bei einer Liste wird jedes Element aus *Liste1* hoch dem entsprechenden Element aus *Liste2* zurückgegeben.

Im reellen Bereich benutzen Bruchpotenzen mit gekürztem ungeradem Nenner den reellen statt den Hauptzeig im komplexen Modus.

```
4^2 ENTER                                  16
{a,2,c}^{1,b,3} ENTER                      {a  2^b  c^3}
```

Term **^** *Liste1* ⇒ *Liste*

Gibt *Term* hoch den Elementen von *Liste1* zurück.

```
p^{a,2,-3} ENTER                           {p^a  p^2  1/p^3}
```

Liste1 **^** *Term* ⇒ *Liste*

Gibt die Elemente von *Liste1* hoch *Term* zurück.

```
{1,2,3,4}^-2 ENTER
                                           {1  1/4  1/9  1/16}
```

Quadratische_Matrix1 **^** *Ganze_Zahl* ⇒ *Matrix*

Gibt *Quadratische_Matrix1* hoch *Ganze_Zahl* zurück.

Quadratische_Matrix1 muß eine quadratische Matrix sein.

Ist *Ganze_Zahl* = ⁻1, wird die inverse Matrix berechnet.
Ist *Ganze_Zahl* < ⁻1, wird die inverse Matrix hoch der entsprechenden positiven Zahl berechnet.

```
[1,2;3,4]^2 ENTER
[1,2;3,4]^-1 ENTER
[1,2;3,4]^-2 ENTER
```

$$\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix}^2 \qquad \begin{bmatrix}7 & 10\\ 15 & 22\end{bmatrix}$$

$$\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix}^{-1} \qquad \begin{bmatrix}-2 & 1\\ 3/2 & -1/2\end{bmatrix}$$

$$\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix}^{-2} \qquad \begin{bmatrix}11/2 & -5/2\\ -15/4 & 7/4\end{bmatrix}$$

.+ (Pkt.-Add.) **.** **+** **-Tasten**

Matrix1 **.+** *Matrix2* ⇒ *Matrix*
Term **.+** *Matrix1* ⇒ *Matrix*

Matrix1 **.+** *Matrix2* gibt eine Matrix zurück, die Summe jedes Elementpaars von *Matrix1* und *Matrix2* ist.

Term **.+** *Matrix1* gibt eine Matrix zurück, die die Summe von *Term* und jedem Element von *Matrix1* ist.

```
[a,2;b,3].+[c,4;5,d] ENTER
x.+[c,4;5,d] ENTER
```

$$\begin{bmatrix}a & 2\\ b & 3\end{bmatrix} .+ \begin{bmatrix}c & 4\\ 5 & d\end{bmatrix} \qquad \begin{bmatrix}a+c & 6\\ b+5 & d+3\end{bmatrix}$$

$$x .+ \begin{bmatrix}c & 4\\ 5 & d\end{bmatrix} \qquad \begin{bmatrix}x+c & x+4\\ x+5 & x+d\end{bmatrix}$$

.− (Pkt.-Sub.) **.** **−** **-Tasten**

Matrix1 **.−** *Matrix2* ⇒ *Matrix*
Term **.−** *Matrix1* ⇒ *Matrix*

Matrix1 **.−** *Matrix2* gibt eine Matrix zurück, die die Differenz jedes Elementpaars von *Matrix1* und *Matrix2* ist.

Term **.−** *Matrix1* gibt eine Matrix zurück, die die Differenz von *Term* und jedem Element von *Matrix1* ist.

```
[a,2;b,3].-[c,4;d,5] ENTER
x.-[c,4;d,5] ENTER
```

$$\begin{bmatrix}a & 2\\ b & 3\end{bmatrix} .- \begin{bmatrix}c & 4\\ d & 5\end{bmatrix} \qquad \begin{bmatrix}a-c & -2\\ b-d & -2\end{bmatrix}$$

$$x .- \begin{bmatrix}c & 4\\ d & 5\end{bmatrix} \qquad \begin{bmatrix}x-c & x-4\\ x-d & x-5\end{bmatrix}$$

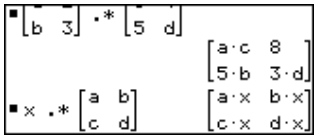
.* (Pkt.-Mult.) [.][×]-Tasten

Matrix1 .* *Matrix2* ⇒ *Matrix*
Term .* *Matrix1* ⇒ *Matrix*

Matrix1 .* *Matrix2* gibt eine Matrix zurück, die das Produkt jedes Elementpaars von *Matrix1* und *Matrix2* ist.

Term .* *Matrix1* gibt eine Matrix zurück, die das Produkt von *Term* und jedem Element von *Matrix1* ist.

```
[a,2;b,3].*[c,4;5,d] ENTER
x.*[a,b;c,d] ENTER
```



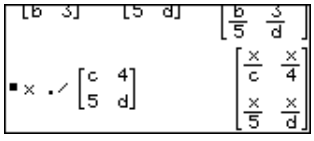
./ (Pkt.-Div.) [.][÷]-Tasten

Matrix1 ./ *Matrix2* ⇒ *Matrix*
Term ./ *Matrix1* ⇒ *Matrix*

Matrix1 ./ *Matrix2* gibt eine Matrix zurück, die der Quotient jedes Elementpaars von *Matrix1* und *Matrix2* ist.

Term ./ *Matrix1* gibt eine Matrix zurück, die der Quotient von *Term* und jedem Element von *Matrix1* ist.

```
[a,2;b,3]./[c,4;5,d] ENTER
x./[c,4;5,d] ENTER
```



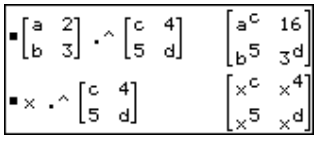
.^ (Pkt.-Potenz) [.][^]-Tasten

Matrix1 .^ *Matrix2* ⇒ *Matrix*
Term .^ *Matrix1* ⇒ *Matrix*

Matrix1 .^ *Matrix2* gibt eine Matrix zurück, in der jedes Element aus *Matrix2* Exponent des entsprechenden Elements aus *Matrix1* ist.

Term .^ *Matrix1* gibt eine Matrix zurück, in der jedes Element aus *Matrix1* Exponent von *Term* ist.

```
[a,2;b,3].^[c,4;5,d] ENTER
x.^[c,4;5,d] ENTER
```



− (Negation) [(-)]-Taste

−*Term1* ⇒ *Term*
− *Liste1* ⇒ *Liste*
− *Matrix1* ⇒ *Matrix*

Gibt die Negation des Parameters zurück.

Bei einer Liste oder Matrix werden alle Elemente negiert zurückgegeben.

Ist *Term1* eine binäre oder hexadezimale Ganzzahl, ergibt die Negation das Zweierkomplement.

```
-2.43 ENTER                                  -2.43
-{-1,0.4,1.2E19} ENTER
                                  {1  -.4  -1.2E19}
-a*-b ENTER                                  a·b
```

Im Modus Bin base:

```
0b100101▸dec ENTER                            37
```

└ **Wichtig:** Null, nicht Buchstabe O.

```
-0b100101 ENTER
 0b11111111111111111111111111011011
ans(1)▸dec ENTER                             -37
```

Hinweis: Zur Eingabe von ▸ drücken Sie [2nd] [▸].

% (Percent) CHAR/Punctuation-Menü

Term1 % ⇒ *Term*
Liste1 % ⇒ *Liste*
Matrix1 % ⇒ *Matrix*

Gibt $\frac{Parameter}{100}$ zurück.

Bei einer Liste oder einer Matrix wird eine Liste/Matrix zurückgegeben, in der jedes Element durch 100 dividiert ist.

```
13% [◆] ENTER                                .13
{1, 10, 100}% [◆] ENTER {.01 .1  1.}
```

= (gleich) [=]-Taste

Term1 = *Term2* ⇒ *Boolescher Term*
Liste1 = *Liste2* ⇒ *Boolesche Liste*
Matrix1 = *Matrix2* ⇒ *Boolesche Matrix*

Gibt wahr zurück, wenn *Term1* bei Auswertung gleich *Term2* ist.

Gibt falsch zurück, wenn *Term1* bei Auswertung ungleich *Term2* ist.

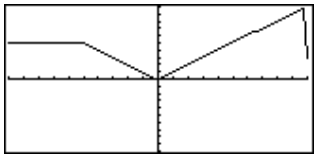
In allen anderen Fällen wird eine vereinfachte Form der Gleichung zurückgegeben.

Bei Listen und Matrizen werden die Ergebnisse des Vergleichs der einzelnen Elemente zurückgegeben.

Beispielfunktion mit den mathematischen Vergleichssymbolen =, ≠, <, ≤, > und ≥

```
:g(x)
:Func
:If x≤⁻5 Then
:  Return 5
:  ElseIf x>⁻5 and x<0 Then
:  Return ⁻x
:  ElseIf x≥0 and x≠10 Then
:  Return x
:  ElseIf x=10 Then
:  Return 3
:EndIf
:EndFunc
```

Graph g(x) [ENTER]



≠ [◆] [=]-Taste

Term1 /= *Term2* ⇒ *Boolescher Term*
Liste1 /= *Liste2* ⇒ *Boolesche Liste*
Matrix1 /= *Matrix2* ⇒ *Boolesche Matrix*

Gibt "true" zurück, wenn *Term1* bei Auswertung ungleich *Term2* ist.

Gibt "false" zurück, wenn *Term1* bei Auswertung gleich *Term2* ist.

In allen anderen Fällen wird eine vereinfachte Form der Gleichung zurückgegeben.

Bei Listen und Matrizen werden die Ergebnisse des Vergleichs der einzelnen Elemente zurückgegeben.

Siehe Beispiel bei "=" (gleich).

< [2nd] [<] -Taste

Term1 < *Term2* ⇒ *Boolescher Term*
Liste1 < *Liste2* ⇒ *Boolesche Liste*
Matrix1 < *Matrix2* ⇒ *Boolesche Matrix*

Gibt "true" zurück, wenn *Term1* bei Auswertung kleiner als *Term2* ist.

Gibt "false" zurück, wenn *Term1* bei Auswertung größer oder gleich *Term2* ist.

In allen anderen Fällen wird eine vereinfachte Form der Gleichung zurückgegeben.

Bei Listen und Matrizen werden die Ergebnisse des Vergleichs der einzelnen Elemente zurückgegeben.

Siehe Beispiel bei "=" (gleich).

≤ ◆ 0 -Tasten

Term1 <= *Term2* ⇒ *Boolescher Term*
Liste1 <= *Liste2* ⇒ *Boolesche Liste*
Matrix1 <= *Matrix2* ⇒ *Boolesche Matrix*

Gibt "true" zurück, wenn *Term1* bei Auswertung kleiner oder gleich *Term2* ist.

Gibt "false" zurück, wenn *Term1* bei Auswertung größer *Term2* ist.

In allen anderen Fällen wird eine vereinfachte Form der Gleichung zurückgegeben.

Bei Listen und Matrizen werden die Ergebnisse des Vergleichs der einzelnen Elemente zurückgegeben.

Siehe Beispiel bei "=" (gleich).

> 2nd [>] -Tasten

Term1 > *Term2* ⇒ *Boolescher Term*
Liste1 > *Liste2* ⇒ *Boolesche Liste*
Matrix1 > *Matrix2* ⇒ *Boolesche Matrix*

Gibt "true" zurück, wenn *Term1* bei Auswertung größer *Term2* ist.

Gibt "false" zurück, wenn *Term1* bei Auswertung kleiner oder gleich *Term2* ist.

In allen anderen Fällen wird eine vereinfachte Form der Gleichung zurückgegeben.

Bei Listen und Matrizen werden die Ergebnisse des Vergleichs der einzelnen Elemente zurückgegeben.

Siehe Beispiel bei "=" (gleich).

≥ ◆ . -Tasten

Term1 >= *Term2* ⇒ *Boolescher Term*
Liste1 >= *Liste2* ⇒ *Boolesche Liste*
Matrix1 >= *Matrix2* ⇒ *Boolesche Matrix*

Gibt "true" zurück, wenn *Term1* bei Auswertung größer oder gleich *Term2* ist.

Gibt "false" zurück, wenn *Term1* bei Auswertung kleiner als *Term2* ist.

In allen anderen Fällen wird eine vereinfachte Form der Gleichung zurückgegeben.

Bei Listen und Matrizen werden die Ergebnisse des Vergleichs der einzelnen Elemente zurückgegeben.

Siehe Beispiel bei "=" (gleich).

! (Fakultät) ◆ ÷ Taste 2nd Taste W

Term1! ⇒ *Term*
Liste1! ⇒ *Liste*
Matrix1! ⇒ *Matrix*

Gibt die Fakultät des Parameters zurück.

Bei Listen und Matrizen wird eine Liste/ Matrix mit der Fakultät der einzelnen Elemente zurückgegeben.

Der TI-89 errechnet einen numerischen Wert nur für nicht-negative ganzzahlige Werte.

5! ENTER → 120

{5,4,3}! ENTER → {120 24 6}

[1,2;3,4]! ENTER → $\begin{bmatrix} 1 & 2 \\ 6 & 24 \end{bmatrix}$

& (Verketten) ◆ × Taste 2nd Taste H

String1 & *String2* ⇒ *String*

Gibt eine Zeichenkette zurück, die durch Anfügen von *String2* an *String1* gebildet wurde.

"Hello " & "Nick" ENTER → "Hello Nick"

∫() (Integral) 2nd [∫] -Taste

∫(*Term1*, *Var*[, *unten*] [,*oben*]) ⇒ *Term*
∫(*Liste1*, *Var*[,*order*]) ⇒ *Liste*
∫(*Matrix1*, *Var*[,*order*]) ⇒ *Matrix*

Gibt das Integral von *Term1* bezüglich der Variablen *Var* zwischen *unten* und *oben* zurück.

∫(x^2,x,a,b) ENTER

$$\frac{b}{3} - \frac{a^3}{3}$$

Gibt ein unbestimmtes Integral zurück, wenn *unten* und *oben* nicht angegeben werden. Eine symbolische Integrationskonstante wie etwa C wird weggelassen.

∫(x^2,x) ENTER

$$\frac{x^3}{3}$$

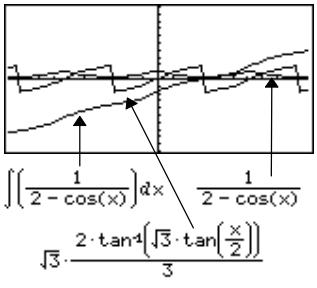
∫(a∗x^2,x,c) ENTER

$$\frac{a \cdot x^3}{3} + c$$

Wenn nur *oben* weggelassen wird, wird *unten* jedoch als Integrationskonstante hinzugefügt.

Gleichwertig gültige unbestimmte Integrale können durch eine numerische Konstante voneinander abweichen. Eine solche Konstante kann verborgen sein—insbesondere, wenn ein unbestimmtes Integral logarithmische oder inverse trigonometrische Funktionen enthält. Außerdem werden manchmal stückweise konstante Terme hinzugefügt, um einem unbestimmten Integral über ein größeres Intervall Gültigkeit zu verleihen als bei der üblichen Formel.

∫(1/(2−cos(x)),x)→tmp(x) ENTER
ClrGraph:Graph tmp(x):Graph
1/(2−cos(x)):Graph √(3)
(2tan⁻¹(√(3)(tan(x/2)))/3)
ENTER

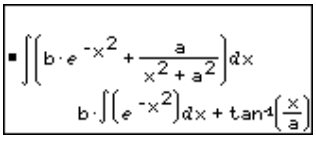


Falls das unbestimmte Integral nicht vollständig als endliche Kombination der verfügbaren Funktionen und Operatoren dargestellt werden kann, liefert ∫() eine Teillösung.

Sind sowohl *unten* als auch *oben* angegeben, wird versucht, Unstetigkeiten oder unstetige Ableitungen im Intervall *unten* < *Var* < *oben* zu finden, um das Intervall an diesen Stellen unterteilen zu können.

Ist der Modus Exact/Approx auf AUTO eingestellt, wird eine numerische Integration vorgenommen, wo dies möglich ist, wenn kein unbestimmtes Integral oder kein Grenzwert ermittelt werden kann.

∫(b∗e^(−x^2)+a/(x^2+a^2),x) ENTER

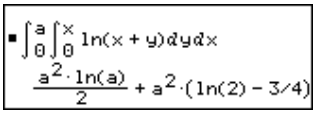


Bei der Einstellung APPROX wird die numerische Integration, wo möglich, zuerst versucht. Unbestimmte Integrale werden nur dann gesucht, wenn die numerische Integration unzulässig ist oder fehlschlägt.

∫(e^(−x^2),x,−1,1)◆ ENTER 1.493...

Sie können ∫() verschachteln, um Mehrfach-Integrale zu bearbeiten. Die Integrationsgrenzen können von außerhalb liegenden Integrationsvariablen abhängen.

∫(∫(ln(x+y),y,0,x),x,0,a) ENTER



Hinweis: Siehe auch **nInt()**.

√() (Qdr.-wurzel) 2nd [√] -Tasten

√(*Term1*) ⇒ *Term*
√(*Liste1*) ⇒ *Liste*

Gibt die Quadratwurzel des Parameters zurück.

Bei einer Liste wird die Quadratwurzel für jedes Element von *Liste1* zurückgegeben.

√(4) ENTER 2

√({9,a,4}) ENTER {3 √a 2}

Π() (Produkt) **MATH/Calculus-Menü**

Π(*Term1***,** *Var***,** *unten***,** *oben***)** ⇒ *Term*

Wertet *Term1* für jeden Wert von *Var* zwischen *unten* und *oben* aus und gibt das Produkt der Ergebnisse zurück.

Π(1/n,n,1,5) ENTER → $\frac{1}{120}$

Π(k^2,k,1,n) ENTER → $(n!)^2$

Π({1/n,n,2},n,1,5) ENTER → $\{\frac{1}{120}\ 120\ 32\}$

Π(*Term1***,** *Var***,** *unten***,** *unten*− 1**)** ⇒ 1

Π(k,k,4,3) ENTER → 1

Π(*Term1***,** *Var***,** *unten***,** *oben***)** ⇒ 1/Π(*Term1*, *Var*, *oben*+1, *unten*− 1) wenn *oben*< *unten*− 1

Π(1/k,k,4,1) ENTER → 6

Π(1/k,k,4,1)*Π(1/k,k,2,4) ENTER → 1/4

Σ() (Summe) **MATH/Calculus Menü**

Σ(*Term1***,** *Var***,** *unten***,** *oben***)** ⇒ *Term*

Wertet *Term1* für jeden Wert von *Var* zwischen *unten* und *oben* aus und gibt die Summe der Ergebnisse zurück.

Σ(1/n,n,1,5) ENTER → $\frac{137}{60}$

Σ(k^2,k,1,n) ENTER → $\frac{n\cdot(n+1)\cdot(2\cdot n+1)}{6}$

Σ(1/n^2,n,1,∞) ENTER → $\frac{\pi^2}{6}$

Σ(*Term1***,** *Var***,** *unten***,** *unten*− 1**)** ⇒ 0

Σ(k,k,4,3) ENTER → 0

Σ(*Term1***,** *Var***,** *unten***,** *oben***)** ⇒ −**Σ(***Term1*, *Var*, *oben*+1, *unten*− 1) wenn *oben*< *unten*− 1

Σ(k,k,4,1) ENTER → −5

Σ(k,k,4,1)+Σ(k,k,2,4) ENTER → 4

(Umleitung) **CATALOG**

*VarNameString*

Greift auf die Variable namens *VarNameString* zu. So können Sie innerhalb eines Programms Variablen unter Verwendung von Zeichenketten (Strings) erzeugen oder ändern.
(Indirekte Adressierung)

Programmsegment:

```
  ⋮
:Request "Enter Your Name",str1
:NewFold #str1
  ⋮
  ⋮
:For i,1,5,1
:  ClrGraph
:  Graph i*x
:  StoPic #("pic" & string(i))
:EndFor
  ⋮
```

ʳ (rad) **MATH/Angle-Menü**

*Term1*ʳ ⇒ *Term*
*Liste1*ʳ ⇒ *Liste*
*Matrix1*ʳ ⇒ *Matrix*

Im Degree-Modus für Winkel wird *Term1* mit 180/π multipliziert. Im Radian-Modus wird *Term1* unverändert zurückgegeben.

Diese Funktion ermöglicht es Ihnen, im Ordnungmodus Bogenmaßangaben zu verwenden. (Im Winkelmodus Degree erwarten die Funktionen **sin()**, **cos()**, **tan()** und die Konvertierung Polar-nach-Kartesisch, dass der Winkelparameter in Ordnung vorliegt.)

Tipp: Verwenden Sie ʳ in einer Funktions- oder Programmdefinition, wenn Sie bei Ausführung der Funktion/des Programms das Bogenmaß frei von der Winkelmoduseinstellung erzwingen möchten.

Im Winkelmodus Ordnung oder Bogenmaß:

`cos((π/4)ʳ)` ENTER $\frac{\sqrt{2}}{2}$

`cos({0ʳ,(π/12)ʳ,⁻πʳ})` ENTER

$\{1 \quad \frac{(\sqrt{3}+1)\cdot\sqrt{2}}{4} \quad -1\}$

° (Ordnung) 2nd [°]-Taste

Term° ⇒ *Wert*
Liste1° ⇒ *Liste*
Matrix1° ⇒ *Matrix*

Im Radian-Modus für Winkel wird *Term* mit π/180 multipliziert. Im Degree-Modus wird *Term* unverändert zurückgegeben.

Diese Funktion ermöglicht es Ihnen, im Bogenmaßmodus Angaben in Ordnung vorzunehmen. (Im Winkelmodus Radian erwarten die Funktionen **sin()**, **cos()**, **tan()** und die Konvertierung Polar-nach-Kartesisch, dass der Winkelparameter im Bogenmaß vorliegt.)

Im Radian-Modus:

`cos(45°)` ENTER $\frac{\sqrt{2}}{2}$

`cos({0,π/4,90°,30.12°})` ◆ ENTER

{1 .707… 0 .864…}

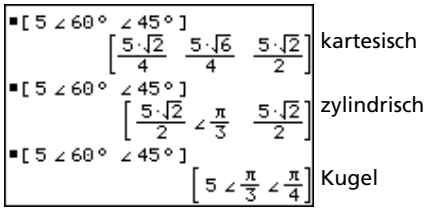
∠ (Winkel) 2nd [∠]-Tasten

[*radius*,∠θ_*winkel*] ⇒ *Vektor* (Eingabe polar)
[*radius*,∠θ_*winkel*,*Z_koordinate*] ⇒ *Vektor* (Eingabe cylindrical)
[*radius*,∠θ_*winkel*,∠ϕ_*winkel*] ⇒ *Vektor* (Eingabe spherical)

Gibt Koordinaten als Vektor zurück, wobei die aktuelle Einstellung für Vector Format gilt: kartesische, zylindrische oder Kugelkoordinaten.

`[5,∠60°,∠45°]` ENTER

Im Radian-Modus mit Vektorformat eingestellt auf:



(*Absolutwert* ∠ *Winkel*) ⇒ *komplexer_Wert* (Eingabe polar)

Dient zur Eingabe eines komplexen Werts in polarer (r∠θ) Form. Der *Winkel* wird gemäß der aktuellen Winkelmodus-Einstellung interpretiert.

Im Winkelmodus Radian und im Komplex-Formatmodus "Rectangular":

`5+3i-(10∠π/4)` ENTER

$5-5\cdot\sqrt{2}+(3-5\cdot\sqrt{2})\cdot i$

◆ ENTER $-2.071\ldots-4.071\ldots\cdot i$

°, ', " 2nd [°]-Taste (°), 2nd [′]-Taste ('), 2nd ["]-Taste (")

dd°*mm*'*ss.ss*" ⇒ *Term*

dd Eine positive oder negative Zahl
mm Eine nicht-negative Zahl
ss.ss Eine nicht-negative Zahl

Gibt zurück: *dd*+(*mm*/60)+(*ss.ss*/3600).

Mit einer solchen Eingabe auf der Basis 60 können Sie:

- Einen Winkel unabhängig vom aktuellen Winkelmodus in Ordnung/Minuten/Sekunden eingeben.
- Uhrzeitangaben in Stunden/Minuten/ Sekunden vornehmen.

Im Degree-Modus für Winkel:

25°13'17.5" ENTER 25.221...

25°30' ENTER 51/2

'(Strich) 2nd [′]-Taste

Variable'
Variable''

Dient zur Eingabe eines Strichs bei einer Differentialgleichung. Ein einfacher Strich kennzeichnet eine Differentialgleichung erster Ordnung, zwei Striche stehen für zweite Ordnung usw.

```
deSolve(y''=y^(⁻1/2) and
y(0)=0 and y'(0)=0,t,y) ENTER
```

$$\frac{2 \cdot y^{3/4}}{3} = t$$

(Unterstr.) [◆][] Taste [2nd][_] Taste

Term_Einheit

Kennzeichnet die Einheiten für einen *Term*. Alle Einheitennamen müssen mit einem Unterstrich beginnen.

Sie können entweder vordefinierte Einheiten verwenden oder Ihre eigenen erstellen. Eine Liste vordefinierter Einheiten finden Sie in diesem Handbuch im modul über Konstanten und Maßeinheiten. Sie können entweder:
[2nd] [UNITS]
[◆] [UNITS]
drücken und Maßeinheiten aus einem Menü wählen, oder Sie können die Maßeinheitsbezeichnungen direkt eingeben.

3_m▶_ft ENTER 9.842…·_ft

Hinweis: Zur Eingabe von ▶ drücken Sie [2nd] [▶].

---

*Variable_*

Besitzt *Variable* keinen Wert, so wird sie behandelt, als würde sie eine komplexe Zahl darstellen. Die Variable wird ohne das Zeichen _ standardmäßig als reell behandelt.

Besitzt *Variable* einen Wert, so wird das Zeichen _ ignoriert, und *Variable* behält ihren ursprünglichen Datentyp bei.

Hinweis: Eine komplexe Zahl kann ohne Unterstrich _ in Variablen einer gespeichert werden. Bei Berechnungen wie **cSolve()** und **cZeros()** empfiehlt sich allerdings die Verwendung von _ , um beste Ergebnisse zu erzielen.

z sei undefiniert:

```
real(z) ENTER                z
real(z_) ENTER          real(z_)

imag(z) ENTER                0
imag(z_) ENTER          imag(z_)
```

▸(konvertieren) 2nd [▸]-Taste

Term_Einheit1▸*_Einheit2* ⇒ *Term_Einheit2*

Konvertiert einen Term von einer Einheit in eine andere. Diese Einheiten müssen sich in derselben Kategorie befinden.

Der Unterstrich _ kennzeichnet die Einheiten. Eine Liste vordefinierter Einheiten finden Sie in diesem Handbuch im modul über Konstanten und Maßeinheiten. Sie können entweder:

2nd [UNITS]

◆ [UNITS]

Drücken und Maßeinheiten aus einem Menü wählen, oder Sie können die Maßeinheitsbezeichnungen direkt eingeben.

Der Unterstrich (_) bei direkter Eingabe der Maßeinheiten wird wie folgt erzeugt:

◆ [_]

2nd [_]

Hinweis: Der Konvertierungsoperator ▸ ist nicht für Temperatureinheiten anwendbar. Verwenden Sie statt dessen **tmpCnv()** und **ΔtmpCnv()**.

```
3_m▸_ft ENTER                    9.842…·_ft
```

10^() CATALOG

10^ (*Term1***)** ⇒ *Term*
10^ (*Liste1***)** ⇒ *Liste*

Gibt 10 hoch Parameter zurück.

Bei einer Liste wird 10 hoch jedem Element von *Liste1* zurückgegeben.

```
10^1.5 ENTER                     31.622…
10^{0,-2,2,a} ENTER
                          {1  1/100  100  10^a}
```

10^(*quadrat_Matrix1***)** ⇒ *quadrat_Matrix*

Ergibt 10 hoch *quadrat_Matrix1*. Dies ist *nicht* gleichbedeutend mit der Berechnung von 10 hoch jedem Element. Näheres zum Berechnungsverfahren finden Sie im Abschnitt **cos**.

Quadrat_Matrix1 muß diagonalisierbar sein. Das Ergebnis enthält stets Fließpunktzahlen.

```
10^([1,5,3;4,2,1;6,-2,1]) ENTER

[1.143…E7   8.171…E6   6.675…E6]
[9.956…E6   7.115…E6   5.813…E6]
[7.652…E6   5.469…E6   4.468…E6]
```

x^{-1} **CATALOG** (^-1)

Term1 x^{-1} ⇒ *Term*
Liste1 x^{-1} ⇒ *Liste*

Gibt den Kehrwert des Parameters zurück.

Bei einer Liste wird für jedes Element von *Liste1* der Kehrwert zurückgegeben.

```
3.1^-1 ENTER                                  .322581
{a,4,-.1,x-2}^-1 ENTER
                                   {1/a  1/4  -10  1/(x-2)}
```

Quadr_Matrix1 x^{-1} ⇒ *Quadr_Matrix*

Gibt die Inverse von *Quadr_Matrix1* zurück.

Quadr_Matrix1 muß eine nicht-singuläre quadratische Matrix sein.

```
[1,2;3,4]^-1 ENTER
[1,2;a,4]^-1 ENTER
```

$$\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix}^{-1} \qquad \begin{bmatrix}-2 & 1\\ 3/2 & -1/2\end{bmatrix}$$

$$\begin{bmatrix}1 & 2\\ a & 4\end{bmatrix}^{-1} \qquad \begin{bmatrix}\frac{-2}{a-2} & \frac{1}{a-2}\\ \frac{a}{2\cdot(a-2)} & \frac{-1}{2\cdot(a-2)}\end{bmatrix}$$

| ("with") ▦ [|] **Taste** ⌨ [2nd] [|] **Taste**

Term | *Boolescher Term1* [*and Boolescher Term2*]...[*and Boolescher TermN*]

Das Symbol "with" (|) dient als binärer Operator. Der Operand links des | ist ein Term. Der Operand rechts des | gibt eine oder mehrere Relationen an, die auf die Vereinfachung des Terms einwirken sollen. Bei Angabe mehrerer Relationen nach dem | sind diese jeweils mit einem logischen "and" miteinander zu verketten.

```
x+1| x=3 ENTER                                  4
x+y| x=sin(y) ENTER                        sin(y)+y
x+y| sin(y)=x ENTER                            x+y
```

Der Operator "with" erfüllt drei Grundaufgaben: Ersetzung (Substitution), Intervallbeschränkung sowie Ausschließung.

Ersetzungen werden in Form einer Gleichung angegeben, etwa x=3 oder y=sin(x). Am wirksamsten ist eine Ersetzung, wenn die linke Seite eine einfache Variable ist.
Term | *Variable* = *Wert* bewirkt, daß jedesmal, wenn *Variable* in *Term* vorkommt, *Variable* durch *Wert* ersetzt wird.

```
x^3-2x+7→f(x) ENTER                            Done
f(x)| x=√(3) ENTER                             √3+7
(sin(x))^2+2sin(x)-6| sin(x)=d ENTER
                                              d²+2d-6
```

Intervallbeschränkungen werden in Form einer oder mehrerer mit logischem "and" verknüpfter Ungleichungen angegeben. Intervallbeschränkungen ermöglichen auch Vereinfachungen, die andernfalls ungültig oder nicht berechenbar wären.

```
solve(x^2-1=0,x)|x>0 and x<2 ENTER
                                                x=1
√(x)*√(1/x)|x>0 ENTER                             1
√(x)*√(1/x) ENTER                          √(1/x)·√x
```

Ausschließungen verwenden den relationalen Operator "ungleich" (/= oder ≠), um einen bestimmten Wert bei der Operation auszuschließen. Sie dienen hauptsächlich zum Ausschließen einer exakten Lösung bei Verwendung von **cSolve()**, **cZeros()**, **fMax()**, **fMin()**, **solve()**, **zeros()** usw.

```
solve(x^2-1=0,x)| x≠1 ENTER x=-1
```

→ (Speichern) STO▸ -Taste

Term → *Var*
Liste → *Var*
Matrix → *Var*
Term → *Funktionsname(Parameter1,...)*
Liste → *Funktionsname(Parameter1,...)*
Matrix → *Funktionsname(Parameter1,...)*

Wenn *Var* noch nicht existiert, wird *Var* erzeugt und auf *Term*, *Liste* oder *Matrix* initialisiert.

Wenn *Var* existiert und nicht gesperrt oder geschützt ist, wird der Variableninhalt durch *Term*, *Liste* bzw. *Matrix* ersetzt.

Tipp: Wenn Sie symbolische Rechnungen mit undefinierten Variablen vornehmen möchten, sollten Sie vermeiden, Werte in Variablen mit häufig benutzten Einzeichennamen abzuspeichern (etwa den Variablen a, b, c, x, y, z usw.).

```
π/4→myvar ENTER                                  π/4
2cos(x)→Y1(x) ENTER                              Done
{1,2,3,4}→Lst5 ENTER                             {1 2 3 4}
[1,2,3;4,5,6]→MatG ENTER                         [1 2 3]
                                                 [4 5 6]
"Hello"→str1 ENTER                               "Hello"
```

◉ (Kommentar) Menü Program Editor/Control oder

▦ ◆) **Taste**

⌨ 2nd **Taste** X

◉ **[***Text***]**

◉ bewirkt, daß *Text* als Kommentar behandelt wird. Kommentare können zur Erläuterung von Programmanweisungen benutzt werden.

◉ kann an den Zeilenanfang oder an eine beliebige Stelle der Zeile gesetzt werden. Alles, was rechts vom ◉ bis zum Zeilenende steht, gilt als Kommentar.

Programmsegment:

```
   :
:◉ Get 10 points from the Graph
   screen
:For i,1,10 ◉ This loops 10
times
   :
```

0b, 0h ▦ 0 alpha [B] Tasten ⌨ 0 B Tasten

▦ 0 alpha **[H] Tasten** ⌨ 0 **H Tasten**

0b *binäre_Zahl*
0h *hexadezimale_Zahl*

Kennzeichnet eine Dual- bzw. Hexadezimalzahl. Zur Eingabe einer Dual- oder Hexadezimalzahl muß unabhängig vom jeweiligen Base-Modus das Präfix 0b bzw. 0h verwendet werden. Eine Zahl ohne Präfix wird als dezimal behandelt (Grundzahl 10).

Die Ergebnisse werden im jeweiligen Base-Modus angezeigt.

Im Modus Dec base:

```
0b10+0hF+10 ENTER                                27
```

Im Modus Bin base:

```
0b10+0hF+10 ENTER                          0b11011
```

Im Modus Hex base:

```
0b10+0hF+10 ENTER                            0h1B
```

Anhang B: Technische Referenz

Dieser Abschnitt enthält eine umfassende Zusammenstellung der Fehlermeldungen und Zeichencodes des TI-89 Titanium / Voyage™ 200. Er liefert außerdem Informationen zur Berechnungsweise bestimmter TI-89 Titanium / Voyage 200 - Operationen.

TI-89 Titanium / Voyage™ 200 Fehlermeldungen 998
TI-89 Titanium / Voyage™ 200 Modi 1006
Zeichen-Codes auf TI-89 Titanium / Voyage™ 200 1012
Tasten-Codes auf TI-89 1013
Tasten-Codes auf Voyage™ 200 1016
Komplexe Zahlen eingeben 1020
Hinweise zur Verarbeitungs- und Darstellungsgenauigkeit 1023
Systemvariablen und reservierte Namen 1024
Die Auswertungrangfolge des EOS™ (Equation Operating System) 1025
Regressionsformeln 1027
Die Algorithmen “Contour Levels” und “Implicit Plot” 1029
Die Runge-Kutta-Methode 1030
Hinweise zu den Batterien 1031
Bei Betriebsproblemen 1034
Hinweise zu TI Produktservice und Garantieleistungen 1036

TI-89 Titanium / Voyage™ 200 Fehlermeldungen

In der nachfolgenden Tabelle sind die Fehlermeldungen zusammengestellt, welchen Sie bei Eingabefehlern oder internen Fehlern begegnen. Die in der linken Spalte aufgeführten Fehlernummern werden nicht auf dem Bildschirm angezeigt. Wenn ein Fehler innerhalb eines Blocks Try...EndTry auftritt, wird die Fehlernummer in die Systemvariable *errornum* abgelegt. Viele der Fehlermeldungen sind so eindeutig, daß keine zusätzliche Erläuterung nötig ist. Für manche Meldungen sind jedoch nähere Erläuterungen angeführt. Die Liste enthält neben der angezeigten englischen Meldung die deutsche Übersetzung.

Fehler-nummer	Beschreibung
10	**A function did not return a value -** *(Ein Funktionswert ließ sich nicht berechnen)*
20	**A test did not resolve to TRUE or FALSE -** *(Test ergab weder TRUE noch FALSE)* Undefinierte Variablen können beim Vergleich nicht ausgewertet werden. So ergibt beispielsweise die Prüfung von If a<b einen Fehler, wenn a oder b bei Ausführung der Anweisung If undefiniert ist.
30	**Argument cannot be a folder name -** *(Ein Verzeichnisname kann nicht als Argument verwendet werden)*
40	**Argument error -** *(Fehler im Argument)*
50	**Argument mismatch -** *(Argumente passen nicht zusammen)* Angezeigt, wenn zwei oder mehr Parameter den gleichen Typ besitzen müssen. So sind z. B. sowohl **PtOn** *Term1,Term2* als auch **PtOn** *Liste1,Liste2* gültig, aber **PtOn** *Term,Liste* ist ein Fehler.
60	**Argument must be a Boolean expression or integer -** *(Argument muß ein Boolescher Ausdruck sein)*
70	**Argument must be a decimal number -** *(Dezimalzahl erforderlich)*
80	**Argument must be a label name -** *(Argument muß eine Marke sein)*
90	**Argument must be a list-** *(Argument muß eine Liste sein)*
100	**Argument must be a matrix -** *(Argument muß eine Matrix sein)*
110	**Argument must be a Pic -** *(Argument muß eine Graphik sein)*
120	**Argument must be a Pic or string -** *(Argument muß eine Graphik oder ein String sein)*
130	**Argument must be a string -** *(Argument muß ein String sein)*
140	**Argument must be a variable name -** *(Argument muß eine Variable sein)* Beispiel: DelVar 12 ist unzulässig, da eine Zahl kein Variablenname sein kann.
150	**Argument must be an empty folder name -** *(Argument muß der Name eines leeren Verzeichnisses sein)*

Fehler-nummer	Beschreibung
160	**Argument must be an expression -** *(Argument muß ein Term sein)* Beispiel: zeros(2x+3=0,x) ist unzulässig, da der erste Parameter eine Gleichung ist.
161	**ASAP or Exec string too long -** *(Strings ASAP oder Exec zu lang)*
163	**Attribute (8-digit number) of object (8-digit number) not found -** *(Attribut (8-stellige Zahl) des Objekts (Zahl) nicht zu finden)*
165	**Batteries too low for sending or receiving** *(Tauschen Sie vor Versand/Empfang die Batterien aus)*
170	**Bound -** *(Grenze)* Bei interaktiven Graphfunktionen wie 2:Zero muß der untere Rand kleiner sein als der obere, damit ein gültiges Suchintervall definiert ist.
180	**Break -** *(Abbruch)* Während einer Berechnung oder der Ausführung eines Programms wurde die Taste [ON] gedrückt.
185	**Checksum error -** *(Prüfsummenfehler)*
190	**Circular definition -** *(Zirkuläre Definition)* Diese Meldung weist Sie darauf hin, daß bei einer Vereinfachung eine Endlosersetzung von Variablenwerten auftreten würde, wodurch der Speicherplatz erschöpft würde. Beispiel: a+1→a, wobei a eine undefinierte Variable ist, führt zu diesem Fehler.
200	**Constraint expression invalid -** *(Zusammengesetzter Ausdruck ungültig)* Beispiel: solve(3x^2− 4=0, x) | x<0 or x>5 würde diese Fehlermeldung hervorrufen, da die Beschränkung fälschlicherweise “or” statt “and” enthält.
205	**Data is too big to save to a variable. Please use F6 Util to reduce the size.** *(Daten sind zu umfangreich zum Speichern in einer Variablen. Bitte verringern Sie die Größe mit F6 Util.)* Der Umfang der Daten im Editor überschreitet die maximale Größe, die in einer Variablen gespeichert werden kann. Mit den Operationen im Menü F6 Util kann der Umfang der Daten verringert werden.
210	**Data type -** *(Datentyp)* Ein Parameter besitzt einen falschen Datentyp.
220	**Dependent Limit -** *(Abhängiger Grenzwert)* Die Grenze eines Integrals ist von der Integrationsvariablen abhängig. Beispiel: ∫(x^2,x,1,x) ist nicht gestattet.
225	**Diff Eq setup -** *(Einstellung der Diff Glch.)*
230	**Dimension -** *(Dimension)* Ungültige Liste oder Matrix. Beispiel: Ist die Liste {1,2,3,4} in L1 gespeichert, stellt L1[5] einen Dimensionsfehler dar, da L1 nur vier Elemente enthält.

Fehler-nummer	Beschreibung

240 **Dimension mismatch -** *(Dimensionsfehler)*

Angezeigt, wenn zwei oder mehr Parameter die gleiche Dimension besitzen müssen. Beispiel: [1,2]+[1,2,3] verursacht diesen Fehler, da die Matrizen eine unterschiedlich große Anzahl Elemente enthalten.

250 **Divide by zero -** *(Division durch Null)*

260 **Domain error -** *(Bereichsfehler)*

Ein Parameter muß innerhalb eines bestimmten Bereichs liegen, tut es jedoch hier nicht. Beispiel: ans(100) ist ungültig, da der Parameterwert für **ans()** zwischen 1 und 99 liegen muß.

270 **Duplicate variable name -** *(Variablenname doppelt vergeben)*

280 **Else and ElseIf invalid outside of If..EndIf block -** *(Else und ElseIf können nur innerhalb eines If ... EndIf Blocks verwendet werden)*

290 **EndTry is missing the matching Else statement -** *(EndTry benötigt die Else-Anweisung)*

295 **Excessive iteration -** *(Zu viele Iterationen)*

300 **Expected 2 or 3-element list or matrix -** *(2 oder 3-elementige Liste oder Matrix erwartet)*

307 **Flash application extension (function or program) not found -** *(Erweiterung der Flash-Anwendung (Funktion bzw. Programm) nicht zu finden)*

308 **Flash application not found -** *(Flash-Anwendung nicht zu finden)*

310 **First argument of nSolve must be a univariate equation -** *(Erstes Argument von nSolve muß eine Gleichung mit einer Unbekannten sein)*

Der erste Parameter muß eine Gleichung sein, und die Gleichung darf als einzige Variable ohne Wert die gesuchte Variable enthalten. Beispiel: nSolve(3x^2− 4=0, x) ist eine gültige Gleichung; dagegen ist nSolve(3x^2− 4, x) unzulässig, weil keine Gleichung vorliegt, und nSolve(3x^2− y=0,x) enthält keine Ein-Variablen-Gleichung, da y in diesem Beispiel keinen Wert besitzt.

320 **First argument of solve or cSolve must be an equation or inequality -** *(Erstes Argument von solve oder cSolve muß eine Gleichung oder Ungleichung sein)*

Beispiel: solve(3x^2− 4, x) ist ungültig, weil der erste Parameter keine Gleichung ist.

330 **Folder -** *(Verzeichnis)*

Im Menü VAR-LINK wurde versucht, eine Variable in ein nicht existierendes Verzeichnis zu speichern.

335 **Graph functions y1(x)...y99(x) not available in Diff Equations mode -**
(Die Graphikfunktionen y1(x)...y99(x) sind im Modus Diff Equations nicht verfügbar)

345 **Inconsistent units -** *(Widersprüchliche Einheiten)*

350 **Index out of range -** *(Index außerhalb der Bereichsgrenzen)*

360 **Indirection string is not a valid variable name -** *(Umleitungs-String ist kein gültiger Variablenname)*

Fehler-nummer	Beschreibung
380	**Invalid ans() -** *(ans() Ungültig)*
390	**Invalid assignment -** *(UngültigeZuweisung)*
400	**Invalid assignment value -** *(Ungültiger Zuweisungswert)*
405	**Invalid axes -** *(Ungültige Achsen)*
410	**Invalid command-** *(Ungültiger Befehl)*
420	**Invalid folder name -** *(Ungültiger Verzeichnisname)*
430	**Invalid for the current mode settings -** *(Ungültigt im aktuellen Arbeits-Mode)*
440	**Invalid implied multiply -** *(Ungültige implizite Multiplikation)* Beispiel: x(x+1) ist ungültig; dagegen ist x∗ (x+1) die korrekte Syntax. Damit werden Verwechselungen zwischen impliziter Multiplikation und Funktionsaufrufen verhindert.
450	**Invalid in a function or current expression -** *(In einer Funktion oder dem aktuellen Ausdruck)* Nur bestimmte Befehle sind in einer benutzerdefinierten Funktion zulässig. Eingaben, die im Window-Editor, Tabellen-Editor, Daten/Matrix Editor, Solver und der Geometry-Anwendung vorgenommen werden, sowie Systemeingabe-aufforderungen wie Lower Bound dürfen keine Funktionen oder Doppelpunkte (:) enthalten. Siehe auch “Benutzerdefinierte Funktionen erstellen und auflösen” in Kapitel 5.
460	**Invalid in Custom..EndCustm block -** *(Ungültig in einem Custom ... EndCustm Block)*
470	**Invalid in Dialog..EndDlog block -** *(Ungültig in einem Dialog ... EndDlog Block)*
480	**Invalid in Toolbar..EndTBar block -** *(Ungültig in einem ToolBar ... EndTBar Block)*
490	**Invalid in Try..EndTry block -** *(Ungültig zwischen Try ... EndTry)*
500	**Invalid label -** *(Ungültige Marke)* Sie müssen die gleichen Benennungsregeln wie für Variablen einhalten.
510	**Invalid list or matrix -** *(Ungültige Liste oder Matrix)* Beispiel: Eine Liste innerhalb einer Liste {2,{3,4}} ist ungültig.
520	**Invalid outside Custom..EndCustm or ToolBar..EndTbar blocks -** *(Ungültig außerhalb eines Custom..EndCustm oder ToolBar..EndTBar Blocks)* Beispiel: Ein Befehl **Item**, der außerhalb einer Struktur **Custom** oder **ToolBar** eingegeben wird.
530	**Invalid outside Dialog..EndDlog, Custom..EndCustm, or ToolBar..EndTBar blocks -** *(Ungültig außerhalb Dialog..EndDlog, Custom..EndCustm oder ToolBar..EndTBar Blocks)* Beispiel: Ein Befehl Title, der außerhalb einer Struktur **Dialog**, **Custom** oder **ToolBar** eingegeben wird.
540	**Invalid outside Dialog..EndDlog block -** *(Ungültig außerhalb eines Dialog..EndDlog Blocks)* Beispiel: Ein Befehl **DropDown**, der außerhalb einer Struktur **Dialog** eingegeben wird.

Fehler-nummer	Beschreibung
550	**Invalid outside function or program -** *(Ungültig außerhalb function oder program)* Manche Befehle sind außerhalb eines Programms oder einer Funktion nicht zulässig. Beispiel: Local darf nur in einem Programm/einer Funktion benutzt werden.
560	**Invalid outside Loop..EndLoop, For..EndFor, or While..EndWhile blocks -** *(Ungültig außerhalb Loop..EndLoop, For..EndFor, While..EndWhile Blocks)* Beispiel: Der Befehl **Exit** darf nur innerhalb dieser Schleifen benutzt werden.
570	**Invalid pathname -** *(Ungültiger Pfadname)* Beispiel: \\var ist ungültig.
575	**Invalid polar complex -** *(Ungültige komplexe Zahl in Polarform)*
580	**Invalid program reference -** *(Ungültiger Programmaufruf)* Programme können nicht in Funktionen oder Termen wie 1+p(x), wobei p ein Programm ist, aufgerufen werden.
585	**Invalid relocation data in ASM program** *(Ungültige Verschiebungsdaten in ASM-Program)* Notwendige Verschiebungsdaten in einem ASM (Assembler) Programm fehlen oder sind beschädigt.
590	**Invalid syntax block -** *(Ungültige Syntax)* Ein Block **Dialog..EndDlog** ist leer oder enthält mehr als einen Titel. Ein Block **Custom..EndCustm** darf keine Variablen des Typs PIC enthalten, und die “Item”-Angaben können nur nach einem “Title” stehen. Ein Block **Toolbar..EndTBar** muß einen zweiten Parameter besitzen, wenn keine “Item”-Angaben folgen; oder “Item”-Angaben müssen einen zweiten Parameter besitzen und auf einen “Title” folgen.
600	**Invalid table -** *(Ungültige Tabelle)*
605	**Invalid use of units -** *(Ungültige Verwendung von Einheiten)*
610	**Invalid variable name in a Local statement -** *(Ungültige Variable in einem “local statement”)*
620	**Invalid variable or function name -** *(Ungültiger Variablen- oder Funktionsname)*
630	**Invalid variable reference -** *(Ungültiger Zugriff auf eine Variable)*
640	**Invalid vector syntax -** *(Ungültige Vektor-Schreibweise)*
650	**Link transmission -** *(Übertragung von Daten)* Eine Übertragung zwischen zwei Geräten ist fehlgeschlagen. Überprüfen Sie, ob das Verbindungskabel korrekt sitzt.
665	**Matrix not diagonalizable -** *(Matrix nicht diagonalisierbar)*

Fehler-nummer	Beschreibung
670 **673**	**Memory -** *(Speicher)* Für eine Berechnung stand nicht genügend Speicherplatz zur Verfügung. Erhalten Sie bei der Ausführung eines großen Programms diese Fehlermeldung, sollten Sie das Programm möglicherweise in kleinere Programme oder Funktionen aufspalten (wobei ein Programm - bzw. eine Funktion - ein anderes aufruft).
680	**Missing (-** *((fehlt)*
690	**Missing) -** *() fehlt)*
700	**Missing " -** *(" fehlt)*
710	**Missing] -** *(] fehlt)*
720	**Missing } -** *(} fehlt)*
730	**Missing start or end of block syntax -** *(Anfangs- oder Endbefehl eines Blocks fehlt)*
740	**Missing Then in the If..EndIf block -** *(Fehlendes Then in einem If ... EndIf Block)*
750	**Name is not a function or program -** *(Name bezeichnet weder eine Funktion noch ein Programm)*
765	**No functions selected -** *(Es wurden keine Funktionen gewählt)*
780	**No solution found -** *(Keine Lösung gefunden)* Dieser Fehler kann bei Benutzung der interaktiven Funktionen des Menüs F5:Math in der Anwendung Graph auftreten. Beispiel: Sie versuchen einen Wendepunkt der Parabel $y1(x)=x^2$ (nicht vorhanden) zu ermitteln.
790	**Non-algebraic variable in expression -** *(Nichtalgebraische Variable im Ausdruck)* Ist a der Name einer Variablen des Typs PIC, GDB, MAC, FIG etc., dann ist a+1 ungültig. Verwenden Sie einen anderen Variablennamen, oder löschen Sie die Variable.
800	**Non-real result -** *(Nichtreelles Ergebnis)* Beispiel: Ist der Modus Complex Format auf REAL eingestellt, dann ist $\ln(^{-}2)$ ungültig.
810	**Not enough memory to save current variable. Please delete unneeded variables on the Var-Link screen and re-open editor as current OR re-open editor and use F1 8 to clear editor. -** *(Nicht genügend Speicher, um die aktuelle Variable zu speichern. Bitte löschen Sie überflüssige Variable im Var-link Bildschirm, und öffnen Sie den Editor erneut als aktuellen Editor oder als neuen Editor, und benutzen Sie F1 8, um den Editor zu löschen)* Diese Fehlermeldung erhalten Sie bei akutem Speicherplatzmangel im Daten/Matrix-Editor.
830	**Overflow -** *(Überlauf)*
840	**Plot setup -** *(Graphik -Setup)*

Fehler-nummer	Beschreibung
850	**Program not found -** *(Programm nicht gefunden)* Ein Programm, das innerhalb eines anderen aufgerufen wird, konnte bei Ausführung nicht in dem angegebenen Pfad gefunden werden.
860	**Recursion is limited to 255 calls deep -** *(Rekursionstiefe ist auf 255 begrenzt)*
870	**Reserved name or system variable -** *(Reservierter Name oder Systemvariable)*
875	**Reserved name or system variable -** *(ROM-residente Routine nicht verfügbar)*
880	**Sequence setup -** *(Folge-Setup)*
885	**Signature error -** *(Signaturfehler)*
890	**Singular matrix -** *(Singuläre Matrix)*
895	**Slope fields need one selected function and are used for 1st-order equations only -** *(Für Steigungsfelder ist eine Funktion zu wählen. Sie dürfen nur für Gleichungen erster Ordnung verwendet werden)*
900	**Stat -** *(Statistik)*
910	**Syntax -** *(Syntax)* Die Struktur der Eingabe ist falsch. Beispiel: x+− y (x plus minus y) ist ungültig; x+− y (x plus negatives y) dagegen korrekt.
930	**Too few arguments -** *(Zu wenige Argumente)* In einem Term oder einer Gleichung fehlt mindestens ein Parameter. Beispiel: *d*(f(x)) ist ungültig; korrekt ist *d*(f(x),x).
940	**Too many arguments -** *(Zu viele Argumente)* Ein Term oder eine Gleichung enthält zu viele Parameter und kann deshalb nicht ausgewertet werden.
950	**Too many subscripts -** *(Zu viele Indizes)*
955	**Too many undefined variables -** *(Zu viele undefinierte Variablen)*
960	**Undefined variable -** *(Undefinierte Variable)*
965	**Unlicensed OS or Flash application -** *(BS oder Flash-Anwendung ohne Lizenz)*
970	**Variable in use so references or changes are not allowed -** *(Variable wird derzeit verwendet, deshalb kann sie weder an anderer Stelle aufgerufen noch geändert werden)*
980	**Variable is locked, protected, or archived -** *(Variable ist gesperrt, geschützt oder archiviert)*
990	**Variable name is limited to 8 characters -** *(Variablennamen sind auf 8 Zeichen beschränkt)*

Fehler-nummer	Beschreibung
1000	**Window variables domain -** *(Fenstervariablen-Bereich)*
1010	**Zoom -***(Zoom)*
	Warning: ∞^0 or undef^0 replaced by 1 - *(Warnung: ∞^0 oder undef^0 wird durch 1 ersetzt)*
	Warning: 0^0 replaced by 1 - *(Warnung: 0^0 wird durch 1 ersetzt)*
	Warning: 1^∞ or 1^undef replaced by 1 - *(Warnung: 1^∞ oder 1^ undef wird durch 1 ersetzt)*
	Warning: cSolve may specify more zeros - *(Warnung: cSolve könnte weitere Nullstellen berechnen)*
	Warning: May produce false equation - *(Warnung: Kann zu falscher Gleichung führen)*
	Warning: Expected finite real integrand - *(Warnung: Endlicher, reeller Integrand erwartet)*
	Warning: More solutions may exist - *(Warnung: Es können weitere Lösungen vorhanden sein)*
	Warning: May introduce false solutions - *(Warnung: Kann zu falschen Lösungen führen*
	Warning: Operation may lose solutions - *(Warnung: Operation könnte Lösungen verlieren)*
	Warning: Requires and returns 32 bit value - *(Warnung: Operation benötigt und liefert 32-Bit-Wert)*
	Warning: Overflow replaced by ∞ or ⁻∞ - *(Warnung: Bereichsüberschreitung wird durch ∞ oder ⁻∞ ersetzt)*
	Warning: Questionable accuracy - *(Warnung: Fragliche Genauigkeit)*
	Warning: Questionable solution - *(Warnung: Unsichere Lösung)*
	Warning: Solve may specify more zeros - *(Warnung: Solve könnte weitere Nullstellen berechnen)*
	Warning: Trig function argument too big for accurate reduction - *(Warnung: Argument der trigonometrischen Funktion zu umfangreich)*

TI-89 Titanium / Voyage™ 200 Modi

Graph

Im vorliegenden Abschnitt werden die Modi des TI-89 Titanium / Voyage™ 200 mit den möglichen Einstellungen aufgeführt. Die Moduseinstellungen können Sie durch Drücken von MODE anzeigen lassen.

1:FUNCTION	Funktionen y(x) (Kapitel 6)
2:PARAMETRIC	Parameterdarstellungen x(t) und y(t) (Kapitel 7)
3:POLAR	Polardarstellungen r(θ) (Kapitel 8)
4:SEQUENCE	Folgen u(n) (Kapitel 9)
5:3D	3D-Darstellungen z(x,y) (Kapitel 10)
6:DIFF EQUATIONS	y'(t) Differentialgleichungen (Kapitel 11)

Hinweis: Wenn Sie einen geteilten Bildschirm mit Number of Graphs = 2 benutzen, gilt Graph für den oberen bzw. den linken und Graph 2 für den unteren bzw. den rechten Bildschirmteil.

Current Folder

Gibt das aktuelle Verzeichnis an. Sie können mehrere Verzeichnisse mit eindeutigen Konfigurationen von Variablen, Graphik-Einstellungen, Programmen etc. einrichten.

Hinweis: *Nähere Informationen zu Verzeichnissen finden Sie in Kapitel 5.*

1:main	Vorgegebenes Verzeichnis des TI-89 Titanium / Voyage 200.
2: — (benutzerdefiniertes Verzeichnis)	Weiteres Verzeichnis, die jedoch vom Benutzer angelegt werden müssen, bevor sie hier angezeigt werden.

Display Digits

Legt die Anzahl der angezeigten Stellen fest. Diese Einstellung wirkt sich nur auf die Anzeige aus. Die Zahleneingabe kann in jedem beliebigen Format erfolgen.

Intern bearbeitet der TI-89 Titanium / Voyage 200 Dezimalzahlen mit 14 signifikanten Stellen. Für die Anzeige werden sie auf maximal 12 signifikante Stellen gerundet.

1:FIX 0 2:FIX 1 … D:FIX 12	Ergebnisse werden stets mit der ausgewählten Anzahl von Dezimalstellen angezeigt.
E:FLOAT	Die Anzahl der Dezimalstellen ist je nach Ergebnis unterschiedlich.
F:FLOAT 1 G:FLOAT 2 … Q:FLOAT 12	Besteht der ganzzahlige Teil aus mehr Stellen als ausgewählt wurden, wird das Ergebnis gerundet und in wissenschaftlicher Schreibweise angezeigt. Beispiel: Bei FLOAT 4: wird 12345. angezeigt als 1.235E4

Angle

Bestimmt die Einheit, in der Winkelwerte in trigonometrischen Funktionen und bei der Umwandlung polar-kartesisch interpretiert und angezeigt werden.

1:RADIAN
2:DEGREE

Exponential Format

Bestimmt die benutzte Notation. Das Format wirkt sich nur auf die Anzeige aus; die Zahleneingabe können Sie in einem beliebigen Format vornehmen. Numerische Antworten können mit bis zu 12 Stellen und einem dreistelligen Exponenten angezeigt werden.

1:NORMAL	Anzeige von Zahlen im Standardformat. Beispiel: 12345.67
2:SCIENTIFIC	Anzeige von Zahlen erfolgt zweiteilig: ¦ Eine der signifikanten Stellen wird links des Dezimaltrennzeichens angezeigt, die anderen rechts davon. ¦ Die Zehnerpotenz wird rechts des E angezeigt. Beispiel: 1.234567E4 bedeutet 1.234567×10^4
3:ENGINEERING	Ähnlich wie die vorstehende wissenschaftliche Schreibweise. Aber: ¦ Vor dem Dezimaltrennzeichen können eine, zwei oder drei Ziffern stehen. ¦ Der Zehnerpotenzexponent ist ein Vielfaches von 3. Beispiel: 12.34567E3 bedeutet 12.34567×10^3

Hinweis: Kann bei der Einstellung NORMAL die Antwort nicht mit der Anzahl Stellen angezeigt werden, die für Display Digits ausgewählt wurde, zeigt der TI-89 Titanium / Voyage™ 200 die Antwort in der Notation SCIENTIFIC an. Ist Display Digits = FLOAT, wird die wissenschaftliche Schreibweise für Exponenten von 12 und darüber sowie −4 und darunter benutzt.

Complex Format

Bestimmt, ob und in welchem Format komplexe Ergebnisse angezeigt werden.

1:REAL	Keine Anzeige komplexer Ergebnisse. (Ist ein Ergebnis eine komplexe Zahl und enthält die Eingabe nicht die Komplex-Einheit ***i***, wird eine Fehlermeldung angezeigt.)
2:RECTANGULAR	Zeigt komplexe Zahlen in der Form: a+b***i***
3:POLAR	Zeigt komplexe Zahlen in der Form: $re^{i\theta}$

Vector Format

Bestimmt, wie Vektoren mit 2 und 3 Elementen angezeigt werden. Sie können Vektoren in jedem der Koordinatensystemformate eingeben.

1:RECTANGULAR	Koordinatenwiedergabe als x, y und z. Beispiel: [3,5,2] gibt x = 3, y = 5 und z = 2 wieder.
2:CYLINDRICAL	Koordinatenwiedergabe als r, θ und z. Beispiel: [3,$\angle$45,2] gibt r = 3, θ = 45 und z = 2 wieder.
3:SPHERICAL	Koordinatenwiedergabe als r, θ und ϕ. Beispiel: [3, $\angle$45, $\angle$90] gibt r = 3, θ = 45 und ϕ = 90 wieder.

Pretty Print

Entscheidet, wie Ergebnisse im Hauptbildschirm angezeigt werden.

1:OFF	Lineare, eindimensionale Anzeige. Beispiel: π^2, π/2 oder $\sqrt{}$((x-3)/x)
2:ON	Üblichtes mathematisches Anzeigeformat. Beispiel: π^2, $\frac{\pi}{2}$ oder $\sqrt{\frac{x-3}{x}}$

Hinweis: Ausführliche Erläuterungen zu diesen Einstellungen finden Sie in “Formate angezeigter Ergebnisse” in Kapitel 2.

Split Screen

Dient zum Einrichten eines zweigeteilten Bildschirms. Beispiel: Sie können einen Graph und gleichzeitig den Y= Editor anzeigen (Kapitel 14).

1:FULL	Der Bildschirm wird nicht geteilt.
2:TOP-BOTTOM	Die Anwendungen werden auf zwei Bildschirmen angezeigt, die übereinander angeordnet sind.
3:LEFT-RIGHT	Die Anwendungen werden auf zwei Bildschirmen angezeigt, die nebeneinander angeordnet sind.

Dieser Modus legt die Grundeigenschaften fest. Die Art und Weise der Anzeige, die Anwendungen etc. legen Sie mit anderen Modi wie Split 1 App, Split 2 App, Number of Graphs, und Split Screen Ratio fest. (Die letztere Funktion ist nur auf dem Voyage™ 200 verfügbar).

Split 1 App und Split 2 App

Legt fest, welche Anwendung wo auf dem Bildschirm angezeigt wird.

- Bei ungeteiltem Bildschirm ist nur Split 1 App aktiv.
- Bei geteiltem Bildschirm ist Split 1 App im oberen bzw. linken Teil und Split 2 App im unteren bzw. im rechten Teil.

Die auswählbaren Anwendungen werden angezeigt, wenn Sie bei angezeigtem Page 2-Modus ▷ drücken oder wenn Sie APPS drücken. Sofern Sie nicht den 2-Graph-Modus benutzen, müssen unterschiedliche Anwendungen für beide Bildschirme ausgewählt werden.

Number of Graphs

Legt fest, ob in beiden Teilen eines geteilten Bildschirms gleichzeitig Graphiken angezeigt werden können.

1	Nur in einem Teil können Graphiken angezeigt werden.
2	Es können in beiden Teilen Graphiken angezeigt werden (Graph bzw. Graph 2), für die separate Einstellungen gelten.

Graph 2

Legt den Graphiktyp fest, der als zweiter auf einem 2-Graphen-Bildschirm angezeigt wird. Nur aktiv, wenn Number of Graphs = 2. Im 2-Graph-Modus legt Graph den Typ fest, der im oberen/linken Teil angezeigt werden kann und Graph 2 den unten/rechts angezeigten. Die Auswahlmöglichkeiten sind die gleichen wie für Graph.

Bildschirmteilung (nur Voyage™ 200)

Gibt das Größenverhältnis der Fenster bei Bildschirmteilung an.

1:1	Beide Fenster gleich groß.
1:2	Das untere bzw. rechte Fenster ist ungefähr doppelt so groß wie das obere bzw. das linke Fenster.
2:1	Das obere bzw. linke Fenster ist ungefähr doppelt so groß wie das untere bzw. das rechte Fenster.

Exact/Approx

Legt fest, wie Bruchterme und symbolische Terme berechnet und angezeigt werden. Bei der Einstellung EXACT werden rationale und symbolische Formen beibehalten, wodurch der TI-89 Titanium / Voyage 200 die Genauigkeit erhöht, da die meisten numerischen Rundungsfehler vermieden werden.

1:AUTO	Benutzt meist EXACT. Enthält der Eintrag jedoch ein Dezimaltrennzeichen, wird APPROXIMATE benutzt.
2:EXACT	Zeigt nicht-ganzzahlige Ergebnisse in ihrer rationalen oder symbolischen Form an.
3:APPROXIMATE	Zeigt numerische Ergebnisse in Gleitkommaform an.

Hinweis: Ausführliche Erläuterungen zu diesen Einstellungen finden Sie in “Formate angezeigter Ergebnisse” in Kapitel 2.

Base

Dient zur Durchführung von Berechnungen, für die Sie Zahlen in dezimaler, binärer oder hexadezimaler Form eingeben müssen.

1:DEC	Dezimalzahlen verwenden 0 - 9 im Format Basis 10
2:HEX	Hexadezimalzahlen verwenden 0 - 9 und A - F im Format Basis 16.
3:BIN	Dualzahlen verwenden 0 und 1 im Format Basis 2.

Unit System

Dient zur Eingabe einer Einheit für Werte in einem Term, wie z.B. 6_m * 4_m oder 23_m/_s * 10_s, Konvertieren der Werte von einer Einheit in eine andere derselben Kategorie und Erstellen von benutzerdefinierten Einheiten.

1:SI	Wählen Sie SI, um das metrische Einheitensystem einzustellen
2:ENG/US	Wählen Sie ENG/US für das nichtmetrische Einheitensystem
3:CUSTOM	Dient zum Wählen benutzerdefinierter Standardeinstellungen.

Custom Units

Ermöglicht die Auswahl benutzerdefinierter Standardeinstellungen. Dieser Modus ist unscharf bis Unit System, 3:CUSTOM gewählt wurde.

Language

Ermöglicht die Auswahl einer Arbeitssprache für TI-89 Titanium / Voyage™ 200 in Abhängigkeit von den installierten Flash-Sprachanwendungen.

1:Englisch	Standardsprache, Teil des Betriebssystems von TI-89 Titanium / Voyage 200.
2: — (Flash-Sprachanwendung)	Weitere Sprachen sind nur nach entsprechender Installation der Flash-Anwendung verfügbar.

Apps-Arbeitsfläche

Erlaubt Ihnen die Anzeige des Apps-Arbeitsfläches an oder aus zu schalten.

AN	Anzeigen des navigierbaren Apps- Arbeitsfläches. Der Apps-Arbeitsfläche wird angezeigt, wenn Sie: ¦ [APPS] drücken. ¦ Das Gerät einschalten, nachdem es mit [2nd] [OFF] abgeschaltet wurde. ¦ In einer App, die im Vollbildmodus angezeigt wird, [2nd] [QUIT] drücken.
AUS	Der navigierbare Apps-Arbeitsfläche wird nicht angezeigt. Die Voreinstellung des Geräts ist der Startbildschirm des Rechners. Der Startbildschirm des Rechners wird durch Drücken von [2nd] [QUIT] angezeigt. Das Menü APPLICATIONS wird angezeigt, wenn Sie [APPS] drücken.

Zeichen-Codes auf TI-89 Titanium / Voyage™ 200

Mit der Funktion **char()** können Sie jedes Zeichen mittels seines Zeichencodes angeben. Beispiel: Das Zeichen ◆ können Sie auf dem Programm-I/O-Bildschirm mit der Anweisung `Disp char(127)` anzeigen. Mit der Funktion **ord** können Sie den numerischen Code eines Zeichens ermitteln. Beispiel: `ord("A")` gibt den Wert 65 zurück.

1.	SOH	38.	&	75.	K	112.	p	149.	E	186.	º	223.	ß
2.	STX	39.	'	76.	L	113.	q	150.	*e*	187.	»	224.	à
3.	ETX	40.	(	77.	M	114.	r	151.	***i***	188.	*d*	225.	á
4.	EOT	41.	)	78.	N	115.	s	152.	ʳ	189.	∫	226.	â
5.	ENQ	42.	*	79.	O	116.	t	153.	ᵀ	190.	∞	227.	ã
6.	ACK	43.	+	80.	P	117.	u	154.	$\bar{x}$	191.	¿	228.	ä
7.	BELL	44.	,	81.	Q	118.	v	155.	$\bar{y}$	192.	À	229.	å
8.	BS	45.	−	82.	R	119.	w	156.	≤	193.	Á	230.	æ
9.	TAB	46.	.	83.	S	120.	x	157.	≠	194.	Â	231.	ç
10.	LF	47.	/	84.	T	121.	y	158.	≥	195.	Ã	232.	è
11.		48.	0	85.	U	122.	z	159.	∠	196.	Ä	233.	é
12.	FF	49.	1	86.	V	123.	{	160.	…	197.	Å	234.	ê
13.	CR	50.	2	87.	W	124.	|	161.	¡	198.	Æ	235.	ë
14.		51.	3	88.	X	125.	}	162.	¢	199.	Ç	236.	ì
15.	✓	52.	4	89.	Y	126.	~	163.	£	200.	È	237.	í
16.	▪	53.	5	90.	Z	127.	◆	164.	¤	201.	É	238.	î
17.	◄	54.	6	91.	[	128.	α	165.	¥	202.	Ê	239.	ï
18.	►	55.	7	92.	\	129.	β	166.	¦	203.	Ë	240.	ð
19.	▲	56.	8	93.	]	130.	Γ	167.	§	204.	Ì	241.	ñ
20.	▼	57.	9	94.	^	131.	γ	168.	√	205.	Í	242.	ò
21.	←	58.	:	95.	_	132.	Δ	169.		206.	Î	243.	ó
22.	→	59.	;	96.	`	133.	δ	170.	ª	207.	Ï	244.	ô
23.	↑	60.	<	97.	a	134.	ε	171.	«	208.	Đ	245.	õ
24.	↓	61.	=	98.	b	135.	ζ	172.	¬	209.	Ñ	246.	ö
25.		62.	>	99.	c	136.	θ	173.	-	210.	Ò	247.	÷
26.		63.	?	100.	d	137.	λ	174.	®	211.	Ó	248.	ø
27.		64.	@	101.	e	138.	ξ	175.	¯	212.	Ô	249.	ù
28.	∪	65.	A	102.	f	139.	Π	176.	°	213.	Õ	250.	ú
29.	∩	66.	B	103.	g	140.	π	177.	±	214.	Ö	251.	û
30.	⊂	67.	C	104.	h	141.	ρ	178.	²	215.	×	252.	ü
31.	∈	68.	D	105.	i	142.	Σ	179.	³	216.	Ø	253.	ý
32.	SPACE	69.	E	106.	j	143.	σ	180.	⁻¹	217.	Ù	254.	þ
33.	!	70.	F	107.	k	144.	τ	181.	µ	218.	Ú	255.	ÿ
34.	"	71.	G	108.	l	145.	φ	182.	¶	219.	Û		
35.	#	72.	H	109.	m	146.	ψ	183.	·	220.	Ü		
36.	$	73.	I	110.	n	147.	Ω	184.	⁺	221.	Ý		
37.	%	74.	J	111.	o	148.	ω	185.	¹	222.	Þ		

Tasten-Codes auf TI-89

Die Funktion **getKey()** gibt einen Wert zurück, der der zuletzt gedrückten Taste entspricht. Die Werte sind in den folgenden Tabellen zusammengestellt. Beispiel: Enthält ein Programm **getKey()** und drücken Sie [2nd] [F6], wird der Wert 273 zurückgegeben.

Tabelle 1: Codes für primäre Tasten

Taste	Modifikatortaste									
	Keine		[↑]		[2nd]		[◆]		[alpha]	
	Zugeor.	Wert	Zugeor.	Wert	Zugeor.	Wert	Zugeor.	Wert	Zugeor.	Wert
[F1]	F1	268	F1	268	F6	273	Y=	8460	F1	268
[F2]	F2	269	F2	269	F7	274	WINDOW	8461	F2	269
[F3]	F3	270	F3	270	F8	275	GRAPH	8462	F3	270
[F4]	F4	271	F4	271	F4	271	TblSet	8463	F4	271
[F5]	F5	272	F5	272	F5	272	TABLE	8464	F5	272
[◆]			COPY	24576	CUT	12288				
[alpha]					a-lock					
[ESC]	ESC	264	ESC	264	QUIT	4360	PASTE	8456	ESC	264
[APPS]	APPS	265	APPS	265	SWITCH	4361		8457	APPS	265
[HOME]	HOME	277	HOME	277	CUST	4373	HOME	277	HOME	277
[MODE]	MODE	266	MODE	266	▸	18	_	95	MODE	266
[CATALOG]	CATLG	278	CATLG	278	*i*	151	∞	190	CATLG	278
[←]	BS	257	BS	257	INS	4353	DEL	8449	BS	257
[CLEAR]	CLEAR	263	CLEAR	263	CLEAR	263		8455	CLEAR	263
[X]	x	120	X	88	LN	4184	e^x	8280	x	120
[Y]	y	121	Y	89	SIN	4185	SIN^{-1}	8281	y	121
[Z]	z	122	Z	90	COS	4186	COS^{-1}	8282	z	122
[T]	t	116	T	84	TAN	4180	TAN^{-1}	8276	t	116
[^]	^	94	^	94	π	140	θ	136	^	94
[|]	|	124	F	70	°	176	Format d/b	8316	f	102
[(]	(	40	B	66	{	123			b	98
[)]	)	41	C	67	}	125	◉	169	c	99
[,]	,	44	D	68	[	91		8236	d	100
[÷]	/	47	E	69	]	93	!	33	e	101
[×]	*	42	J	74	√	4138	&	38	j	106
[−]	-	45	O	79	VAR-LNK	4141	Contr. -		o	111
[+]	+	43	U	85	CHAR	4139	Contr. +		u	117

Tabelle 1: Codes für primäre Tasten (Forts.)

Taste	Modifikatortaste									
	Keine		↑		2nd		◆		alpha	
	Zugeor.	Wert	Zugeor.	Wert	Zugeor.	Wert	Zugeor.	Wert	Zugeor.	Wert
ENTER	CR	13	CR	13	ENTRY	4109	APPROX	8205	CR	13
STO▸	STO▸	258	P	80	RCL	4354	@	64	p	112
=	=	61	A	65	'	39	≠	157	a	97
EE	EE	149	K	75	∠	159	SYMB	8341	k	107
(-)	-	173	SPACE	32	ANS	4372		8365	SPACE	32
.	.	46	W	87	>	62	≥	158	w	119
0	0	48	V	86	<	60	≤	156	v	118
1	1	49	Q	81	"	34		8241	q	113
2	2	50	R	82	\	92		8242	r	114
3	3	51	S	83	UNITS	4147		8243	s	115
4	4	52	L	76	:	58		8244	l	108
5	5	53	M	77	MATH	4149		8245	m	109
6	6	54	N	78	MEM	4150		8246	n	110
7	7	55	G	71	∫	4151		8247	g	103
8	8	56	H	72	d	4152		8248	h	104
9	9	57	I	73	;	59		8249	i	105

Tabelle 2: Pfeiltasten (inkl. diagonale Cursorsteuerung)

Taste	Normal	↑	2nd	◆	alpha
▲	338	16722	4434	8530	33106
▶	340	16724	4436	8532	33108
▼	344	16728	4440	8536	33112
◀	337	16721	4433	8529	33105
▲ und ◀	339	16723	4435	8531	33107
▲ und ▶	342	16726	4438	8534	33110
▼ und ◀	345	16729	4441	8537	33113
▼ und ▶	348	16732	4444	8540	33116

Tabelle 3: Griechische Buchstaben

(vor der Taste des Buchstabens ◆ (drücken)

Taste	Zweite Modifikatortaste			
	alpha		↑	
	Zugeor.	**Wert**	**Zugeor.**	**Wert**
= [A]	α	128		
([B]	β	129		
, [D]	δ	133	Δ	132
÷ [E]	ε	134		
) [F]	ϕ	145		
7 [G]	γ	131	Γ	130
4 [L]	λ	137		
5 [M]	μ	181		
STO▸ [P]	π	140	Π	139
2 [R]	ρ	141		
3 [S]	σ	143	Σ	142
T [T]	τ	144		
. [W]	ω	148	Ω	147
X	ξ	138		
Y	ψ	146		
Z	ζ	135		

Tasten-Codes auf Voyage™ 200

Die Funktion **getKey()** gibt einen Wert zurück, der der zuletzt gedrückten Taste entspricht. Die Werte sind in den folgenden Tabellen zusammengestellt. Beispiel: Enthält ein Programm **getKey()** und drücken Sie [2nd] [F1], wird der Wert 268 zurückgegeben.

Tabelle 1: Codes für die primären Tasten

Taste	Modifikatortaste							
	Keine		[↑]		[2nd]		[♦]	
	Zugeor.	Wert	Zugeor.	Wert	Zugeor.	Wert	Zugeor.	Wert
[F1]	F1	268	F1	268	F1	268		8460
[F2]	F2	269	F2	269	F2	269		8461
[F3]	F3	270	F3	270	F3	270		8462
[F4]	F4	271	F4	271	F4	271		8463
[F5]	F5	272	F5	272	F5	272		8464
[F6]	F6	273	F6	273	F6	273		8465
[F7]	F7	274	F7	274	F7	274		8466
[F8]	F8	275	F8	275	F8	275		8467
[MODE]	MODE	266	MODE	266	MODE	266		8458
[CLEAR]	CLEAR	263	CLEAR	263	CLEAR	263		8455
[LN]	LN	262	LN	262	e^x	4358		8454
[ESC]	ESC	264	ESC	264	QUIT	4360		8456
[APPS]	APPS	265	APPS	265	SWITCH	4361		8457
[ENTER]	CR	13	CR	13	ENTRY	4109	APPROX	8205
[SIN]	SIN	259	SIN	259	SIN^{-1}	4355		8451
[COS]	COS	260	COS	260	COS^{-1}	4356		8452
[TAN]	TAN	261	TAN	261	TAN^{-1}	4357		8453
[^]	^	94	^	94	π	140		8286
[(]	(	40	(	40	{	123		8232
[)]	)	41	)	41	}	125		8233
[,]	,	44	,	44	[	91		8236
[÷]	/	47	/	47	]	93		8239
[×]	*	42	*	42	√	4138		8234
[−]	-	45	-	45	VAR-LNK	4141	Contrast −	
[+]	+	43	+	43	CHAR	4139	Contrast +	
[STO▸]	STO▸	258	STO▸	258	RCL	4354		8450
SPACE		32		32		32		8224
[=]	=	61	=	61	\	92		8253
[←]	BS	257	BS	257	INS	4353	DEL	8449
[θ]	θ	136	θ	136	:	58		8328
[(-)]	-	173	-	173	ANS	4372		8365
[.]	.	46	.	46	>	62		8238

Tabelle 1: Codes für die primären Tasten (Fortsetzung)

Taste	Modifikatortaste							
	Keine		⬆		2nd		♦	
	Zugeor.	Wert	Zugeor.	Wert	Zugeor.	Wert	Zugeor.	Wert
0	0	48	0	48	<	60		8240
1	1	49	1	49	E	149		8241
2	2	50	2	50	CATALOG	4146		8242
3	3	51	3	51	CUST	4147		8243
4	4	52	4	52	Σ	4148		8244
5	5	53	5	53	MATH	4149		8245
6	6	54	6	54	MEM	4150		8246
7	7	55	7	55	∫	4151		8247
8	8	56	8	56	*d*	4152		8248
9	9	57	9	57	x^{-1}	4153		8249
A	a	97	A	65	Table 3			8257
B	b	98	B	66	‘	39		8258
C	c	99	C	67	Table 4		COPY	8259
D	d	100	D	68	°	176		8260
E	e	101	E	69	Table 5		WINDOW	8261
F	f	102	F	70	∠	159	FORMAT	8262
G	g	103	G	71	Table 6			8263
H	h	104	H	72	&	38		8264
I	i	105	I	73	i	151		8265
J		106	J	74	∞	190		8266
K	k	107	K	75	|	124	TASTE	8267
L	l	108	L	76	“	34		8268
M	m	109	M	77	;	59		8269
N	n	110	N	78	Table 7		NEW	8270
O	o	111	O	79	Table 8		OPEN	8271
P	p	112	P	80	_	95	EINHEITEN	8272
Q	q	113	Q	81	?	63	CALC HOME	8273
R	r	114	R	82	@	64	GRAPH	8274
S	s	115	S	83	β	223	SAVE	8275
T	t	116	T	84	#	35	TBLSET	8276
U	u	117	U	85	Table 9			8277
V	v	118	V	86	≠	157	PASTE	8278
W	w	119	W	87	!	33	Y=	8279
X	x	120	X	88	◉	169	CUT	8280
Y	y	121	Y	89	▶	18	TABLE	8281
Z	z	122	Z	90	CAPS			8282

Tabelle 2: Pfeiltasten (inklusive diagonaler Bewegung)

Taste	Normal	⇧	2nd	◆	⎋
↑	338	16722	4434	8530	33106
→	342	16726	4438	8534	33110
↓	340	16724	4436	8532	33108
←	348	16732	4444	8540	33116
↑ und ←	344	16728	4440	8536	33112
↑ und →	345	16729	4441	8537	33113
↓ und ←	337	16721	4433	8529	33105
↓ und →	339	16723	4435	8531	33107

Hinweis: Die Zieh-Taste (⎋)wirkt sich nur auf die Pfeiltasten aus.

Tabelle 3: Gravis Accent Buchstaben mit (zuerst 2nd A drücken)

Taste	Zugeord.	Normal	⇧
A	à	224	192
E	è	232	200
I	ì	236	204
O	ò	242	210
U	ù	249	217

Tabelle 4: Buchstaben mit Cedille (zuerst 2nd C drücken)

Taste	Zugeord.	Normal	⇧
C	ç	231	199

Tabelle 5: Acute Accent Buchstaben mit (zuerst 2nd E drücken)

Taste	Zugeord.	Normal	⇧
A	á	225	193
E	é	233	201
I	í	237	205
O	ó	243	211
U	ú	250	218
Y	ý	253	221

Tabelle 6: Griechische Buchstaben (zuerst 2nd G drücken)

Taste	Zugeord.	Normal	⬆
A	α	128	
B	β	129	
D	δ	133	132
E	ε	134	
F	ϕ	145	
G	γ	131	130
L	λ	137	
M	μ	181	
P	π	140	139
R	ρ	141	
S	σ	143	142
T	τ	144	
W	ω	148	147
X	ξ	138	
Y	ψ	146	
Z	ζ	135	

Tabelle 7: Tilde Buchstaben (zuerst 2nd N drücken)

Taste	Zugeord.	Normal	⬆
N	ñ	241	209
O	õ	245	

Tabelle 8: Caret Buchstaben (zuerst 2nd O drücken)

Taste	Zugeord.	Normal	⬆
A	â	226	194
E	ê	234	202
I	î	238	206
O	ô	244	212
U	û	251	219

Tabelle 9: Umlaute (zuerst 2nd U drücken)

Taste	Zugeord.	Normal	⬆
A	ä	228	196
E	ë	235	203
I	ï	239	207
O	ö	246	214
U	ü	252	220
Y	ÿ	255	

Komplexe Zahlen eingeben

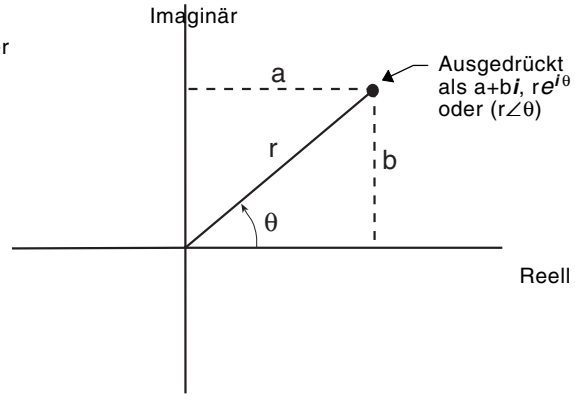
Sie können komplexe Zahlen in der polaren Form ($r\angle\theta$) eingeben, wobei r der Betrag und θ der Winkel ist, oder in der polaren Form $re^{i\theta}$. Außerdem können komplexe Zahlen in der kartesischen Form $a+b\boldsymbol{i}$ eingegeben werden.

Übersicht über komplexe Zahlen

Eine komplexe Zahl besitzt eine reelle und eine imaginäre Komponente, die einen Punkt in der komplexen Ebene identifizieren. Die Komponenten werden entlang der reellen und der imaginären Achse gemessen, die der x- und y-Achse in der reellen Ebene ähnlich sind .

Der Punkt kann in kartesischer oder einer der beiden polaren Formen ausgedrückt werden.

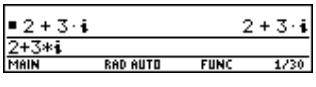
Das Symbol ***i*** identifiziert die imaginäre Einheit $\sqrt{-1}$.



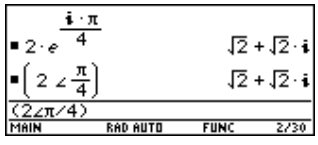
Wie im folgenden dargestellt, hängt die mögliche Eingabeform vom aktuellen Angle-Modus ab.

Sie können folgende Form verwenden:	Wenn Angle eingestellt ist auf:
a+b***i***	Radian oder Degree
$re^{i\theta}$	nur Radian (Im Winkelmodus Degree verursacht diese Form den Fehler Domain error).
($r\angle\theta$)	Radian oder Degree

Geben Sie komplexe Zahlen folgendermaßen ein.

Zur Eingabe dieser Form:	Benutzen Sie diese Tastenfolge:
Kartesisch a+b***i***	Ersetzen Sie a und b durch die gewünschten Werte bzw. Variablennamen. a [+] b [2nd] [i] Beispiel: 

Hinweis: *Das Symbol **i** wird mit [2nd] [i] erzeugt, nicht durch Drücken der Taste i auf der Tastatur.*

Für die Eingabe von:	gehen Sie folgendermaßen vor:
Polar $re^{i\theta}$ – oder – $(r\angle\theta)$ Für die Form $(r\angle\theta)$ sind Klammern erforderlich.	Ersetzen Sie r und θ durch die gewünschten Werte bzw. Variablennamen, wobei θ gemäß dem eingestellten Winkelmodus interpretiert wird. **TI-89 Titanium:** alpha [R] ◆ [e^x] 2nd [i] ◆ [θ]) – oder – (alpha [R] 2nd [∠] ◆ [θ]) **Voyage 200:** R 2nd [e^x] 2nd [i] θ) – oder – (R 2nd [∠] θ) Beispiel:  Die Ergebnisse werden in kartesischer Form angezeigt, Sie können aber auch polare Form wählen.

Wichtig: *Verwenden Sie die polare Form $re^{i\theta}$ nicht im Winkelmodus* Degree. *Dies führt zu einem* Domain error.

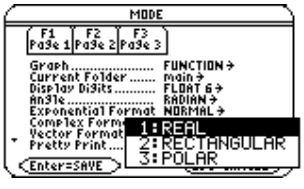
Hinweis: *Das Symbol e wird wie folgt erzeugt:*
TI-89 Titanium: ◆ [e^x].
Voyage™ 200: 2nd [e^x]
e auf der Tastatur genügt in diesem Fall nicht.

Tipp: *Für ∠ drücken Sie* 2nd [∠].

Tipp: *Zur Eingabe von θ in Grad für $(r\angle\theta)$ können Sie das Symbol ° eingeben (z.B. 45°). Das Symbol ° erhalten Sie durch Drücken von* 2nd [°]. *Sie sollten Grad nicht für $re^{i\theta}$ verwenden.*

Der Modus Complex Format für die Anzeige von Ergebnissen

Mit der Taste MODE können Sie den Modus Complex Format auf eine von drei Einstellungen setzen.



Eingeben können Sie komplexe Zahlen jederzeit und unabhängig von der Einstellung für Complex Format. Die Moduseinstellung gilt nur für die Anzeige von Ergebnissen.

Hinweis: *Je nach Winkelmodus können Sie komplexe Zahlen in jeder Form (oder einer Mischung aller Formen) eingeben.*

Einstellung für Complex Format ist:	Der TI-89 Titanium / Voyage 200:
REAL	Zeigt keine Ergebnisse in komplexer Form an, es sei denn: ¦ Sie geben für eine Berechnung eine komplexe Zahl ein. – oder – ¦ Sie verwenden eine spezielle komplexe Funktion **cFactor()**, **cSolve()**, or **cZeros()**. Komplexe Ergebnisse werden entweder in der Form a+b***i*** or $re^{i\theta}$ angezeigt.
RECTANGULAR	Zeigt komplexe Ergebnisse als a+b***i*** an.
POLAR	Zeigt komplexe Ergebnisse wie folgt an: ¦ $re^{i\theta}$, wenn Angle = Radian – oder – ¦ $(r\angle\theta)$, wenn Angle = Degree

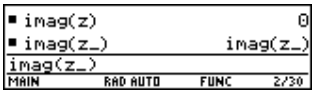
Komplexe Variablen in symbolischen Rechnungen verwenden

Unabhängig von der Moduseinstellung für Complex Format werden alle undefinierten Variablen als reelle Zahlen behandelt Zur Durchführung komplexer symbolischer Berechnungen müssen Sie eine komplexe Variable definieren. Hierfür haben Sie zwei Möglichkeiten:

Methode 1: Verwenden Sie zur Kennzeichnung von komplexen Variablen als letztes Zeichen des Variablennamens einen Unterstrich _ (**TI-89 Titanium:** ♦ [_]; **Voyage™ 200:** 2nd [_]). Beispiel:

Hinweis: *Beste Ergebnisse bei Berechnungen wie* **cSolve()** *und* **cZeros()** *erzielen Sie mit Methode 1.*

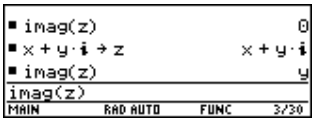
z_ wird als komplexe Variable behandelt (es sei denn, z ist bereits vorhanden; in diesem Fall bleibt z in der vorhandenen Datenart bestehen).



Methode 2: Definieren Sie eine komplexe Variable. Beispiel:

x+y*i*→z

Dann wird z als komplexe Variable behandelt.



Komplexe Zahlen und der Winkelmodus Degree

Hinweis: *Im Winkelmodus* Degree *müssen Sie die polare Form (r∠θ) verwenden. Im Modus* Degree *verursacht eine Eingabe der Form* $re^{i\theta}$ *einen Fehler.*

Radian wird als Winkelmodus für Berechnungen mit komplexen Variablen empfohlen. Der TI-89 Titanium / Voyage 200 konvertiert intern alle eingegebenen Trigonomiewerte in Radian. Die Werte für exponentielle, logarithmische oder hyperbolische Funktionen konvertiert er nicht.

Im Winkelmodus Degree sind komplexe Identitäten wie $e^{(i\theta)} = \cos(\theta) + i\sin(\theta)$ nicht generell wahr, da die Werte für cos und sin in Radian konvertiert werden, die für $e^{(\ )}$ aber nicht. So wird z.B. $e^{(i45)} = \cos(45) + i\sin(45)$ intern als $e^{(i45)} = \cos(\pi/4) + i\sin(\pi/4)$ behandelt. Komplexe Identitäten sind hingegen im Winkelmodus Radian immer wahr.

Hinweise zur Verarbeitungs- und Darstellungsgenauigkeit

Zur Erzielung höchster Genauigkeit arbeitet der TI-89 Titanium / Voyage™ 200 intern mit mehr Stellen als auf dem Display angezeigt werden.

Rechengenauigkeit

Gleitkommawerte (Dezimalwerte) werden im Speicher mit bis zu 14 Stellen mit einem 3-stelligen Exponenten gespeichert.

- Für die Fenstervariablen min und max (xmin, xmax, ymin, ymax etc.) können Sie Werte von bis zu 12 Stellen Länge speichern. Für die anderen Fenstervariablen werden 14 Stellen benutzt.
- Zur Anzeige eines Gleitkommawerts wird der Anzeigewert gemäß der jeweiligen Moduseinstellungen (Display Digits, Exponential Format etc.) auf maximal 12 Stellen und einen dreistelligen Exponenten gerundet.
- RegEQ zeigt Koeffizienten mit bis zu 14 Stellen Länge an.

Ganzzahlige Werte werden im Speicher mit einer Länge von bis zu 614 Stellen gespeichert.

Genauigkeit der graphischen Darstellung

Hinweis: *Eine Tabelle mit der Anzahl der Pixel bei geteiltem und ungeteiltem Bildschirm finden Sie in "Den Split- Screen-Modus einstellen und beenden" in Kapitel 14.*

Die Fenstervariable xmin gibt den Mittelpunkt des am weitesten links befindlichen benutzten Pixels wieder, und xmax ist die Mitte des am weitesten rechts befindlichen benutzten Pixels. Δx ist der Abstand zwischen den Mittelpunkten zweier Pixel in der Horizontalen.

- Δx wird berechnet als (xmax − xmin) / (Anzahl x-Pixel − 1).
- Wird Δx auf dem Hauptbildschirm oder in einem Programm eingegeben, wird xmax berechnet als xmin + Δx ∗ (Anzahl x-Pixel − 1).

Die Fenstervariable ymin gibt den Mittelpunkt des untersten benutzten Pixels wieder, und ymax ist die Mitte des obersten benutzten Pixels. Δy ist der Abstand zwischen den Mittelpunkten zweier Pixel in der Vertikalen.

- Δy wird berechnet als (ymax − ymin) / (Anzahl y-Pixel − 1).
- Wird Δy auf dem Hauptbildschirm oder in einem Programm eingegeben, wird xmax berechnet als ymin + Δy ∗ (Anzahl y-Pixel − 1).

Cursorkoordinaten werden acht Zeichen lang angezeigt (wobei dies ein negatives Vorzeichen, einen Dezimalpunkt und einen Exponenten einschließen kann). Die Koordinatenwerte (xc, yc, zc etc.) werden mit maximal 12-stellinger Genauigkeit aktualisiert.

Systemvariablen und reservierte Namen

In diesem Abschnitt sind die Systemvariablen und die reservierten Funktionsnamen des TI-89 Titanium / Voyage™ 200 zusammengestellt. Nur Systemvariablen, die mit einem Stern (*) markiert sind, können in der Eingabezeile mit dem Befehl DelVar *Variable* gelöscht werden.

Graphen

y1(x)–y99(x)*	y1'(t)–y99'(t)*	yi1–yi99*	r1(θ)–r99(θ)*
xt1(t)–xt99(t)*	yt1(t)– yt99(t)*	z1(x,y)–z99(x,y)*	u1(n)–u99(n)*
ui1–ui99*	xc	yc	zc
tc	rc	θc	nc
xfact	yfact	zfact	xmin
xmax	xscl	xgrid	ymin
ymax	yscl	ygrid	xres
Δx	Δy	zmin	zmax
zscl	eyeθ	eyeϕ	eyeψ
ncontour	θmin	θmax	θstep
tmin	tmax	tstep	t0
tplot	ncurves	diftol	dtime
Estep	fldpic	fldres	nmin
nmax	plotStrt	plotStep	sysMath

Graphik-Vergrößerung/-Verkleinerung

zxmin	zxmax	zxscl	zxgrid
zymin	zymax	zyscl	zygrid
zxres	zθmin	zθmax	zθstep
ztmin	ztmax	ztstep	zt0de
ztmaxde	ztstepde	ztplotde	zzmin
zzmax	zzscl	zeyeθ	zeyeϕ
zeyeψ	znmin	znmax	zpltstrt
zpltstep			

Statistik

$\overline{x}$	$\overline{y}$	Σx	σx
Σx^2	Σxy	Σy	σy
Σy^2	corr	maxX	maxY
medStat	medx1	medx2	medx3
medy1	medy2	medy3	minX
minY	nStat	q1	q3
regCoef*	regEq(x)*	seed1	seed2
Sx	Sy	R^2	

Tabelle

tblStart	Δtbl	tblInput

Daten/Matrix

c1–c99	sysData*

Sonstige

main	ok	errornum

Gleichunglöser

eqn*	exp*

Die Auswertungrangfolge des EOS™ (Equation Operating System)

In diesem Abschnitt wird das Equation Operating System (EOS™) des TI-89 Titanium / Voyage™ 200 erläutert. Zahlen, Variablen und Funktionen geben Sie in einfacher, leicht verständlicher Folge ein. Das EOS wertet Terme und Gleichungen anhand der nachfolgend aufgeführten und erläuterten Prioritäten aus.

Auswertungs-rangfolge

Priorität	Operator
1	Runde ( ), eckige[ ] und geschweifte { } Klammern
2	Umleitung (#)
3	Funktionsaufrufe
4	Nachgestellte Operatoren: Grad-Minuten-Sekunden (°,',"), Fakultät (!), Prozent (%), Bogenmaß (ᵣ), Indizierung ([ ]), Transponieren (ᵀ)
5	Exponenten, Potenzoperator (^)
6	Negation (⁻)
7	Konkatenation (&)
8	Multiplikation (∗), Division (/)
9	Addition (+), Subtraktion (–)
10	Gleichheit: gleich (=), ungleich (≠ oder /=), kleiner als (<), kleiner gleich (≤ oder <=), größer als (>), größer gleich (≥ oder >=)
11	Logisches nicht, **not()**
12	Logisches und, **and**
13	Logisches oder, **or**, exklusives logisches oder, **xor**
14	Einschränkungsoperator “with”, (|)
15	Speichern (→)

Runde, eckige und geschweifte Klammern

Alle Berechnungen innerhalb eines Paars runder, eckiger oder geschweifter Klammern werden zuerst ausgewertet. Beispiel: Im Term 4(1+2) wertet das EOS zuerst den Teil innerhalb der Klammern aus, also 1+2, und multipliziert anschließend das Ergebnis 3 mit 4.

Die Anzahl der öffnenden Klammern (runde, eckige, geschweifte) muß innerhalb des Terms oder der Gleichung mit der Anzahl der schließenden Klammern übereinstimmen. Andernfalls wird eine Fehlermeldung angezeigt, die auf das fehlende Element hinweist. Beispiel: (1+2)/(3+4 bewirkt die Ausgabe der Meldung “Missing).”

Hinweis: Da Sie mit dem TI-89 Titanium / Voyage 200 eigene Funktionen definieren können, wird ein Variablenname mit einem anschließenden Ausdruck in runden Klammern als "Funktionsaufruf" interpretiert und nicht als implizite Multiplikation. So ist z. B. a(b+c) die Funktion a ausgewertet nach den Parametern b+c. Zum Multiplizieren des Terms b+c mit der Variablen a müssen Sie die explizite Multiplikation verwenden: a∗(b+c).

Umleitung

Der Umleitungsoperator (#) wandelt eine Zeichenfolge (String) in einen Variablen- oder einen Funktionsnamen um. Beispiel: #(“x”&”y”&”z”) erzeugt den Variablenamen xyz. Der Umleitungsoperator ermöglicht auch die Erzeugung und Änderung von Variablen aus einem Programm heraus. Beispiel: if 10→r and “r”→s1 then #s1=10.

Nachgestellte Operatoren

Nachgestellte Operatoren sind solche, die unmittelbar auf einen Parameter folgen, etwa 5!, 25% oder 60° 15' 45". Parameter, denen ein nachgestellter Operator folgt, werden mit Prioritätsstufe 4 ausgewertet. Beispiel: Im Term 4^3! wird zuerst 3! ausgewertet. Das Ergebnis 6 wird dann zum Exponenten von 4, was als Endergebnis 4096 ergibt.

Exponenten

Exponentenoperationen (^) und elementweise Exponentenoperationen (.^) werden von rechts nach links ausgewertet. Beispiel: 2^3^2 wird wie 2^(3^2) ausgewertet und ergibt 512. Dies ist nicht das gleiche wie (2^3)^2, dessen Ergebnis 64 ist.

Negation

Zum Eingeben einer negativen Zahl drücken Sie [(-)] und geben dann die Zahl ein. Nachgestellte Operatoren und Exponentialoperationen werden vor der Negation ausgeführt. Beispiel: Das Ergebnis von $^{-}x^2$ ist eine negative Zahl, und $^{-}9^2 = {}^{-}81$. Benutzen Sie runde Klammern, um das Quadrat einer negativen Zahl wie $(^{-}9)^2$ mit dem positiven Ergebnis 81 zu erhalten. Beachten Sie auch: die negative Zahl minus 5 ($^{-}5$) ist nicht dasselbe wie Minusoperator 5 (−5), und $^{-}3!$ wird ausgewertet als $^{-}(3!)$.

Einschränkung (|)

Der auf den Operator “with” (|) folgende Parameter gibt Einschränkungen an, die die Auswertung des Parameters beeinflussen, der sich vor dem “with”-Operator befindet.

Regressionsformeln

In diesem Abschnitt wird beschrieben, wie statistische Regressionen berechnet werden.

Methode der kleinsten Quadrate

In den meisten Regressionen wird eine nichtlineare rekursive Technik der kleinsten Quadrate angewendet, um folgende Kostenfunktion zu optimieren, bei welcher es sich um die Summe der Quadrate des Residuums handelt:

$$J = \sum_{i=1}^{N} \left[residualExpression \right]^2$$

Hierbei gilt: *residualExpression* bezüglich x_i und y_i
x_i ist die Liste der unabhängigen Variablen
y_i ist die Liste der abhängigen Variablen
N ist die Listendimension.

Mit dieser Technik wird versucht, die Konstante im Modellterm rekursiv zu schätzen, um ein kleinstmögliches *J* zu erhalten.

Beispiel: *y=a sin(bx+c)+d* ist die Modellgleichung für **SinReg**. Das Residuum ist also:

$a\ sin(bx_i+c)+d-y_i$

Mit der Methode der kleinsten Quadrate werden für **SinReg** deshalb die Konstanten *a*, *b*, *c* und *d* gefunden, welche folgende Funktion minimieren:

$$J = \sum_{i=1}^{N} \left[a\ sin(bx_i + c) + d - y_i \right]^2$$

Regressionen

Regression	Beschreibung
CubicReg	Verwendet die Methode der kleinsten Quadrate, um an folgendes Polynom dritten Grades anzugleichen: $y=ax^3+bx^2+cx+d$ Bei vier Datenpunkten ist die Gleichung ein Polynom; bei fünf oder mehr Datenpunkten ist sie eine polynomische Regression. Es sind mindestens vier Datenpunkte erforderlich.
ExpReg	Verwendet die Methode der kleinsten Quadrate und die transformierten Werte *x* und ln(*y*), um an folgende Modellgleichung anzugleichen: $y=ab^x$
LinReg	Verwendet die Methode der kleinsten Quadrate, um an folgende Modellgleichung anzugleichen: $y=ax+b$ wobei *a* die Steigung und *b* der y-Achsenabschnitt ist.

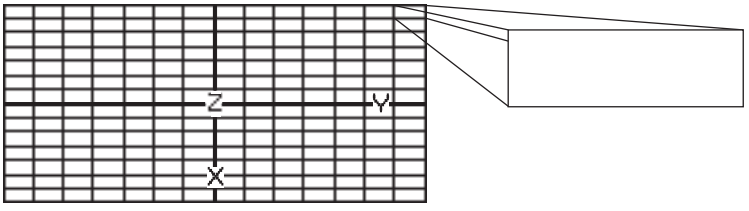
Regression	Beschreibung
LnReg	Verwendet die Methode der kleinsten Quadrate und die transformierten Werte ln(*x*) und *y*, um an folgende Modellgleichung anzugleichen: $y=a+b\ln(x)$
Logistic	Verwendet die Methode der kleinsten Quadrate, um an folgende Modellgleichung anzugleichen: *y=a/(1+b*e^(c*x))+d*
MedMed	Verwendet die Methode der Median-Median-Geraden zur Berechnung der statistischen Häufungspunkte x1, y1, x2, y2, x3 und y3, und gleicht an folgende Modellgleichung an: $y=ax+b$ wobei *a* die Steigung und *b* der y-Achsenabschnitt ist.
PowerReg	Verwendet die Methode der kleinsten Quadrate und die transformierten Werte ln(*x*) und ln(*y*), um an folgende Modellgleichung anzugleichen: $y=ax^b$
QuadReg	Verwendet die Methode der kleinsten Quadrate, um an folgendes Polynom zweiten Grades anzugleichen: $y=ax^2+bx+c$ Bei drei Datenpunkten ist die Gleichung ein Polynom; bei vier oder mehr Datenpunkten ist sie eine polynomische Regression. Es sind mindestens drei Datenpunkte erforderlich.
QuartReg	Verwendet die Methode der kleinsten Quadrate, um an folgendes Polynom vierten Grades anzugleichen: $y=ax^4+bx^3+cx^2+dx+e$ Bei fünf Datenpunkten ist die Gleichung ein Polynom; bei sechs oder mehr Datenpunkten ist sie eine polynomische Regression. Es sind mindestens fünf Datenpunkte erforderlich.
SinReg	Verwendet die Methode der kleinsten Quadrate, um an folgende Modellgleichung anzugleichen: *y=a sin(bx+c)+d*

Die Algorithmen “Contour Levels” und “Implicit Plot”

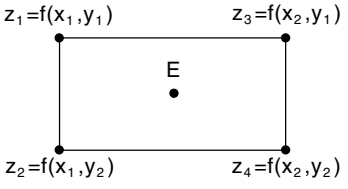
Konturen werden berechnet und geplottet wie nachfolgend beschrieben. Ein impliziter Plot ist gleichbedeutend mit einer Kontur. Der einzige Unterschied besteht darin, daß ein impliziter Plot nur für die Kontur z=0 möglich ist.

Algorithmus

Der Abstand zwischen xmin und xmax sowie zwischen ymin und ymax ist auf Grundlage Ihrer Fenstervariablen x und y in eine durch xgrid und ygrid angegebene Anzahl von Gitterlinien unterteilt. Durch das Überschneiden dieser Gitterlinien entsteht eine Folge von Rechtecken.



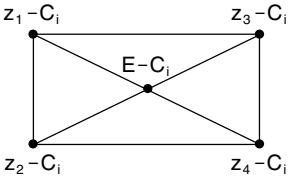
Die Gleichung wird pro Rechteck in allen vier Ecken (auch Scheitel oder Gitterpunkte genannt) ausgewertet, und es wird ein Durchschnittswert (E) berechnet:



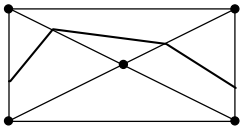
$$E = \frac{z_1 + z_2 + z_3 + z_4}{4}$$

Der Wert E wird als der Wert der Gleichung im Zentrum des Rechtecks behandelt.

Für alle angegebenen Konturenwerte (C_i) gilt:



- In jedem der fünf nebenstehenden Punkten wird der Unterschied zwischen dem z-Wert des Punkts und dem Konturenwert berechnet.
- Bei Vertauschen der Vorzeichen zweier nebeneinander liegender Punkte liegt eine Kontur vor, welche die Verbindungslinie zwischen den beiden Punkten kreuzt. Anhand linearer Interpolation wird näherungsweise ermittelt, wo die Nullstelle die Linie kreuzt.
- Innerhalb des Rechtecks werden die die Linie kreuzenden Nullstellen durch gerade Linien miteinander verbunden.



- Dies wird für jeden Konturenwert wiederholt.

Alle Rechtecke im Gitter werden gleich behandelt.

Die Runge-Kutta-Methode

Für Runge-Kutta-Integrationen gewöhnlicher Differentialgleichungen verwendet der TI-89 Titanium / Voyage™ 200 die Bogacki-Shampine 3(2)-Formel, wie sie im Magazin *Applied Math Letters*, 2 (1989), Seiten 1–9 veröffentlicht wurde.

Bogacki-Shampine 3(2)-Formel

Die Bogacki-Shampine 3(2)-Formel liefert ein Ergebnis mit Genauigkeit dritter Ordnung und eine auf einer eingebetteten Formel zweiter Ordnung basierenden Fehlerschätzung. Für eine Aufgabe der Form:

$$y' = f(x, y)$$

und eine gegebene Schrittweite h kann die Bogacki-Shampine-Formel folgendermaßen geschrieben werden:

$$F_1 = f(x_n, y_n)$$

$$F_2 = f\left(x_n + h\frac{1}{2}, y_n + h\frac{1}{2}F_1\right)$$

$$F_3 = f\left(x_n + h\frac{3}{4}, y_n + h\frac{3}{4}F_2\right)$$

$$y_{n+1} = y_n + h\left(\frac{2}{9}F_1 + \frac{1}{3}F_2 + \frac{4}{9}F_3\right)$$

$$x_{n+1} = x_n + h$$

$$F_4 = f(x_{n+1}, y_{n+1})$$

$$errest = h\left(\frac{5}{72}F_1 - \frac{1}{12}F_2 - \frac{1}{9}F_3 + \frac{1}{8}F_4\right)$$

Die Fehlerschätzung *errest* wird zur automatischen Schrittweitenkontrolle verwendet. Eine gründliche Diskussion dieses Vorgangs finden Sie in *Numerical Solution of Ordinary Differential Equations* von L. F. Shampine (New York: Chapman & Hall, 1994).

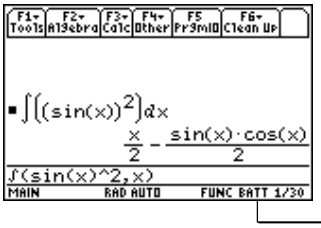
Die TI-89 Titanium / Voyage 200-Software nimmt die Schrittweitenanpassung nicht so vor, dass bestimmte Ausgabepunkte erreicht werden. Es werden hingegen die größtmöglichen Schritte vorgenommen (abhängig von der Fehlertoleranz diftol) und Ergebnisse für $x_n \le x \le x_{n+1}$ geliefert. Hierzu wird das mit der Steigung F_1 durch den Punkt (x_n, y_n) und mit der Steiung F_4 durch (x_{n+1}, y_{n+1}) gehende kubische Interpolationspolynom verwendet. Die Interpolierende ist effektiv und liefert über die gesamte Schrittweite Resultate, die ebenso genau sind, wie die Resultate an den Schrittenden.

Hinweise zu den Batterien

Der TI-89 arbeitet mit zwei Batterietypen: vier Alkali-Batterien für den eigentlichen Betrieb sowie einer Lithiumbatterie, die beim Wechseln der Alkali-Batterien den Speicherinhalt erhält.

Zeitpunkt des Batteriewechsels

Wenn die Alkali-Batterien leer werden, wird das Display dunkler (insbesondere während des Rechnens). Dies werden Sie durch Höherstellen des Display-Kontrasts ausgleichen. Wenn Sie feststellen, daß Sie den Kontrast häufig höherstellen müssen, ist es Zeit für den Austausch der Alkali-Batterien. Wenn die Batterien soweit erschöpft sind, daß der Austausch in Kürze notwendig wird, erscheint in der Statuszeile des Geräts zusätzlich eine Warnanzeige (BATT). Wenn diese Warnanzeige von Normaldarstellung zu inverser Darstellung wechselt (BATT), müssen Sie die Alkali-Batterien unverzüglich austauschen. Die Lithium-Reservebatterie ist etwa alle drei Jahre auszutauschen.



BATT Anzeige

Hinweis: *Damit der Speicherinhalt des TI-89 Titanium / Voyage™ 200 nicht gelöscht wird, muß das Gerät ausgeschaltet sein. Außerdem dürfen Sie die Alkali-Batterien und die Lithiumbatterie nicht gleichzeitig herausnehmen.*

Zur Vermeidung von Datenverlust sollten Sie die Lithiumbatterie nur dann entfernen, wenn vier neue Alkali-Batterien eingelegt sind. Ersetzen Sie die Lithiumbatterie ungefähr alle drei bis vier Jahre.

Folgen des Batterieaustauschs

Der Batterieaustausch ist problemlos und geschieht ohne Datenverlust, wenn Sie nicht gleichzeitig alle Batterien herausnehmen, und wenn Sie es vermeiden, daß alle Batterien zur gleichen Zeit erschöpft sind.

Beim Austausch der Batterien zu beachten

Achten Sie beim Austausch der Batterien auf folgendes:

- Batterien dürfen nicht in die Hände von Kindern gelangen.
- Kombinieren Sie keine neuen und bereits im Gebrauch befindlichen Batterien sowie keine unterschiedlichen Marken oder Typen.
- Kombinieren Sie keine Batterien mit Akkus.
- Legen Sie Batterien stets entsprechend der Anweisung zur Anordnung der Pole (+ und -) ein.
- Versuchen Sie nicht, Batterien in einem Ladegerät zu laden.
- Entsorgen Sie leere Batterien gemäß dem Recycling-Angebot des Geschäfts, in dem Sie die Battterien erworben haben.
- Verbrennen oder öffnen Sie Batterien niemals!

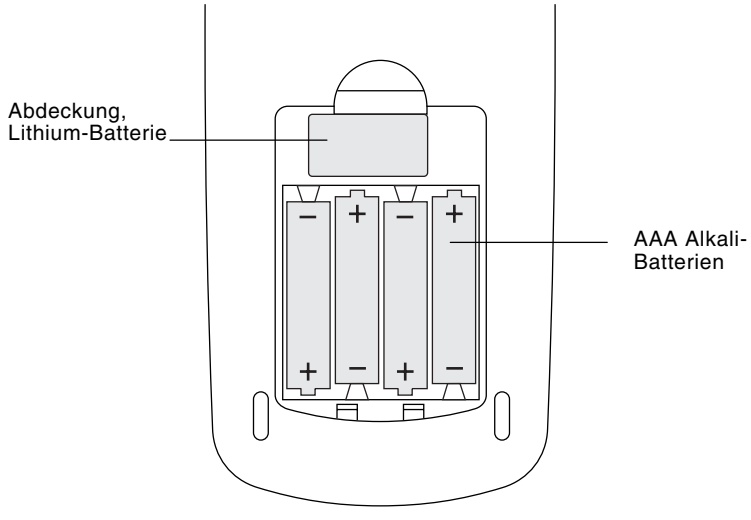
Austauschen der Alkali-Batterie im TI‑89 Titanium

1. Wenn der TI-89 Titanium eingeschaltet ist, schalten Sie ihn aus (drücken Sie 2nd [OFF], um den Verlust von gespeicherten Informationen zu vermeiden.
2. Schieben Sie die Schutzabdeckung über die Tastatur und legen Sie das Gerät mit der Vorderseite nach unten.
3. Halten Sie den Taschenrechner senkrecht und drücken Sie die Verriegelung des Batteriedeckels nach unten; dann nach oben ziehen, um den Deckel zu entfernen.
4. Entnehmen Sie die vier alten AAA-Batterien.
5. Legen Sie vier neue AAA-Alkali-batterien ein und beachten Sie beim Einlegen die im Batteriegehäuse angegebene Polarität (+ und -).
6. Bringen Sie die Batterieabdeckung wieder an, indem Sie die beiden Zähne in die Schlitze im Boden des Batteriegehäuses einsetzen und den Deckel dann bis zum Einrasten drücken.

Austauschen der Lithium-Batterie im TI-89 Titanium

Zum Ersetzen der Lithiumbatterie nehmen Sie den Batteriedeckel ab und lösen die feine Schraube, durch welche die BACK UP BATTERY (LITHIUMBATTERIE) befestigt ist.

Nehmen Sie die leere Batterie heraus, und legen Sie eine neue Batterie vom Typ SR44SW oder 303 mit dem positiven Pol (+) nach oben ein. Setzen Sie die Abdeckung wieder ein, und drehen Sie die Schraube fest.



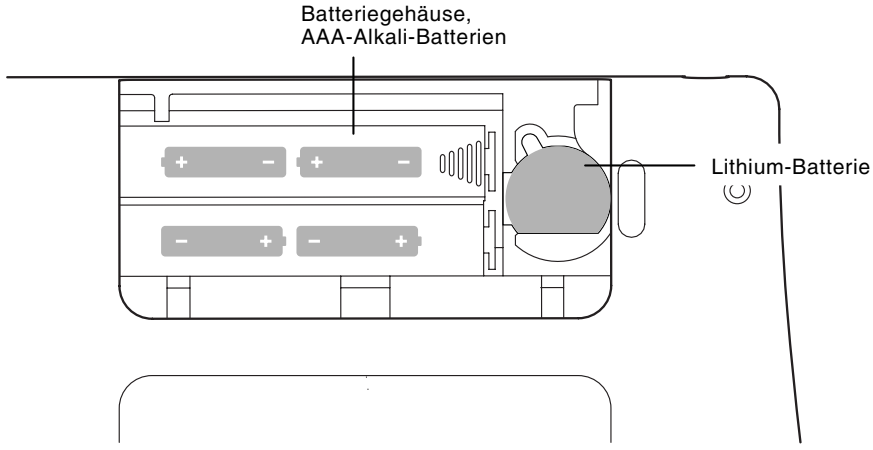
Austauschen der Alkali-Batterie im Voyage 200

1. Wenn der Voyage™ 200 eingeschaltet ist, schalten Sie ihn aus (drücken Sie [2nd] [OFF], um den Verlust von gespeicherten Informationen zu vermeiden.
2. Schieben Sie die Schutzabdeckung über die Tastatur und legen Sie das Gerät mit der Vorderseite nach unten.
3. Drücken Sie auf die mit Zähnen versehene Batterieabdeckung, um sie vom Gerät abzuschieben.
4. Entnehmen Sie die vier alten AAA-Batterien.
5. Legen Sie vier neue AAA-Alkali-Batterien ein und beachten Sie beim Einlegen die im Batteriegehäuse angegebene Polarität (+ und -).
6. Schieben Sie die Batterieabdeckung wieder auf, und zwar mit den Zähnen nach vorne. Den Deckel dann bis zum Einrasten schieben.

Austauschen der Lithium-Batterie im Voyage 200

Zum Austauschen der Lithium-Reservebatterie entfernen Sie zunächst die Batterieabdeckung. Schieben Sie ein stumpfes Instrument (z. B. eine Bleistiftspitze) in die kreisförmige Aussparung neben der Batterie. Drücken Sie mit dem Finger leicht auf die Lithium-Batterie und nehmen Sie die Batterie heraus.

Schieben Sie eine neue CR1616 oder CR1620 Batterie ein, und zwar mit der Plusseite (+) nach oben. Drücken Sie fest auf die neue Lithium-Batterie, bis Sie einrückt.



Bei Betriebsproblemen

Wenn Probleme beim Betrieb des TI-89 Titanium / Voyage™ 2004 auftreten, schlagen Sie zunächst hier nach, um die Ursachen zu ermitteln und Abhilfe zu schaffen.

Probleme und Lösungen

Ihr Problem:	Lösungsvorschlag:
Sie sehen auf dem Anzeige nichts.	Drücken Sie ◆ + , um die Anzeige dunkler bzw. ◆ − , um sie heller zu stellen.
Die Anzeige BATT ist sichtbar.	Wechseln Sie die Batterien. Wenn BATT invers (unterlegt) angezeigt wird (BATT), tauschen Sie die Batterien bitte unverzüglich aus.
Die Anzeige BUSY ist sichtbar.	Es wird soeben ein Rechenvorgang ausgeführt. Möchten Sie den Vorgang abbrechen, drücken Sie ON.
Die Anzeige PAUSE ist sichtbar.	Die Ausführung eines Vorgangs ist vorübergehend gestoppt (Graph oder Programm), und der TI-89 Titanium / Voyage 200 wartet auf eine Eingabe; drücken Sie ENTER.
Es wird eine Fehlermeldung angezeigt.	Schlagen Sie die Liste der Fehlermeldungen in diesem Modul nach. Drücken Sie zum Löschen der Meldung ESC.
Der TI-89 Titanium / Voyage 200 scheint nicht ordnungsgemäß zu arbeiten.	Drücken Sie mehrmals ESC, um ein Menü/ ein Dialogfeld zu verlassen und den Cursor wieder in die Eingabezeile zu bringen. — oder — Überprüfen Sie, ob die Batterien korrekt eingesetzt sind. Die Batterien könnten auch erschöpft sein.
Der TI-89 Titanium "hängt"; er reagiert nicht auf Tastatureingaben.	Der folgende Vorgang löscht den RAM. Dabei werden alle Daten, Programme, benutzerdefinierten Variablen, Funktionen oder Ordner gelöscht. Drücken und halten Sie ◀, ▶ und 2nd. Drücken Sie dann ON und lassen Sie wieder los. Der folgende Vorgang löscht den RAM *und* den Flash-ROM. Dabei werden alle Daten, Programme, benutzerdefinierten Variablen, Funktionen, Ordner Flash-Anwendungen und Benutzerdaten gelöscht. 1. Entfernen Sie eine der 4 AAA-Batterien. 2. Halten Sie die Tasten (-) und) gedrückt während Sie die Batterie wieder einsetzen.

Hinweis: *Wenn das Gerät "hängt" und Sie dies wie hier beschrieben beheben, wird der TI-89 Titanium / Voyage 200 zurückgesetzt und der Speicherinhalt gelöscht.*

Ihr Problem:	Lösungsvorschlag:
	3. Halten Sie die Tasten (-) und) danach noch etwa fünf Sekunden gedrückt.
Der Voyage 200 “hängt”; er reagiert nicht auf Tastatureingaben.	Der folgende Vorgang löscht den RAM. Dabei werden alle Daten, Programme, benutzerdefinierten Variablen, Funktionen oder Ordner gelöscht. Drücken Sie 2nd und ◁, und halten Sie diese Tasten gedrückt. Drücken Sie dann ON, und lassen Sie ON wieder los. Der folgende Vorgang löscht den RAM *und* den Flash-ROM. Dabei werden alle Daten, Programme, benutzerdefinierten Variablen, Funktionen, Ordner Flash-Anwendungen und Benutzerdaten gelöscht. 1. Entfernen Sie eine der 4 AAA-Batterien. 2. Halten Sie die Tasten (-) und) gedrückt während Sie die Batterie wieder einsetzen. 3. Halten Sie die Tasten (-) und) danach noch etwa fünf Sekunden gedrückt.

Hinweise zu TI Produktservice und Garantieleistungen

Informationen über Produkte und Dienstleistungen von TI

Wenn Sie mehr über das Produkt- und Serviceangebot von TI wissen möchten, senden Sie uns eine E-Mail oder besuchen Sie uns im World Wide Web.

E-Mail-Adresse: ti-cares@ti.com

Internet-Adresse: education.ti.com

Service- und Garantiehinweise

Informationen über die Garantiebedingungen oder über unseren Produktservice finden Sie in der Garantieerklärung, die dem Produkt beiliegt. Sie können diese Unterlagen auch bei Ihrem Texas Instruments Händler oder Distributor anfordern.

Anhang C: Programmierhandbuch

Die in den Funktionen defModus( ), holModus( ), defGraph( ) und defTabl( ) verwendeten Parameter-/Funktionsstrings werden bei Verwendung in einem Programm nicht in andere Sprachen übersetzt. Beispiel: Wird ein Programm in Französisch als Arbeitssprache geschrieben und danach zu Italienisch gewechselt, gibt das Programm einen Fehler aus. Um diesen Fehler zu vermeiden, müssen die Buchstaben durch Ziffern ersetzt werden. Diese Ziffern funktionieren in sämtlichen Sprachen. In diesem Anhang sind die Ziffern zum Ersetzen der Strings aufgelistet.

In den folgenden Beispielen wird das Austauschen von Buchstaben gegen Ziffern in der Funktion defModus( ) dargestellt.

Beispiel 1: Programm verwendet Parameter-/Modusstrings aus Buchstaben:

```
defModus("Graph","Folge")
```

Beispiel 2: Dasselbe Programm mit Ziffern als Buchstabenersatz:

```
defModus("1","4")
```

Parameter-/Moduseinstellung	Strings
ALLE	0
Graph	1
FUNKTION	1
PARAMETRISCH	2
POLAR	3
FOLGE	4
3D	5
DIFFGLEICHUNG	6
angez. Ziffern	2
FIX 0	1
FIX 1	2
FIX 2	3
FIX 3	4
FIX 4	5
FIX 5	6
FIX 6	7
FIX 7	8
FIX 8	9
FIX 9	10
FIX 10	11
FIX 11	12
FIX 12	13
FLIESS	14
FLIESS 1	15
FLIESS 2	16
FLIESS 3	17
FLIESS 4	18
FLIESS 5	19
FLIESS 6	20
FLIESS 7	21
FLIESS 8	22
FLIESS 9	23

Parameter-/Moduseinstellung	Strings
FLIESS 10	24
FLIESS 11	25
FLIESS 12	26
Winkel	3
BOGENMASS	1
GRAD	2
Exponentialformat	4
NORMAL	1
WISSENSCH	2
TECHNISCH	3
Komplexes Format	5
REEL	1
KARTESISCH	2
POLAR	3
Vektorformat	6
KARTESISCH	1
ZYLINDRISCH	2
SPHÄRISCH	3
Math AnzFmt	7
AUS	1
ON	2
Bildsch teilen	8
VOLL	1
OBEN-UNTEN	2
LINKS-RECHTS	3
Applik im 1.BS	9
(Anwendungen sind nicht mit Nummern versehen)	
Applik im 2.BS	10
(Anwendungen sind nicht mit Nummern versehen)	
Anz. der GraphBS	11
1	1
2	2

Parameter-/Moduseinstellung	Strings
Graph 2	12
FUNKTION	1
PARAMETRISCH	2
POLAR	3
FOLGE	4
3D	5
DIFFGLEICHUNG	6
BildschTeilVerh	13
1:1	1
1:2	2
2:1	3
Exakt/Näherung	14
AUTO	1
EXAKT	2
APPROXIMIERT	3
Basis	15
DEZ	1
HEX	2
BIN	3

Parameter-/Moduseinstellung	Strings
Koordinaten	1
RECHTW	1
POLAR	2
AUS	3
GraphFolge	2
FOLG	1
GLEICHZTG	2
Raster	3
AUS	1
ON	2
Achsen	4
Im 3D-Modus:	
AUS	1
ACHSEN	2
BOX	3
Nicht im 3D-Modus:	
AUS	1
ON	2
Cursor	5
AUS	1
ON	2
Bezeichnungen	6
AUS	1
ON	1
Seq Achsen	7
ZEIT	1
NETZ	2
EIGENE	3
Lösungsmethode	8
RK	1
EULER	2

Parameter-/Moduseinstellung	Strings
Felder	9
STEIGFLD	1
RICHTFLD	2
FELDAUS	3
DE Achsen	10
ZEIT	1
Y1-VS-Y2	2
T-VS-Y'	3
Y-VS-Y'	4
Y1-VS-Y2'	5
Y1'-VS-Y2'	6
XR ZeiStil	11
DRAHTMODELL	1
VERBORG OBERFL	2
KONTURSTUFEN	3
DRAHT + KONTUR	4
PLOT IMPLIZIT	5

Parameter-/Moduseinstellung	Strings
Graph <-> Tabl	1
AUS	1
ON	2
Unabhängig	2
AUTO	1
FRAGE	2
Achsen	4

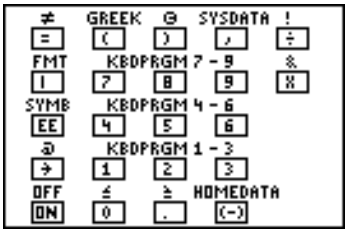
TI-89 Titanium Schnellzugriffstasten

Allgemeines

◆ APPS	Liste der Flash-Anwendungen
2nd [↔]	Wechsel zwischen den beiden zuletzt gewählten Anwendungen oder geteilten Bildschirm
◆ −, ◆ +	Kontrast dunkler/heller
◆ ENTER	Näherungsweise Berechnung eines Resultats
◆ ▲, ◆ ▼	Cursor (in Editor-Anwendungen) ganz nach oben oder ganz nach unten
⇧ ▲, ⇧ ▼	Große Objekte im History-Bereich durchlaufen
⇧ ◀, ⇧ ▶	Zeichen links oder rechts neben Cursor markieren
2nd ▲, 2nd ▼	Bild auf oder Bild ab (in Editor-Anwendungen)
2nd ◀, 2nd ▶	Cursor ganz nach links oder ganz nach rechts

Angezeige der Tastaturbelegung (◆ EE)

Um diese wieder zu verlassen, drücken Sie ESC.



In der Übersicht sehen Sie Schnellzugriffstasten, die auf der Tastatur selbst nicht angezeigt sind. Drücken Sie zunächst ◆ und dann wie folgt die entsprechende Taste:

◆ =	≠
◆ (	Zugriff auf griechische Buchstaben (siehe nächste Spalte)
◆)	◕ (Kommentar)
◆ ,	Grafikkoordinaten in sysdata kopieren
◆ ÷	! (Fakultät)
◆ |	Dialogfeld FORMATS aufrufen
◆ 1 – ◆ 9	Programme kbdprgm1() bis kbdprgm9() ausführen
◆ ×	& (anfügen)
◆ EE	Anzeige der Tastaturbelegungs-übersicht
◆ STO▸	@
◆ ON	Gerät so ausschalten, daß es beim nächsten Einschalten zur aktuellen Anwendung zurückkehrt
◆ 0 (null)	≤
◆ .	≥
◆ (-)	Grafikkoordinaten in den History-Bereich des Ausgangsbildschirms kopieren

Buchstabeneingabe

alpha	Eingabe eines Kleinbuchstabens
⇧	Eingabe eines Großbuchstabens
2nd [a-lock]	Kleinbuchstaben-Feststellfunktion
⇧ alpha	Großbuchstaben-Feststellfunktion
alpha	Feststellfunktion beenden

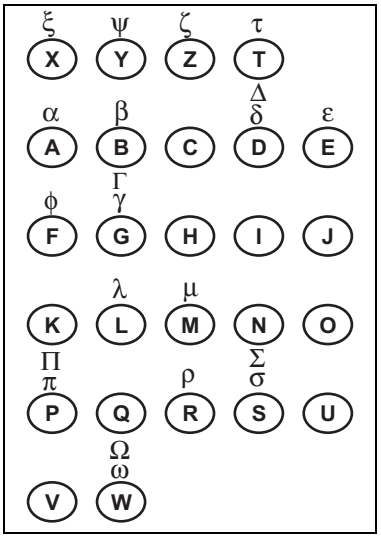
3D-Darstellung

▲, ▼, ◀, ▶	Graph bewegt anzeigen
+, −	Animationsgeschwindigkeit ändern
X, Y, Z	Blick entlang der Achse
0	Zur ursprünglichen Ansicht zurückkehren
|	Grafikformat-Stil ändern
×	Erweiterte/normale Ansicht

Griechische Buchstaben

◆ (	Zugriff auf den griechischen Zeichensatz.
◆ (alpha + Buchstaben	Zugriff auf griechische Kleinbuchstaben. Beispiel: ◆ (alpha [W] zeigt ω an.
◆ (⇧ + Buchstaben	Zugriff auf griechische Großbuchstaben. Beispiel: ◆ (⇧ [W] zeigt Ω an.

Wenn Sie eine Tastenkombination drücken, die nicht auf einen griechischen Buchstaben zugreift, wird der dieser Taste zugeordnete normale Buchstabe eingegeben.



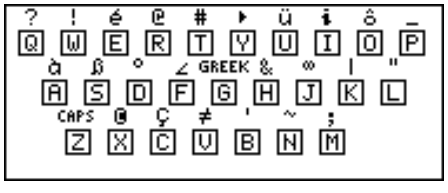
Voyage™ 200 Schnellzugriffstasten

Allgemeines

♦ APPS	Liste der Flash-Anwendungen
2nd [↔]	Wechsel zwischen den beiden zuletzt gewählten Anwendungen oder geteilten Bildschirm
♦ D	Grafikkoordinaten in sysdata kopieren
♦ F	Dialogfeld FORMATS aufrufen
♦ H	Grafikkoordinaten in den Hisory-Bereich des Ausgangsbildschirms kopieren
♦ N	Eine neue Variable erstellen
♦ O	Eine vorhandene Variable öffnen
♦ S	Kopie speichern als
♦ −, ♦ +	Kontrast dunkler/heller
♦ ENTER	Näherungsweise Berechnung eines Resultats
♦ ON	Gerät so ausschalten, daß es beim nächsten Einschalten zur aktuellen Anwendung zurückkehrt
♦ 1 – ♦ 9	Programme kbdprgm1() bis kbdprgm9() ausführen

Angezeige der Tastaturbelegung (♦ KEY)

Um diese wieder zu verlassen, drücken Sie ESC.



Die folgende Tabelle enthält nicht auf der Tastatur des Voyage 200 markierten Tastenkombinationen. Die nächste Spalte enthält Akzentzeichen und griechiesche Buchstaben.

2nd Q	?
2nd W	! (Fakultät)
2nd R	@
2nd T	# (indirekt)
2nd H	& (anfügen)
2nd X	◕ (Kommentar)
♦ =	≠
♦ 0 (null)	≤
♦ .	≥

Bearbeiten

♦ ⏶	Cursor nach ganz oben
♦ ⏷	Cursor nach ganz unten
2nd ◀	Cursor nach ganz links
2nd ▶	Cursor nach ganz rechts
⎋ ⏶, ⎋ ⏷	Große Objekte im History-Bereich durchlaufen
⎋ ⏶, ⎋ ⏷	Bild auf und Bild ab
♦ X	Ausschneiden
♦ C	Kopieren
♦ V	Einfügen

3D-Darstellung

⏶, ⏷, ◀, ▶	Graph bewegt anzeigen
+, −	Animationsgeschwindigkeit ändern
X, Y, Z	Blick entlang der Achse
0 (null)	Zur ursprünglichen Ansicht zurückkehren
F	Grafikformat-Stil ändern
×	Erweiterte/normale Ansicht

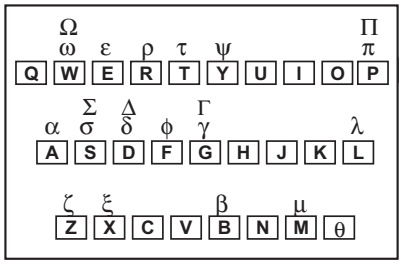
Akzente

2nd A + Buchstabe	à, è, ì, ò, ù, À, È, Ì, Ò, Ù
2nd C + Buchstabe	ç, Ç
2nd E + Buchstabe	á, é, í, ó, ú, ý, Á, É, Í, Ó, Ú, Ý
2nd N + Buchstabe	ã, ñ, õ, Ã, Ñ, Õ
2nd O + Buchstabe	â, ê, î, ô, û, Â, Ê, Î, Ô, Û
2nd U + Buchstabe	ä, ë, ï, ö, ü, ÿ, Ä, Ë, Ï, Ö, Ü

Griechische Buchstaben

2nd G	Zugriff auf den griechischen Zeichensatz.
2nd G + Buchstabe	Zugriff auf griechische Kleinbuchstaben. Beispiel: 2nd G W zeigt ω an.
2nd G ⬆ + Buchstabe	Zugriff auf griechische Großbuchstaben. Beispiel: 2nd G ⬆ W zeigt Ω an.

Wenn Sie eine Tastenkombination drücken, die nicht auf einen griechischen Buchstaben zugreift, wird der dieser Taste zugeordnete normale Buchstabe eingegeben.



Unterschiedliche Tasten und Tastenkombinationen

Bei einer Reihe von Operationen sind auf TI-89 Titanium / Voyage™ 200 unterschiedliche Tasten erforderlich. Die folgende Tabelle enthält die Tasten und Tastenkombinationen für die wichtigsten Befehle auf diesen beiden Taschenrechnern.

FUNKTION	TI-89 Titanium	Voyage™ 200
Buchstaben		
Ein Kleinbuchstabe (a-s, u, v, w)	alpha A-S, U-W	A-S, U-W
Ein Kleinbuchstabe (t, x, y, z)	T, X, Y, Z	T, X, Y, Z
Mehrere Kleinbuchstaben	2nd [a-lock]	
Ende der Kleinbuchstabeneingabe	alpha	
Mehrere Großbuchstaben	⇧ [a-lock]	2nd [CAPS]
Ende der Großbuchstabeneingabe	alpha	2nd [CAPS]
FUNKTIONSTASTEN		
F6	2nd [F6]	F6
F7	2nd [F7]	F7
F8	2nd [F8]	F8
CURSORSTEUERUNG		
Umfangreiche Objekte in History-Bereich durchblättern	⇧ ↑, ⇧ ↓	↑, ↓
Cursor in Eingabezeile zum Zeilenanfang/ende steuern	2nd ←, 2nd →	2nd ←, 2nd →
	↑ und ← ↑ und → ↓ und ← ↓ und →	↑ und ← ↑ und → ↓ und ← ↓ und →
FUNKTIONEN		
Hauptbildschirm anzeigen	HOME	◆ [CALC HOME]
Ausschneiden	◆ [CUT]	◆ X
Kopieren	◆ [COPY]	◆ C
Einfügen	◆ [PASTE]	◆ V
Katalog	CATALOG	2nd [CATALOG]
Dialogfeld für Maßeinheiten anzeigen	2nd [UNITS]	◆ [UNITS]
Sinus	2nd [SIN]	SIN
Cosinus	2nd [COS]	COS
Tangens	2nd [TAN]	TAN
Ln	2nd [LN]	LN
e^x	◆ [e^x]	2nd [e^x]
EE (Exponent der 10er Potenz)	EE	2nd [EE]

FUNKTION	TI-89 Titanium	Voyage™ 200
SYMBOLE		
▶ (Konvertierung)	2nd [▶]	2nd [▶]
_ (Unterstrich)	◆ [_]	2nd [_]
θ (Theta)	◆ [θ]	θ
| ("with")	|	2nd [|]
' (Ableitungsstrich - nur für DGL)	2nd [']	2nd [']
° (Grad)	2nd [°]	2nd [°]
∠ (Winkel)	2nd [∠]	2nd [∠]
Σ (Summensymbol)	CATALOG Σ (	2nd [Σ]
x^{-1} (Kehrwert)	CATALOG ^-1	2nd [x^{-1}]
Leerzeichen	alpha [␣]	Leerzeichen
UNSICHTBARE TASTENKOMBINATIONEN		
Daten in Variable "Sysdata" plazieren	◆ ,	◆ D
Griechische Zeichen	◆ (alpha oder ◆ (⬆	◆ G oder ◆ G ⬆
Tastaturbelegung	◆ EE	◆ [KEY]
Daten in History des Ausgangsbildschirms plazieren	◆ (-)	◆ H
Gravis (à, è, ì, ò, ù)	2nd [CHAR] 5	2nd A a, e, i, o, u
Cedille (ç)	2nd [CHAR] 5 6	2nd C c
Akut (á, é, í, ó, ú, ý)	2nd [CHAR] 5	2nd E a, e, i, o, u, y
Tilde (ã, ñ, õ)	2nd [CHAR] 5 6	2nd N a, n, o
Zirkumflex (â, ê, î, ô, û)	2nd [CHAR] 5	2nd O a, e, i, o, u
Umlaut (ä, ë, ï, ö, ü, ÿ)	2nd [CHAR] 5	2nd U a, e, i, o, u, y
? (Fragezeichen)	2nd [CHAR] 3	2nd Q
β (Beta)	2nd [CHAR] 5 6	2nd S
# (Konvertierung)	2nd [CHAR] 3	2nd T
& (Verkettung)	◆ × (Mal-Zeichen)	2nd H
@ (Klammeraffe)	◆ STO▶	2nd R
≠ (ungleich)	◆ =	2nd V
! (Fakultät)	◆ ÷	2nd W
Kommentar	◆) ◉	2nd X ◉
Neu	F1 3	◆ N
Öffnen	F1 1	◆ O
Kopie speichern als	F1 2	◆ S
Formatdialog	◆ |	◆ F

Inhalt

Symbole

!, Fakultät 73, 988
→, speichern 655, 995
≠, /=, ungleich 664, 987
#, Konvertierung 661
≠, Konvertierung 990
√( ), Quadratwurzel 989
%, Prozent 986
∞, unendlich 298
&, anhängen 661, 988
', Minute Angabe 992
', Prim 992
', zweite Ableitung 992
E Exponent 894
Σ( ), Summe 286
∫( ), Integral 77, 259, 261, 262, 268, 286, 287, ... 989
*, multiplizieren 984
∫f(x)dx (math. Tool für Graphen) 356, 359
+, addieren 982
F1–F8 (Funktionstasten)
  Ort 11
  Verwendung 16
  Wählen von Kategorien 32, 36
  Wählen von Menüs 53
  Wechseln zwischen Menüleistenmenüs 59
°, Gradangabe 991, 992
°, Temperaturangaben 849
∠, Winkel 991
.*, Punktmultiplikation 985
.+, Punktaddition 985
./, Punktdivision 986
.^, Punkt Potenz 986
.−, Punktsubtraktion 985
/, dividieren 984
<, kleiner als 664, 987
<<...>>, nicht genügend Display-Speicher 248
=, gleich 664, 987
>, größer als 664, 988
@, Platzhalter für Ganzzahl 297
Δlist( ), Differenzen auflisten 917
Δtbl, Tabellenintervall 498
ΔtmpCnv( ), Temperaturbereichskonvertierung . . 307
Δx (Fenstervariable) 349
Δy (Fenstervariable) 349
^, hoch 990
_, Unterstrich 992
≤, <=, kleiner gleich 664, 988
− (Subtraktionstaste) 17
≥, >=, größer gleich 664, 988
|, gilt für alle 258
©, Kommentar 638, 995

|, with 77, 78, 269, 270, 994
← / ♦ [DEL] (Lösche Zeichen) 21
[Hand] (Hand-Modifikatortaste)
Beschreibung 16
Ort 11
Status 43
2nd (2nd-Modifikatortaste)
Beschreibung 15
Ort 11
Status 43
2nd [MEM] (MEMORY (Speicher)) 21
2nd [EE] (Exponententaste) 18
2nd [▶] (Maßeinheitenumwandlung) . . 20
2nd [RCL] (Rückholen) 21
2nd [CATALOG] (Catalog)
Befehle 26
Tastenbefehl 21
verlassen 28
2nd [CUSTOM] (Custom)
Beispiel 61
Beschreibung 60
Tastenbefehl 20
2nd [CHAR] (Zeichen)
Auswählen von Zeichen 12
Eingeben von Sonderzeichen . . . 11
Tastenbefehl 21
2nd [QUIT]
Ausschalten des Rechners 6
Eingeben von Befehlen 26
Rechner Hauptbildschirm 28
▸, konvertieren 303, 993
▸Bin, Binäranzeige 869
▸Bin, in Binärform anzeigen 754
▸Cylind, als zylinkdrischen Vektor anzeigen 882
▸DD, als Dezimalwinkel anzeigen . . . 885
▸Dec, als Dezimalzahl anzeigen 754, 886
▸DMS, als Grad/Minute/Sekunde anzeigen 892
▸Hex, als Hexadezimalzahl anzeigen 754, ... 910
▸Polar, als Polarvektor anzeigen 935
▸Rect, als Rechteckvektor anzeigen . 944
▸Sphere, als Kugelvektor anzeigen . . 963
↑ (Umschalt-Modifikatortaste)
Beschreibung 15
Ort 11
Status 43
♦ (Karo-Modifikatortaste)
Beschreibung 15
Ort 11
Status 43
♦ F (FORMATS/GRAPH FORMATS (FORMAT/GRAPHIK FORMATE)) 20
♦ N (Neue Datei) 20
♦ O (Datei öffnen) 20
♦ S (SAVE COPY AS (KOPIE SPEICHERN ALS))
Beispiel 58
Beschreibung 20
Dialogfeld 20
♦ X (Ausschneiden) 19

STO▸ (Speichern) Taste 21
◀▶▲▼ (Cursortasten)
Eingeben von Befehlen 27
Funktion 17
Öffnen von Apps 33
Ort 11
Verwenden der Hand-Taste 16
Verwenden des Menüs CHAR . . . 12
Verwenden mit der Hand-Taste . . 16
wählen von Eingabe/Antwort-Paaren 30
Zusatzfunktionen 17
E Exponent 894
Σ( ), Summe 990
-, negativ 165
-, negatives Vorzeichen 986
−, subtrahieren 983
Π( ), Produkt 286, 990
θmax (Fenstervariable) 370
θmin (Fenstervariable) 370
θstep (Fenstervariable) 370
r, Radians 991
STO▸ (Vorzeichentaste) 17
T, transponieren 967

Ziffern

0b, Hinweis auf Binärzahl 995
0h, Hinweis auf Hexadezimalzahl . . . 995
10^( ), hoch 10 993
2nd-Modifikatortaste (2nd)
Beschreibung 15
Status 43
3D Modus 43
3D-Graphen 405
Animation 97, 423
CONTOUR LEVELS 429
HIDDEN SURFACE 429
WIRE AND CONTOUR 429
WIRE FRAME 429

A

Abdeckung
austauschen 5
entfernen 5
verstauen 5
Ableitungen 77
erste Ableitung, d( ) 77, 268, 285
erste, d( ) 885
numerisch, nDeriv( ) 925
numerische, nDeriv( ) 286
ABOUT (INFO)-Bildschirm 64
Abrufen/Antworten
Rechner, GetCalc 809, 810
abs( ), absoluter Wert 866
Abschalten
nach APD 7
nach Inaktivität 7
absoluter Wert, abs( ) 852, 866
Achsen (Folge), CUSTOM 392
addieren, + ... 982
aktives Fenster, switch( ) 966

Aktivitäten. *Siehe* Beispiele, Einführungen, Aktivitäten
Aktualisieren des Betriebssystems (BS) . 814, ... 815
aktuelle Uhrzeit abrufen, getTime( ) . 907
Aktueller Verzeichniszustand 42
aktuelles Datum abrufen, getDate( ) . 905
Aktuell-Modus 22
Akzentzeichen 723, 727
Algebra 862
Algebra (Menü) 275, 276, 278
Alle Kategorie 36
An Rechner senden, SendCalc 809, 810
an Taschenrechner senden, SendCalc . 698, ... 950
Analysis, Operationen 862
and, (Boole) und 270, 758, 866
and, Boole and 665
AndPic, und Bild 692, 867
Anfangsbedingungen 458
Anforderung, Request . . . 682, 684, 945
Angle (Modus) 325
Angle mode 1007
angle( ), Winkel 867
anhängen, & 661, 988
ans( ), letzte Ausgabe 237, 868
Anschließen
Computer 66
Geräte 66
TI ViewScreen OHP 67
TI-Presenter Videoadapter 67
Voyage 200 PLT 66
Anschluss, Zubehör 66
Anweisungen 168
Hauptbildschirm 28
Katalog 25
Anzahl, dim( ) 891
anzeigen
E/A-Bildschirm, Disp 133
Graph, DispG 682, 691, 891
Hauptbildschirm, DispHome 682, 891
I/O-Bildschirm, Disp . 640, 682, 700, 891
Tabelle, DispTbl 683, 690, 892
anzeigen als
binär, ▸Bin 754, 869
Dezimalwinkel, ▸DD 885
Dezimalzahl, ▸Dec 754, 886
Grad/Minute/Sekunde, ▸DMS . . . 892
Hexadezimalzahl, ▸Hex 754, 910
Kugelvektor, ▸Sphere 963
Polarvektor, ▸Polar 935
Rechteckvektor, ▸Rect 944
zylinkdrischen Vektor, ▸Cylind . . . 882
Anzeigestellen-Modus 22
APD (Automatic Power Down) 151
Einschalten nach 7
im BS Download-Modus 71
während Berechnung oder Programm 8
APPLICATIONS (Menü) 188, 193
APPLICATIONS Menü (APPS) 63

APPROX (NÅHERUNG) Modus-Status . 43
approx( ), nähern 277, 868
Approximate (Modus) 177, 218, 260
Apps (Handheld-Software-Anwendungen)
löschen 66
öffnen 33, 63
Symbolauswahl, zuletzt geöffnet . . 4
Symbole 2
Verweise 38
Apps-Arbeitsfläche
abschalten 44
Ausschalten des Rechners 6
Datum und Zeit 48
Erster Start 2
Hauptbildschirm 28
Kategorien 32
Modus 22, 44
Split-Screen-Status 41
Teile 4
Uhr 46
Arc (math. Tool für Graphen) . .356, 362, 373
Archive, Variablen archivieren .655, 785, 868
arcLen( ), Bogenlänge 286, 868
Assembler 897
Assembler (Programmiersprache) . .706, 707, ... 708, 709
Assembler-Programm ausführen, Exec . 709, ... 897
Aufforderung, Prompt( ) 682, 937
Aufgaben (neu), NewProb 204, 927
Aufräumen Meldung 787, 788, 789
augment( ), erweitern/verketten 830, 868
Ausdrücke 28, 167, 170
Ausdruck in Liste, exp▸list( ) 897
erweitern 74
kürzen 75
String in Ausdruck, expr( ) 661, 662, 682, ... 899
Ausgabe (letzte), ans( ) 237, 868
Ausgabe, Output 683, 931
Auslassung, ... 506
Ausschalten 6
ausschließendes oder (Boole), xor . . 665
ausschneiden 229, 230, 231, 720
Ausschneiden (◆ X) 19
Auswählen
Buchstaben bei der Bearbeitung . . 15
vollen Namen der App 3
Auto (Modus) 177, 218, 261
AUTO Modusstatus 43
Auto-Einfügen 30
Automatic Power Down (APD)
Einschalten nach 7
während Berechnung oder Programm 8
automatisch einfügen 230
automatisch/manuell, Independent AUTO/ASK 499

automatische Erstellung von Tabellen . . 502
automatische Vereinfachung 263
automatisches Einfügen 238, 239
avgRC( ), durchschn. Änderungsrate 869
Axes (Einstellungen) 419, 428
Axes (Graphenformat) 338, 452, 474, 476

B

Basis 753
Boole'sche Operatoren 758
für ln, *e* 298
Konvertierung 754
mathematische Operatoren 755, 757
Basismodus 22
BATT (Meldung) 219
Batterie
Schwach-Status 43
Verlängern der Lebensdauer 7
Batterien 153, 219
Austauschanzeige 43
austauschen 1, 68, 69
Erster Start 2
Umgang 71
bearbeiten 183
Bedingungen 273
beenden
ben.def. Menüleiste, EndCustm . 882
ben.def., EndCustm 684
Dialogfeld, EndDlog 683, 890
Exit 897
for, EndFor 903
Funktion, EndFunc 904
if, EndIf 911
Menüleiste, EndTBar 684, 972
Programm, EndPrgm 649, 936
Schleife, EndLoop 676, 920
Test, EndTry 701, 973
beenden Programm, EndPrgm 130
Befehle
Flash-Apps 25
Tasten 11, 13
Befehlsmarker einfügen 731
Befehls-Skripts 227, 731, 732, 733, 734, 735
in der Praxis 836
Beispiel
abschalten der Uhr 52
Ändern von Moduseinstellungen . . 23
Arbeiten mit der Tastenbelegung . 14
auswählen von Menüoptionen . . . 55
Bearbeiten von Kategorien 39
erstellen eines neuen Programms 34
Verwenden der Tastaturbelegung . 14
Verwenden des Katalogs 27
Verwenden des Menüs CHAR . . . 12
Verwenden von Dialogfeldern 58
wiederherstellen des Vorgabe-Anwendermenüs 61
Beispiele, Ansicht, Konstanten und Maßeinheiten 80
Beispiele, Ansicht, Rechenarten

Gleichungssysteme 77
lineare Gleichungen lösen 281
Polynomfaktoren 279
Beispiele, Einführungen, Aktivitäten
3D-Graphen 97
Ableitung einer Gleichung zweiter Ordnung 826
Ableitungen 77
Baseball 848
Basis 140
Beispiel-Skript mit Text-Editor . . 835
Bevölkerung 118
Bildschirm teilen 113, 849
CBL-Programm 845
cos(x)=sin(x) 831
Daten-/Matrix-Editor 115
Datenfilterung 841
Differentialgleichung dritter Ordnung 471
Differentialgleichung zweiter Ordnung 466, 487
Differentialgleichungen 103
divergierende Netz-Plots 397
Drei3D-Graphen 833
Erweitern von Ausdrücken 74
Faktor 72
Fibonacci-Folge 403
Flugbahn eines Balls 88
Funktionsteile 108
Graphen von Folgen 94
implizite Plots 442
Integrale 77
Kapitalwert 856
komplexe Nullstellen 850
komplexe Oberfläche 436
komplexe Zahlen 72
Faktoren 858
konvergierende Netz-Plots 395
Kürzen von Ausdrücken 75
Lösen linearer Gleichungen . . 75, 76
numerische Auflösungsfunktion . 137
Operationen mit Text 134
oszillierende Netz-Plots 398
parametrische Darstellung 848
parametrische Graphen 88
polare Rose 92
Polynom dritter Ordnung 850
Polynomfaktoren 75
Primfaktoren 72
programmieren . 130, 133, 702, 703, 704, ... 705
rationale Funktion zerlegen 838
rationale Zahlen
Faktoren 858
Raubtier-Beutetier-Modell . . 400, 476
reelle Zahlen
Faktoren 858
Stab-Ecke-Aufgabe 824
Standardrente 854
Statistik 118
Tabellen 111
Theorem des Pythagoras . . 824, 825

Wahrscheinlichkeitsrechnung . . 859
Wald und Bäume 94
benutzerdefinierte Funktionen .210, 240, 292, . 408, 519, 521, 643, 644, 645, 886
benutzerdefinierte Maßeinheiten . . . 310
benutzerdefinierte Menüleiste siehe Menüleiste
benutzerdefinierte Plots, CUSTOM .385, 474, ... 476
Benutzereinheiten-Modus 22
Berechnung stoppen 176
Betonungszeichen
CHAR-Menü 21
Betrachtung in 3D 424
Betrachtungswinkel 419
Betriebssystem 815, 816, 817
Betriebssystem (BS)
Download 70
Betriebssystem-(BS)-Version 253
Betriebssystem, aktualisieren . .813, 814, 815
Bezeichnung, Lbl 651, 666, 670, 677, 914
Bild ersetzen, RplcPic 692, 948
Bild mit ausschließendem oder, XorPic . 693
Bild mit exklusives oder, XorPic 976
Bilder 543, 546
dazu, AndPic 692, 867
ersetzen, RplcPic 692, 948
exklusives oder, XorPic 976
löschen 547
mit ausschließendem oder, XorPic . . 693
neu, NewPic 656, 692, 926
speichern, StoPic 692, 965
zurückholen, RclPic 692, 944
Zyklus, CyclePic 692, 882
Bildlauf 246
Bildschirm teilen 526, 530, 554–564, 734, ... 751
aktives Fenster, switch( ) 966
beenden 558
Eingabezeile 561, 563
Einstellung 554, 555
Fenster, switch( ) 679
Pixelkoordinaten 556
teilen 561
Bildzyklus, CyclePic .547, 548, 692, 882
binär
Anzeige, ▶Bin 869
Hinweis, 0b 995
rotate, rotate( ) 760
shift, shift( ) 761
Binärdarstellung, ▶Bin 754
Blättern 30
BldData, Daten erzeugen .480, 655, 870
Bogenlänge, arcLen( ) 286, 868
Boole
ausschließendes oder, xor 665
exklusives oder, xor 759, 976
nicht, not 666, 758, 928

oder, or 665, 758, 930
und, and 270, 665, 758, 866
Box Plot 612
Brüche 277, 284, 838, 937
BS 813, 814, 815
BS-(Betriebssystem)-Version 253
Buchstaben
Großflschreibung 11
Großschreibung 15
löschen 21
Build Web, Netz erzeugen 392, 394
BUSY 44
BUSY (Zeiger) 220, 340, 630
Busy/Pause-Status 43

C

Calc (Menü) 285
Calculator-Based Laboratory system
Anschließen 67
Calculator-Based Laboratory. *Siehe* CBL
Calculator-Based Ranger system
Anschließen 67
Calculator-Based Ranger. *Siehe* CBR
CATALOG (Menü) 205
CBL
Aktivität 845
holen/ermitteln, Get 904
Listenvariable senden, Send . . . 949
Programme 697, 845
Statistikdaten 624, 626
CBL 2 system
Anschließen 67
CBR
holen/ermitteln, Get 904
Listenvariable senden, Send 949
Programme 697, 845
Statistikdaten 624, 626
CBR system
Anschließen 67
ceiling( ), Dach 833, 870
cFactor( ), komplexer Faktor . . 277, 858, 871
CHAR (Zeichenmenü) 188
char( ), Zeichen-String 661, 871
CHAR-Menü (2nd [CHAR])
Eingeben von Sonderzeichen 11
Tastenbefehl 21
Chat senden, SendChat . 698, 809, 810, 950
checkTmr( ), Timer prüfen 872
Circle, Kreis zeichnen 695, 872
Circular definition-Fehler 654
Clean Up (Menü) 203
CLOCK Dialogfeld 46
ClockOff, Uhr ausschalten 872
ClockOn, Uhr einschalten 872
ClrDraw, Zeichnung löschen . . 535, 694, 872
ClrErr, Fehler löschen 701, 873
ClrGraph, Graph löschen . 519, 691, 750
ClrHome 32
ClrHome, Hauptbildschirm löschen . . 873

ClrIO, I/O löschen 632, 682, 873
colDim( ), Matrizenspaltenanzahl . . . 874
colNorm( ), absolute Summe Matrizenspalten 874
combinations, nCr( ) 925
comDenom( ), gemeinsamer Nenner 277, ... 278, 284, 874
complex
mode, Complex Format 1007
numbers 73
Complex (Menü) 277
Complex Format mode 1007
conj( ), komplexes Komplement 875
Coordinates (Graphenformat) . . 337, 371
CopyVar, Variable kopieren . . . 655, 875
cos( ), Cosinus 875
cos^{-1}( ), Arcuscosinus 876
cosh( ), Cosinus hyperbolicus 876
cosh^{-1}( ), Arcuscosinus hyperbolicus 877
Cosinus, cos( ) 875
cot( ), Kotangens 877
cot^{-1}( ), Arcuskotangens 877
coth( ), Kotangens hyperbolicus 877
coth^{-1}( ), Areakotangens (hyperbolicus) . 878
crossP( ), Produkt 878
csc( ), Kosekans 878
csc^{-1}( ), Arcuskosekans 878
csch( ), Kosekans hyperbolicus 878
csch^{-1}( ), Areakosekans (hyperbolicus) . . 879
cSolve( ), komplexe Gleichungen lösen . 259
cSolve( ), komplexe Lösung 879
CubicReg, kubische Regression 600, 881
cumSum( ), kumulative Summe 881
cumSum( ), Summe 584, 586
Cursor
außerhalb der Kurve 417
Auswählen eines Befehls 27
bewegen 17
Drei3D-Graph 413
Einträge anzeigen 30
frei beweglich . . 341, 372, 380, 390, 412, ... 456
Funktionen 17
Löschen eines Eingabe/Ergebnis-Paars 32
löschen von Zeichen 21
nach APD 7
steuern 156, 184
Trace 343, 344, 345, 346, 347
verdeckte Oberfläche 416
Verlauf 30
Cursortasten (◀▶▲▼)
Eingeben von Befehlen 27
Funktion 17
Öffnen von Apps 33
Verwenden des Menüs CHAR . . . 12
Verwenden mit der Hand-Taste . . 16
Zusatzfunktionen 17

CustmOff, ben.def. Menüleiste aus . 250, 881
CustmOn, ben.def. Menüleiste ein . . 250, 881
CUSTOM
Achsen (Folge) 392
benutzerdefinierte Plots . . . 385, 474, 476
Menü 189, 250
CUSTOM ([2nd] [CUSTOM]) Menü 61
Beschreibung 60
Tastenbefehl 20
Custom, Menüleiste definieren . 684, 882
Cycle, Zyklus 882
CyclePic, Bildzyklus . 547, 548, 692, 882
cZeros( ), komplexe Nullstellen 259, 277, 882

D

d( ), erste Ableitung . . 77, 268, 285, 287, 885
Dach, ceiling( ) 833, 870
Datei, neu ([♦] N) 20
Datei, öffnen ([♦] O) 20
Daten (neu), NewData 925
Daten-/Matrix-Editor
Auto-calculate 583
kopieren 589
neu, NewData 582, 656
shift, shift( ) 584, 585
Spalten sortieren 586, 587
Spaltenüberschrift . . . 579, 580, 582, 583, ... 584
statistische Plots . 605, 606, 609, 610
verlagern, shift( ) 956
Daten/Matrix-Editor 512
Bildlauf 575
DatenVariable . . . 569, 570, 572, 573
erzeugen 570, 572
füllen 576
Matrix-Variable . 568, 569, 570, 572, 573
Variable 569, 570, 572, 573
Variablenliste 565, 569, 570, 572, 573
Werte 574
Zellenbreite 578
Datenfilterung 841
Daten-Plots 118
Datum
Einstellung 45
Rücksetzen 53
Datum einstellen, setDate( ) 951
Datumsformat abrufen, getDtFmt( ) . 906
Datumsformat einstellen, setDtFmt( ) 951
Datumszeichenfolge abrufen, getDtStr( ) 906
dayOfWk( ), Wochentag 885
DE (DGL) Modus 43
Define, definieren . . 292, 377, 386, 487, 886

definieren, Define . . 242, 292, 329, 377, 386, . 408, 487, 516, 523, 649, 655, 691, ... 825, 886
definiert/undefiniert, Typ, getType( ) . 256, 656
DEG (GRD) Modus 43
DelFold, Verzeichnis löschen . . 656, 782, 887
DelVar, Variable löschen . 257, 291, 656, 660, ... 782, 887
Derivatives (math. Tool für Graphen) 355, ... 359, 381
deSolve( ), Lösung 286, 487, 888
det( ), Matrixdeterminante 890
dezimal
Winkel anzeigen, ▸DD 885
Zahl anzeigen, ▸Dec 886
diag( ), Matrixdiagonale 890
Diagramme, zeichnen auf 16
Dialog, Dialogfeld definieren . . . 683, 890
Dialogfeld
(◆ S (SAVE COPY AS) (KOPIE SPEICH ALS) 20
Bearbeiten von Kategorien 38
CLOCK 46
FORMATS/GRAPH FORMATS (FORMAT/GRAPHIK FORMATE) (◆ F) 20
Menüanzeige 57
MODE 22
öffnen von Apps 33
Dialogfeld, definieren, Dialog . . 683, 890
Dialogfelder 191
Differential- und Integralrechnungen . 862
Differentialgleichungen
Anfangsbedingungen 458
DIRFLD, Richtungsfeld 451
dritter Ordnung 471
erster Ordnung 463, 487
Fehlerbehebung 490
FLDOFF, Feld aus 451, 494
Graphen 445
Lösungsverfahren 450, 480
SLPFLD, Neigungsfeld 451, 491
zweiter Ordnung 466, 487
Differenzen auflisten, Δlist( ) 917
diftol (Fenstervariable) 454
dim( ), Anzahl 891
dim( ), Zeichenzahl 661
DIRFLD, Richtungsfeld 451, 492
Disp, I/O-Bildschirm anzeigen . 133, 640, 682, ... 700, 891
DispG, Graph anzeigen . . . 682, 691, 891
DispHome, Hauptbildschirm anzeigen . . 682, ... 891
Display Digits (Modus) 180
DispTbl, Tabelle anzeigen . 683, 690, 892
Distance (math. Tool für Graphen) . 356, 361, ... 373, 381
dividieren, / 984
dotP( ), Punktprodukt 892
Draht- und Konturmodelle 429

Drahtmodell 429
DrawFunc, Funktion zeichnen .532, 696, 892
DrawInv, invers zeichnen . 533, 696, 892
DrawParm, parametrisch zeichnen .532, 696, ... 893
DrawPol, polar zeichnen . . 532, 696, 893
DrawSlp, Neigung zeichnen . . .542, 696, 893
drehen, rotate( ) 662
dreidimensionale Graphen
Animation 97, 423
CONTOUR LEVELS 429
HIDDEN SURFACE 429
WIRE AND CONTOUR 429
WIRE FRAME 429
DropDown, Dropdown-Menü . . 684, 894
Dropdown-Menü, DropDown . . 684, 894
DrwCtour, Kontur zeichnen 434, 696, 894
dtime (Fenstervariable) 455
durchschn. Änderungsrate, avgRC( ) 869

E

E (Exponentensymbol) 18
e hoch x, *e*^( ) 895
e, Basis für ln 298
E/A-Anschluss 66
e^( ), *e* hoch x 895
echter Bruch, propFrac 75, 277, 838, 937
Eigenvektor, eigVc( ) 895
Eigenwert, eigVl( ) 895
eigVc( ), Eigenvektor 895
eigVl( ), Eigenwert 895
ein/aus 150, 151
einfügen 229, 230, 231, 232, 720
Einfügen-Modus ([2nd] [INS]) 21
Einführungen. *Siehe* Beispiele, Einführungen, Aktivitäten
Eingabe
Großschreibung von Buchstaben . 11
Eingabe, entry( ) 237, 896
Eingabe, Input 681, 691, 912
Eingabe/Ergebnis-Paare
Status 43
Eingabe-String, InputSt 661
Eingabezeile
Cursor 30
einfügen von Befehlen 26
Löschen des Verlaufs 32
rückholen 30
Eingeben
blättern im Katalog 26
Dateiname 33
Einheiten
festlegen, setUnits( ) 679, 954
holen/ermitteln, getUnits( ) .678, 909
Maße 299
Einheitensystem-Modus 22
Einheitenvektor, unitV( ) 974
Einschalten, Erster Start 2
einstellen Tabellenparameter 498
else if, ElseIf 522, 669, 896

Else, else 669, 911
else, Else 669, 911
ElseIf, else if 669, 896
end
for, EndFor 639, 672
if, EndIf 639, 666, 668, 669
while, EndWhile 674, 975
EndCustm, ben.def. Ende 684, 882
EndDlog, Ende Dialogfeld 683, 890
EndFor, end for 639, 672, 903
EndFunc, Funktion beeden 904
EndIf, end if . . . 639, 666, 668, 669, 911
EndLoop, Schleife beeden 920
EndLoop, Schleife beenden 676
EndPrgm, Programm beenden . 130, 649, 936
EndTBar, Ende Menüleiste 684, 972
EndTry, Ende Test 701, 973
EndWhile, end while 674, 975
Englisch-Kategorie 36
entry( ), Eingabe 237, 896
Ergebnisse 28
Ergebnisse mit zwei Variablen, TwoVar . 600, ... 973
ermitteln
Nenner, getDenom( ) 278
ermitteln *siehe* holen/ermitteln
ermitteln, Return 946
Ersetzung 269, 270, 272, 274
Erster Start 2
erweitern, expand( ) . 74, 276, 279, 828, 851, ... 898
erweitern/verketten, augment( ) 830, 868
erzeugen
Daten, BldData 480, 655, 870
Netz, Build Web 392
Tabelle, Table 690, 967
Estep (Fenstervariable) 454
Eulersches Verfahren 450, 480
EXACT (EXAKT) Modusstatus 43
exact( ), exakt 897
Exact/Approx (Modus) . . . 177, 218, 258, 259, ... 261
exakt, exact( ) 897
Exakt/Ungefähr Modus 22
Exec, Assembler-Programm ausführen . 709, ... 897
Exit, beenden 897
exklusives oder (Boole), xor . . . 759, 976
exp▶list( ), Ausdruck in Liste 897
expand( ), erweitern 276, 279, 898
Exponent, ᴇ 894
Exponentensymbol (ᴇ) 18
Exponententaste (2nd [EE]) 18
Exponential Format (Modus) 182
Exponential Format mode 1007
Exponentialformatmodus 22
exponentielle Regression, ExpReg . 601, 899
expr( ), String in Ausdruck 661, 662, 682, 899

ExpReg, exponentielle Regression .601, 899
Extract (Menü) 278
eyeϕ (Fenstervariable für z-Achse) . .409, 419, ... 421
eyeθ (Fenstervariable für x-Achse) . .409, 419, ... 420
eyeψ (Fenstervariable für Rotation) .409, 419, ... 421

F

factor( ), Faktor . . 74, 75, 259, 276, 279, 899
Faktor 279
Faktor, factor( ) 259, 276, 279, 829, 858, 899
Faktoren 75
  in der Praxis 858
Fakultät, ! 73, 988
false (Meldung) 296
Fehler nach APD 7
Fehler und Fehlerbehebung
  Circular definition 654
  Fehler löschen, ClrErr 701, 873
  Fehler weitergeben, PassErr . . .701, 934
  Memory Error 792, 793
  Out of Memory 294
  Programme 699
  Speicherfehler 792, 793
  Übertragung 806, 818
Fehler weitergeben, PassErr . . .701, 934
Feld aus, FLDOFF 451, 494
Fenster, switch( ) 679
Fenstervariable
  diftol 454
  dtime 455
  Estep 454
  eyeϕ (z-Achse) 409, 419, 421
  eyeθ (x-Achse) 409, 419, 420
  eyeψ (Rotation) 409, 419, 421
  fldres 454
  ncontour 410
  ncurves 454
  nmax 387
  nmin 387
  plotStep 388
  plotStrt 387
  θmax 370
  θmin 370
  θstep 370
  t0 452
  tmax 379, 453
  tmin 379
  tplot 453
  tstep 379, 453
  xgrid 410
  xmax . 335, 370, 379, 388, 409, 453
  xmin . 335, 370, 379, 388, 409, 453
  xres 335
  xscl 335, 370, 379, 388, 453
  ygrid 410

ymax . . 335, 370, 379, 388, 409, 453
ymin . . 335, 370, 379, 388, 409, 453
yscl 335, 370, 379, 388, 453
zmax 409
zmin 409
festlegen
Einheiten, setUnits( ) 679, 954
Graph, setGraph( ) . . . 679, 691, 951
Modus, setMode( ) . . . 679, 691, 952
Tabelle, setTable( ) . . 679, 690, 953
Tabellendefinition, setTable( ) . . 502
Verzeichnis, setFold( ) 678, 775, 951
Fibonacci-Folge 403
Fill, Matrix füllen 901
FLASH APPLICATIONS (♦ APPS)
Beschreibung 54
falls nicht installiert 25
Tastenbefehl 20
Zugriff auf nicht aufgeführte Apps 63
Flash-Anwendungen 194, 209, 295, 764, 765, ... 770, 778
löschen 806
Flashen, Aktualisieren des Betriebssystems 813, 814, 815
FLDOFF, Feld aus 451, 494
fldres (Fenstervariable) 454
floor( ), Unterstrich 833
fMax( ), Funktionshöchstwert . .259, 286, 901
fMin( ), Funktionsmindestwert .259, 286, 902
FnOff, Funktion aus 331, 691, 902
FnOn, Funktion ein 331, 691, 902
Folge, seq( ) 950
For, for 639, 672, 903
for, For 639, 672, 903
format( ), Format-String . . .662, 683, 903
FORMATS (♦ F)
Dialogfeld 20
Tastenbefehl 20
FORMATS (Dialogfeld) . . 426, 429, 431, 440, ...578, 725
FORMATS (Graphenformat) 337
Format-String, format( ) . . .662, 683, 903
Formelanzeige 177, 222
fPart( ), Funktionsteil 903
frei beweglicher Cursor . . 341, 372, 380, 390, ...412, 456
Frobenius-Norm, norm( ) 928
FUNC (FKT) Modus 43
Func, Programmfunktion 904
Funktionen 25, 168
aus, FnOff 331, 691, 902
benutzerdefiniert 210, 240, 292, 408, 519, . . .521, 643, 644, 645, 886
Definition 242
ein, FnOn 331, 691, 902
Graphen 322–365
Höchstwert, fMax( ) . . .259, 286, 901
Mindestwert, fMin( ) 259, 286
mit mehreren Anweisungen 521
Programmfunktion, Func 904

Teil, fPart( ) 903
Tiefstwert, fMin( ) 902
und ihre Bestandteile 521
verzögerte Vereinfachung 267
Funktionstasten (F1–F8)
Operationen 16
Ort 11
Wählen von Kategorien 32, 36
Wählen von Menüs 53
Wechseln zwischen Menüleistenmenüs 59
Funktionstasten (◀▶▲▼)
Ort 11
Funktionsteile 108

G

Ganzzahl teilen, intDiv( ) 913
Ganzzahl, int( ) 913
Ganzzahlteil, iPart( ) 95, 913
Ganzzahlteilung, intDiv( ) 757
gcd( ), größter gemeinsamer Teiler . 904
gehe zu, Goto . . 651, 666, 670, 677, 909
gemeinsamer Nenner, comDenom( ) 277, ... 278, 284, 874
Geräte-ID (Identifier) 254
Geräte-Verbindungskabel 71
Anschließen 67
Get, CBL/CBR-Wert holen/ermitteln . 624, 699, ... 904
GetCalc, Abrufen/Antworten . . . 809, 810
GetCalc, von Taschenrechner holen/ermitteln 698, 904
getConfg( ), Konfiguration holen/ermitteln 678, ... 905
getDate( ), aktuelles Datum abrufen . 905
getDenom( ), Nenner ermitteln 278
getDenom( ), Nenner holen/ermitteln 906
getDtFmt( ), Datumsformat abrufen . 906
getDtStr( ), Datumszeichenfolge abrufen 906
getFold( ), Verzeichnis holen/ermitteln . . 656, ... 678
getKey( ), Taste holen/ermitteln 681, 906
getMode( ), Modus holen/ermitteln . 678, 907
getNum( ), Zahl holen/ermitteln 907
getNum( ), Zahlen ermitteln 278
getTime( ), aktuelle Uhrzeit abrufen . 907
getTmFmt( ), Uhrzeitformat abrufen . 907
getTmStr( ), Uhrzeitzeichenfolge abrufen 907
getTmZn( ), Zeitzone abrufen 908
getType( ), Typ definiert/undefiniert 256, 656
getType( ), Typ holen/ermitteln 908
getUnits( ), Einheiten holen/ermitteln 678, ... 909
gilt für alle, | 258
gleich, = ... 664, 987
Gleichungen mit mehreren Unbekannten, simult( ) 958

Gleichungen, lösen 736
Gleichungssysteme 255
globale Variablen 659
Goto, gehe zu . . 651, 666, 670, 677, 909
Grad-/Minuten-/Sekundenanzeige, ▸DMS 892
Gradangabe, ° 991, 992
Grafik 108
Anzahl Modus 43
Graph 108
Kategorie 36
Modus 43
Modusstatus 43
Zahlenmodus 43
Grafikanzahl-Modus-Status 43
Grafiken
Trace 834, 845, 847, 851
zeichnen in 16
Grafikmodus 22
grafische Benutzeroberfläche, GUI . 683
Graph (Modus) 219, 324, 368, 376, 384, 407, ... 448
Graph <->Table, Tabellen-Graph . . . 499
GRAPH FORMATS (GRAPHIK FORMATE) (◆ F) 20
Dialogfeld 20
Graph Order (Graphenformat) . 337, 450
Graph, Graph . . 329, 517, 525, 691, 910
graph, Graph 691
Graphen
(Modi) 368
∫f(x)dx 356, 359
3D 405
Animation 547
Anzeigefenster . 334, 370, 379, 387, 409
Arc 356, 362, 373
aus Einstellungen zurückholen, RclGDB 692
aus Graphik-Einstellungen zurückholen, RclGDB 943
benutzerdefinierte Achsen 392
benutzerdefinierte Plots . . 385, 474, 476
Bilder 543, 546
Bildschirm teilen .526, 530, 554, 555
Darstellung, Graph 691
Derivatives 355, 359, 373, 381
Differentialgleichungen 445
Distance 356, 361, 373, 381
Doppelgraphmodus . . 526, 528, 554, 555
Fenstervariable . 334, 370, 379, 387, 409
festlegen, setGraph( ) . 679, 691, 951
Folgen 382
Formate 337, 371
freie, unabhängige Variable 514
Funktion aus, FnOff 691
Funktionen 322–365, 520
Funktionen aus, FnOff 902

Funktionen auswählen 330, 378, 386, ... 449
Funktionen ein, FnOn 691, 902
geschachtelte Funktion 520
Graph, Graph 517, 910
Graphik-Einstellungen 550
Hauptbildschirm 514, 517
implizite Plots 439, 442
in Einstellungen speichern, StoGDB 692
in Graphik-Einstellungen speichern, StoGDB 964
Inflection 356, 360
Intersection 355, 358
inverse Funktionen 533
Kontur-Plots 430, 434, 435
Koordinaten 341
Kurvenfamilie 523
Linienstile . . 332, 369, 378, 387, 449
löschen, ClrGraph 519, 691, 750, 873
mathematische Funktionen 355
Matrixdaten 512
Maximum 355, 357
mehrere Graphen gleichzeitig . . 525
Minimum 355, 357
Modi . . 219, 325, 376, 384, 407, 448
Netz-Plots 392, 393
Operationen 862
parametrisch 374
polar 366
Programme 691
schattieren, Shade 696, 955
schwenken 346
Shade 356, 363, 364
Stil, Style 691, 965
Tangent 356, 361, 373, 381
Text 542
Trace . 343, 344, 345, 346, 347, 372, 381, ... 391, 413, 457
Trace, Trace 343, 344, 345, 346, 347, ... 691, 973
Übersicht . . 322, 366, 374, 382, 405, 445
unabhängige Variable 514
unterbrechen 340
Value 355, 373, 381, 391, 413
web plots 385
Wert 357
Y= Editor . . 326, 369, 376, 385, 408, 448, ... 514
zeichnen . . 534, 536, 537, 538, 539, 540, ... 541, 542, 543, 693
Zeit-Plots 385, 392, 476
zentrieren 347
Zero 355, 357
Zoom . 348, 372, 380, 390, 412, 692
Zoom memory 350, 354
Zoom-Faktoren 350, 352
Graphen und Plotten
eingebettete Funktionen 520
Formate 450
Wert 457

Zeit-Plots 474
Graphen von Folgen 382
Grenzwert, limit( ) . . . 268, 286, 288, 915
Grid (Graphenformat) 338
griechische Buchstaben . . 725, 728, 730
Griechische Zeichen 11
größer als, > 664, 988
größer gleich, ≥, >= 664, 988
Großschreibung von Buchstaben 11
größter gemeinsamer Teiler, gcd( ) . 904
GUI, grafische Benutzeroberfläche . 683

H

Handheld-Software-Anwendungen (Apps) 4
Symbole 2
Hand-Modifikatortaste ([hand])
Beschreibung 16
Status 43
Hardwareversion 253
Hauptbildschirm 221
[2nd] [QUIT] 21
Funktionstasten 17
Tastenbefehl 21
zurück zur Apps-Arbeitsfläche . . . 45
Hauptbildschirm. *Siehe* Rechner Hauptbildschirm siehe Hauptbildschirm des Rechners
heller/dunkler 152
hexadezimal Hinweis, 0h 995
Hexadezimalformat anzeigen, ▸Hex . 754
Hexadezimalzahl anzeigen, ▸Hex . . . 910
Hilfsmittel (Utilities) Kategorie 37
Histogram 613
hoch 10, 10^( ) 993
hoch, ^ 990
holen/ermitteln
CBL/CBR-Wert, Get . . 624, 699, 904
Einheiten, getUnits( ) 678, 909
Konfiguration, getConfg( ) . . 678, 905
Modus, getMode( ) 678, 907
Nenner, getDenom( ) 906
Taste, getKey( ) 681, 906
Typ, getType( ) 908
Verzeichnis, getFold( ) 656, 678, 906
von Taschenrechner, GetCalc . 698, 904
Zahl, getNum( ) 907
Home-Symbol 28
hyperbolicus
Cosinus, cosh( ) 876
Sinus, sinh( ) 959
Tangens, tanh( ) 968

I

Identitätsmatrix, identity( ) 911
identity( ), Identitätsmatrix 911
ID-Liste 819, 821
ID-Nummer 252, 813, 814, 815, 819, 821
If, if 911
if, If . . 522, 639, 666, 667, 668, 669, 911
im String, mid( ) 662

imag( ), imaginärer Teil 911
imaginärer Teil, imag( ) 911
implizite Multiplikation 169, 377
implizite Plots 439, 442
in String, inString( ) 662, 912
Independent AUTO/ASK, automatisch/manuell 499
Inflection (math. Tool für Graphen) . . 356, 360
Input, Eingabe 681, 691, 912
InputSt, Eingabe-String . . 661, 682, 912
InputSt, Input-String 810
Input-String, InputSt 810
inString( ), in String 662, 912
int( ), Ganzzahl 913
intDiv( ), Ganzzahl teilen 757, 913
Integral, ∫( ) 77, 259, 261, 262, 268, 286, 287, ... 989
International/Betonungszeichen 11
Intersection (math. Tool für Graphen) . . 355, ... 358
invers, x^{-1} 994
iPart( ), Ganzzahlteil 95, 913
isClkOn( ), ist Uhr ein 913
isPrime( ), Primtest 914
ist Uhr ein, isClkOn( ) 913
Item, Menüobjekt 684, 687, 914

K

Kabel 67, 795, 815, 820
Kapitalwert 856
Karo-Modifikatortaste (♦)
  Beschreibung 15
  Status 43
Katalog (2nd [CATALOG])
  Befehle 26
  Beschreibung 25
  Tastenbefehl 21
  verlassen 28
Kategorie
  Alle 36
  Englisch 36
  Grafik 36
  Hilfsmittel (utilities) 37
  Mathe 36
  SocialSt (SozialWiss) 36
  Wissenschaft 37
Kategorien
  Apps-Arbeitsfläche 37
  auswählen 36
  Bearbeitungsbeispiel 39
  leere auswählen 37
  wählen 36
Klammern 169
kleiner als, < 664, 987
kleiner gleich, ≤, - 664
kleiner gleich, ≤, <= 988
kleinstes gemeinsames Vielfaches, lcm . 914
Kommentar, ◕ 638, 995
komplex
  Faktor, cFactor( ) 277, 871

Komplement, conj( ) 875
Lösung, cSolve( ) 879
Nullstellen, cZeros( ) . 259, 277, 882
Oberfläche 436
komplex Faktor, cFactor( ) 858
komplexe Gleichungen lösen, cSolve( ) . 259
Komplex-Format-Modus 22
Konstanten 299
vordefinierte 313
Kontrast anpassen 69
Kontrast, einstellen 152
Konturen 429
Kontur-Plots 430, 434, 435
Konvertieren von Ma?einheiten 20
konvertieren, ▸ 303, 993
Konvertierung, # 661, 990
kopieren 229, 230, 231, 232, 720
Kosekans hyperbolicus, csch( ) 878
Kosekans, csc( ), ... 878
Kotangens hyperbolicus, coth( ) 877
Kotangens, cot( ), ... 877
Kreis
zeichnen 539
Kreis, Circle 695, 872
kubische Regression, CubicReg 600, 881
Kugelvektoranzeige, ▸Sphere 963
kumulative Summe, cumSum( ) 881
Kurvenfamilie 523

L

Labels (Graphenformat) 338
Lbl, Bezeichnung 651, 666, 670, 677, 914
lcm, kleinstes gemeinsames Vielfaches . 914
Leading Cursor (Graphenformat) . . . 338
left( ), links 278, 662, 914
letzte Ausgabe 161, 174, 234, 237
letzte Eingabe 161, 234, 236
limit( ), Grenzwert 286, 288, 915
Line, Linie zeichnen 696, 916
Lineare Gleichungen lösen 75, 76
lineare Gleichungen lösen 281
lineare Regression, LinReg 601, 917
LineHorz, Horizontallinie zeichnen . 696, 916
LineTan, Tangente zeichnen . . . 696, 916
LineVert, Vertikallinie zeichnen . 696, 917
links, left( ) 278, 662, 914
Links/Rechts-Split-Screen
Status 41
LinReg, lineare Regression 601, 917
list▸mat( ), Liste in Matrix 583, 918
Liste in Matrix, list▸mat( ) 583, 918
Listen
Ausdruck in Liste, exp▸list( ) 897
Auto-calculate 583
Differenzen, Δlist( ) 917
erweitern/verketten, augment( ) . 868
erzeugen 570, 572

in absteigender Reihenfolge sortieren, SortD 963
in aufsteigender Reihenfolge sortieren, SortA 963
kopieren 589
kumulative Summe, cumSum( ) . 881
Liste in Matrix, list▸mat( ) . . 583, 918
Matrix in Liste, mat▸list( ) 921
Maximum, max( ) 922
Minimum, min( ) 924
neu, newList( ) 926
neue Daten, NewData 582, 656, 925
Operationen 862
Produkt, crossP( ) 878
Produkt, product( ) 936
Punktprodukt, dotP( ) 892
Spalten sortieren 586, 587
Spaltenüberschrift . . .579, 580, 582, 583, ... 584
Summe, cumSum( ) 584, 586
Summenbildung, sum( ) 966
Teil-String, mid( ) 923
Variable . . . 565, 569, 570, 572, 573
Listenvariable senden, Send . . 699, 949
ln( ), Logarithmus naturalis 918
LnReg, logarithmische Regression . . 601, 918
Local, lokale Variable 919
Lock, Variable sperren 656, 919
log( ), Logarithmus 919
Logarithmen 918, 919
logarithmische Regression, LnReg . 601, 918
Logarithmus naturalis, ln( ) 918
Logarithmus, log( ) 919
Logistic, logistische Regression 601, 920
logistische Regression, Logistic 920
lokale Variable, Local 647, 654, 656, 657, ... 658, 659, 919
Loop, Schleife 676, 920
Lösche Zeichen (⌫ / ◆ DEL) 21
löschen
Fehler, ClrErr 701, 873
Graph, ClrGraph .519, 691, 750, 873
Hauptbildschirm, ClrHome 873
I/O, ClrIO 632, 682, 873
Variable, DelVar 257, 291, 656, 660, 782
variable, DelVar 887
Verzeichnis, DelFold . . 656, 782, 887
Zeichnung, ClrDraw . . 535, 694, 872
lösen, solve( ) 76, 78, 259, 261, 262, 268, ... 273, 276, 281, 282, 487, 960
Lösung, deSolve( ) 286, 487, 888
Lösungsverfahren Graphikformat . . . 450
LU, untere/obere Matrixzerlegung . . . 921

M

Markieren von Text 183
Maßeinheiten 299
anzeigen 307
benutzerdefinierte 310

konvertieren 303
Standardvorgaben 307
mat▸list( ), Matrix in Liste 921
MATH (Menü) 188, 355
Mathe Kategorie 36
mathematische Operationen 863
Mathematische Zeichen 11
Matrix in Liste, mat▸list( ) 921
Matrizen
Siehe auch Daten/Matrix-Editor
absolute Spaltensumme, colNorm( ) 874
Anzahl, dim( ) 891
Auto-calculate 583
Daten aus Graph 512
Determinante, det( ) 890
Diagonale, diag( ) 890
Eigenvektor, eigVc( ) 895
Eigenwert, eigVl( ) 895
erweitern/verketten, augment( ) . 830, 868
erzeugen 570, 572
füllen, Fill 901
Funktionsausdruck 569
Identität, identity( ) 911
kopieren 589
kumulative Summe, cumSum( ) . 881
list to matrix, list▸mat( ) 918
Matrix in Liste, mat▸list( ) 921
Maximum, max( ) 922
Minimum, min( ) 924
neu, newMat( ) 926
neue Daten, NewData 656, 925
Operationen 863
Produkt, product( ) 936
Punktaddition, .+ 985
Punktdivision, ./ 986
Punktmultiplikation, .* 985
Punktpotenz, .^ 986
Punktsubtraktion, .− 985
QR-Faktorisierung, QR 941
schützen 581
Spalten sortieren 586, 587
Spaltenanzahl, colDim( ) 874
Spaltenüberschrift . . . 579, 580, 582, 583, ... 584
Staffelung, rref( ) 282, 830
Summe, cumSum( ) 584, 586
Summenbildung, sum( ) 966
Teilmatrix, subMat( ) 965
transponieren, ᵀ, transponieren . 967
untere/obere Zerlegung, LU 921
Variable 568, 569, 570, 572, 573
willkürlich, randMat( ) 830
Zeilen in Staffeln anzeigen, ref( ) 945
Zeilen reduzieren, rref( ) 948
Zeilen vertauschen, rowSwap( ) . 948
Zeilenaddition, rowAdd( ) 947
Zeilenanzahl, rowDim( ) 947
Zeilenmultiplikation und -addition, mRowAdd( ) 924
Zeilennorm, rowNorm( ) 948

Zeilenoperation, mRow( ) 924
Zufall, randMat( ) 943
max( ), Maximum 922
Maximum (math. Tool für Graphen) . 355, 357
Maximum, max( ) 922
mean( ), Mittelwert 922
median( ), Mittelelement 922
MedMed, Mittellinienregression 601, 923
Meldungen
Aufräumen 787, 788, 789
BATT 219
false 296
nicht genügend Display-Speicher, <<...>> 248
true 296
undef (undefiniert) 298
MEMORY (Speicher) (2nd [MEM]) 21
Memory (Zoom) 350, 354
Menüleiste
aus, CustmOff 250, 881
definieren, Custom 684, 882
ein, CustmOn 250
Menüleiste definieren, Toolbar . 684, 972
Menüleisten-Menüs
auswählen mathematischer Operationen 17, 28
ersetzt durch Anwendermenüs . . 60
Rechner Hauptbildschirm 53
wechseln zwischen 59
Menüobjekt, Item 684, 687, 914
Menüs
Algebra 275, 276, 278
APPLICATIONS 188, 193
APPLICATIONS ([APPS]) 63
auswählen von Optionen 55
benutzerdefiniert 685, 689
Calc 285
CATALOG 205
CHAR 11, 21
CHAR (Zeichen) 188
Clean Up 203
Complex 277
CUSTOM 189, 250
CUSTOM (2nd [CUSTOM]) . . 20, 60, 61
Extract 278
FLASH APPLICATIONS ([♦] [APPS]) . 54
Flash Applications (Flash-Applikationen)([♦] [APPS]) 20
MATH 188, 355
Menüleiste 250
Optionen 15
Trig 277
Untermenüoptionen 56
verlassen 59
Messung, Umwandlung (2nd [▸]) 20
mid( ), im String 662
mid( ), Teil-String 923
min( ), Minimum 924
Minimum (math. Tool für Graphen) . 355, 357

Minimum, min( ) 924
Minute Angabe, ' 992
Mittelelement, median( ) 922
Mittellinienregression, MedMed 601, 923
Mittelwert, mean( ) 922
mod( ), Rest 924
modes
Angle 1007
Complex Format 1007
Exponential Format 1007
Modi 198
3D 43
Aktuell 22
Angle 325
Anzeigestellen 22
Approximate 177, 218, 260
Apps-Arbeitsfläche 22, 44
AUTO 43
Auto 177, 218, 261
Basis 22
Benutzereinheiten 22
DE (DGL) 43
DEG (GRD) 43
Display Digits 180
Einfügen (2nd [INS]) 21
Einheitensystem 22
Einstellungen 22
EXACT (EXAKT) 43
Exact/Approx . . 177, 218, 258, 259, 260, ... 261
Exakt/Ungefähr 22
Exponential Format 182
Exponentialformat 22
festlegen, setMode( ) .679, 691, 952
FUNC (FKT) 43
Grafik 22
Grafikanzahl 43
Grafiktyp 43
Graph 219, 324, 368, 376, 384, 407, 448
grau 22
holen/ermitteln, getMode( ) .678, 907
in Programmen festlegen 678
Komplex-Format 22
PAR 43
POL 43
Pretty Print 22, 177
RAD 43
SEQ (FOLGE) 43
Split-Screen . . 3, 20, 22, 35, 41, 43
Sprache 22, 23
Überschreiben (2nd [INS]) 21
UNGEFÄHR 43
Vektor-Format 22
Vollbild 21, 35, 41
Winkel 22, 43
Modifikatortasten (2nd ◆ ⬆ ⎋) 15
Ort 11
Status 43
MoveVar, Variable verschieben 656, 924
mRow( ), Matrixzeilenoperation 924

mRowAdd( ), Matrixzeilenmultiplikation und -addition 924
multiplizieren, * 984

N

nähern, approx( ) 277, 868
ncontour (Fenstervariable) 410
nCr( ), Kombinationen 925
ncurves (Fenstervariable) 454
nDeriv( ), numerische Ableitung 286, 925
negativ, - 165
Negative Zahlen 17
negatives Vorzeichen, - 986
Neigungsfeld, SLPFLD 451, 491
Nenner 874
Netz erzeugen, Build Web 394
Netz-Plots
  Divergenz 397
  WEB 385, 392, 393
neu
  Aufgabe, NewProb 204, 927
  Bild, NewPic 656, 692, 926
  Daten, NewData . 582, 626, 656, 925
  Liste, newList( ) 926
  Matrix, newMat( ) 926
  Plot, NewPlot 691, 927
  plot, NewPlot 612
  Verzeichnis, NewFold 656, 926
neu Verzeichnis, NewFold 775
Neue Datei (◆ N) 20
NewData, neue Daten 582, 626, 656, 925
NewFold, neuer Verzeichnis . . 656, 775, 926
newList( ), neue Liste 926
newMat( ), neue Matrix 926
NewPic, neues Bild 656, 692, 926
NewPlot, neuer Plot 612, 691, 927
NewProb, neue Aufgabe 204, 927
nicht (Boole), not 666, 758, 928
nicht genügend Display-Speicher, <<...>> 248
nInt( ), numerisches Integral . . . 286, 927
nmax (Fenstervariable) 387
nmin (Fenstervariable) 387
norm( ), Frobenius-Norm 928
not, (Boole) nicht 758, 928
not, Boole nicht 666
nPr( ), Permutationen 929
nSolve( ), numerische Lösung 929
nSolve( ), numerisches Lösung 277
Nullstellen
  in der Praxis 850
  zeros( ) 259, 276, 283, 826, 976
numerisch
  Ableitung, nDeriv( ) 925
  Integral, nInt( ) 927
  Lösung, nSolve( ) 929
Numerische Tastatur 17
  Stelle 11
numerischer Gleichungslöser 736
  Bildschirm teilen 749
  Gleichungen 736, 738, 740

grafische Darstellung .748, 749, 750, 751, ... 752
Variablen 740

O

Oben/Unten-Split-Screen
Status 41
oder (Boole), or 665, 758, 930
Öffnen Datei (♦ O) 20
OneVar, Statistik mit einer Variablen 600, ... 930
Operatoren 167
or, (Boole) oder 758, 930
or, Boole oder 665
ord( ), numerischer Zeichen-Code . . 662
ord( ), Zahlen-Code für Zeichen 931
Ordner
übertragen 798, 799, 802, 804
Out of Memory (Fehler) 294
Output, Ausgabe 683, 931

P

P▶Rx( ), x-Koordinate 932
P▶Ry( ), y-Koordinate 932
PAR Modus 43
Parallelepiped 833
parametrische Graphen 374
part( ), Teil 932
Partialbrüche, propFrac 284
PassErr, Fehler weitergeben . . 701, 934
PAUSE 44
PAUSE (Zeiger) 220
Pause, Pause 683, 700, 934
Permutationen, nPr( ) 929
Pixel
ändern, PxlChg 695
aus, PxlOff 695, 939
ein, PxlOn 543, 695, 939
Horizontallinie, PxlHorz 696, 939
Kreis um, PxlCrcl 938
Kreis, PxlCrcl 695
Linie, PxlLine 543, 696, 939
Test, pxlTest( ) 695, 940
Text, PxlText 695, 940
umwandeln, PxlChg 938
Vertikallinie, PxlVert 696, 940
Platzhalter für Ganzzahl, @ 297
Plots
Anzeigefenster 617
aus, PlotsOff 331, 691, 935
auswählen 609, 616
ein, PlotsOn 331, 691, 935
löschen 610
neu, NewPlot 612, 691, 927
Trace 618
Y = Editor 615
Plots, Daten 118
PlotsOff, Plots aus 691, 935
PlotsOn, Plots ein 691, 935
plotStep (Fenstervariable) 388
plotStrt (Fenstervariable) 387
POL Modus 43

polar
Graphen 366
Koordinate, R▸Pr( ) 942
Vektor anzeigen, ▸Polar 935
polyEval( ), Polynom berechnen 935
Polynom berechnen, polyEval( ) 935
Polynome 75, 279, 289
berechnen, polyEval( ) 935
in der Praxis 850
Zufall, randPoly( ) 943
PopUp, Popup-Menü 682, 936
Popup-Menü, PopUp 682, 936
Potenzregression, PowerReg . . 601, 936
PowerReg, Potenzregression . . 601, 936
Pretty Print 82, 86
Modus 22
Pretty Print (Modus) 177
Prgm, Programm ausführen . . .130, 649, 936
Prim, ' 992
Primzahlen 74
Primzahltest, isPrime( ) 914
product( ), Produkt 936
Produkt, crossP( ) 878
Produkt, Π( ) 286, 990
Produkt, product( ) 936
Produkt-ID 252, 253
Produkt-ID (Identifier) 253
Programm ausführen, Prgm . . .130, 649, 936
Programme 25, 628–710
anderes Programm aufrufen . . . 648, 649, ... 650
Anforderung, Request 682, 684
Argumente 642
Assembler 706, 707, 708, 709
Assembler-Programm ausführen, Exec 709, 897
Aufforderung, Prompt( ) . . . 682, 937
ausführen 628
ausführen, Prgm 936
Ausgabe 631, 640, 682
Ausgabe, Output 683, 931
Bedingungstests 663
beenden, EndPrgm 936
beenden, Exit 897
ben.def. Ende, EndCustm 684
ben.def. Menüleiste aus, CustmOff . . 250, ... 684, 881
ben.def. Menüleiste ein, CustmOn . . 250, ... 684, 881
ben.def. Menüleiste Ende, EndCustm 882
Bezeichnung, Lbl 651, 666, 670, 677, ... 914
CBL 697, 845
CBR 697, 845
Debugging 700
definieren, Define 649, 691, 825, 886
Dialogfeld beenden, EndDlog . . 683, 890
Dialogfeld definieren Dialog 683, 890

Dropdown-Menü, DropDown . . . 684, 894
E/A-Bildschirm anzeigen, Disp . . 133
Eingabe 631, 641, 681
Eingabe, Input 681, 691, 912
eingeben . 633, 637, 638, 639, 640, 641
Einheiten holen/ermitteln, getUnits( ) 678, ... 909
else if, ElseIf 522, 669, 896
else, Else 669, 911
end for, EndFor 639, 672, 903
end if, EndIf 639, 666, 668, 669, 911
end while, EndWhile 674, 975
Ende Menüleiste, EndTBar 684
Ende Test, EndTry 701, 973
Fehler löschen, ClrErr 701, 873
Fehler weitergeben, PassErr . . . 701, 934
for, For 639, 672, 903
Format-String, format( ) . . . 683, 903
Funktion Ende, EndFunc 904
Funktion, Func 904
Funktionen 633, 643, 644, 645
gehe zu, Goto . . 651, 666, 670, 677, 909
grafische Benutzeroberfläche, GUI . 683
Graph anzeigen, DispG . . . 682, 691, 891
Graph löschen, ClrGraph . 519, 691, 873
Graphen 691
Hauptbildschirm anzeigen, DispHome 682, 891
Hauptbildschirm löschen, ClrHome . . 873
I/O löschen, ClrIO 632, 682, 873
I/O-Bildschirm anzeigen, Disp . . 640, 682, ... 700, 891
if, If 522, 639, 666, 667, 668, 669, 911
Kommentar, ◕ 638, 995
Konfiguration holen/ermitteln, getConfg( ) 678, 905
kopieren 636
lokal, Local 647, 654, 656, 657, 658, 659, ... 919
looping 671, 672
löschen 636
Menüleiste definieren, Custom . 684, 882
Menüleiste definieren, Toolbar . . 684
Menüleiste Ende, EndTBar 972
Menüobjekt, Item 684, 687, 914
Menüs 685, 689
Modus holen/ermitteln, getMode( ) . . 678, ... 907
Operationen 864
Pause, Pause 683, 700, 934
Popup-Menü, PopUp 682, 936
Programm ausführen, Prgm 130, 649

Programm beenden, EndPrgm . 130, 649
return, Return 647, 650, 946
Schleife beenden, EndLoop 676, 920
Schleife, Loop 674, 676, 920
Schleifen 639
Stop, Stop 637, 964
stoppen 630
Subroutinen 648, 649, 650
Tabelle anzeigen, DispTbl . 683, 690, 892
Tabelle löschen, ClrTable 873
Tabellen 690
Taschenrechner verbinden 697
Taste holen/ermitteln, getKey( ) . 681, 906
Test, Try 701, 973
Text, Text 683, 684, 970
Then, Then 666, 668, 669, 911
Titel, Title 684, 971
Variablen 651, 652
Verzeichnis holen/ermitteln, getFold( ) 678, 906
verzweigen 639, 666, 670
von Taschenrechner holen/ermitteln, GetCalc 698, 904
Wert an Programm übermitteln . 642
while, While 674, 975
Programme und Programmieren
Abrufen/Antworten, GetCalc 809, 810
Programm-Editor 34
Prompt( ), Aufforderung 682, 937
propFrac, echter Bruch 75, 277, 284, 838, ... 937
Protokollanzeige 31
Protokoll-Bereich 223, 224, 225, 226, 734
Prozent, % 986
PtChg, Punkt ändern 695
PtChg, Punkt umwandeln 937
PtOff, Punkt aus 695, 937
PtOn, Punkt ein 695, 938
ptTest( ), Punkttest 695, 938
PtText, Punkttext 695, 938
Punkt
Addition, .+ 985
ändern, PtChg 695
aus, PtOff 695, 937
Division, ./ 986
ein, PtOn 695, 938
Multiplikation, .* 985
Potenz, .^ 986
Produkt, dotP( ) 892
Test, ptTest( ) 695, 938
Text, PtText 695, 938
umwandeln, PtChg 937
Punktsubtraktion, .− 985
PxlChg, Pixel ändern 695
PxlChg, Pixel umwandeln 938
PxlCrcl, Pixelkreis 695, 938
PxlHorz, Pixelhorizontallinie . . . 696, 939
PxlLine, Pixellinie 543, 696, 939
PxlOff, Pixel aus 695, 939

PxlOn, Pixel ein 543, 695, 939
pxlTest( ), Pixeltest 695, 940
PxlText, Pixeltext 695, 940
PxlVert, Pixelvertikallinie 696, 940

Q

QR, QR-Faktorisierung 941
QR-Faktorisierung, QR 941
quadratische Regression, QuadReg 602, 941
Quadratwurzel, √( ) 989
QuadReg, quadratische Regression 602, 941
quartische Regression, QuartReg . . 602, 942
QuartReg, quartische Regression . . 602, 942
QuickCenter 347
Quit ([2nd] [QUIT], Beenden) 21
QWERTY-Tastatur 11

R

r, Radians 991
R▸Pθ( ), Polarkoordinate 942
R▸Pr( ), Polarkoordinate 942
RAD Modus 43
Radians, r 991
rand( ), Zufallszahl 943
randMat( ), willkürliche Matrix 830
randMat( ), Zufallsmatrix 943
randNorm( ), Zufallsnorm 943
randPoly( ), Zufallspolynom 943
RandSeed, Werkseinstellungen für Zufallszahlengenerator 943
RandSeed, Zufallszahl 830
rationale Funktionen 838
RclGDB, aus Graphik-Einstellungen zurücholen 692
RclGDB, aus Graphik-Einstellungen zurückholen 943
RclGDB, Graphik-Einstellungen zurückholen 552
RclPic, Bild zurückholen 692, 944
real( ), reell 944
Rechner Hauptbildschirm
  Anwendermenü 60
  Ausschalten des Rechners 6
  Menüleisten-Menüs 53
Rechteckvektoranzeige, ▸Rect 944
rechts, right( ) 278, 662, 946
reell, real( ) 944
ref( ), Zeilen in Staffeln anzeigen . . . 945
Regressionen 917
  Ableitung einer Gleichung zweiter Ordnung 826
  auswählen 600
  exponential, ExpReg 601
  exponentiell, ExpReg 899
  kubisch, CubicReg 600, 881
  linear, LinReg 917
  lineare Regression, LinReg 601
  logarithmisch, LnReg 601, 918

logistisch, Logistic 601, 920
Mittellinie, MedMed 601, 923
Potenz, PowerReg 601
Potenzregression, PowerReg . . 936
quadratisch, QuadReg 602, 941
quartisch, QuartReg 602, 942
Sinus, SinReg 960
sinusoidal, SinReg 602
remain( ), Rest 945
Rename, umbenennen 656, 945
Request, Anforderung . . . 682, 684, 945
Rest, mod( ) 924
Rest, remain( ) 945
Return, ermitteln 946
Return, return 647, 650
return, Return 522, 647, 650
Revision des Zertifikats (Cert. Rev.) . 254
reziprok, x^{-1} 994
Richtungsfeld, DIRFLD 451, 492
right( ), rechts 278, 662, 946
rotate( ), rotate 662, 760
rotate( ), tauschen 946
rotate, rotate( ) 760
round( ), runden 947
rowAdd( ), Matrizenzeilenaddition . . 947
rowDim( ), Zeilenanzahl in Matrix . . . 947
rowNorm( ), Matrixzeilennorm 948
rowSwap( ), Zeilen in Matrix vertauschen 948
RplcPic, Bild ersetzen 692, 948
rref( ), Staffelung 282, 830, 948
Rückholen ([2nd] [RCL]) 21
Rücktaste ([←]) 21
runden, round( ) 947
Runge-Kutta-Verfahren . . . 450, 476, 480

S

SAVE COPY AS (KOPIE SPEICH ALS) ([♦] S)
Beispiel 58
Beschreibung 20
Dialogfeld 20
Scatter-Plots 611
schattieren, Shade 696, 955
Schleife, Loop 676, 920
schwenken 346
sec( ), Sekans 949
sec^{-1}( ), Arcussekans 949
sech( ), Sekans hyperbolicus 949
$sech^{-1}$( ), Areasekans (hyperbolicus) . 949
Sekans hyperbolicus, sech( ) 949
Sekans, sec( ), ... 949
Send, Listenvariable senden . . . 699, 949
SendCalc, An Rechner senden . 809, 810
SendCalc, an Taschenrechner senden . . 698, ... 950
SendChat, Chat senden . 698, 809, 810, 950
SEQ (FOLGE) Modus 43
seq( ), Folge 950
Seriennummer 253
Set factors (Zoom) 350

set time,Zeit einstellen ( ) 953
setDate( ), Datum einstellen 951
setDtFmt( ), Datumsformat einstellen 951
setFold( ), Verzeichnis festlegen . . . 678, 775, ... 951
setGraph( ), Graph festlegen . . 679, 691, 951
setMode( ), Modus festlegen . . 679, 691, 952
setTable( ), Tabelle definieren 502
setTable( ), Tabelle festlegen . . 679, 690, 953
setTime( ), Zeit einstellen 953
setTmFmt( ), Zeitformat einstellen . . 953
setTmZn( ), Zeitzone einstellen 954
setUnits( ), Einheiten festlegen . 679, 954
Shade (math. Tool für Graphen) . . . . 356, 363, ... 364
Shade, schattieren 696, 955
shift( ), shift 584, 585, 663, 761
shift( ), verlagern 956
shift, shift( ) 584, 585, 761
ShowStat, Statistikergebnisse anzeigen 603, ... 957
sign( ), Vorzeichen 957
simult( ), Gleichungen mit mehreren Unbekannten 958
simult( ), simultane Gleichungslösung . . 282
simultane Gleichungslösung, simult( ) . . 282
sin( ), Sinus 958
$\sin^{-1}$( ), Arcussinus 959
sinh( ), Sinus hyperbolicus 959
$\sinh^{-1}$( ), Arcussinus hyperbolicus . . . 959
SinReg, sinusoidale Regression 602, 960
Sinus, sin( ) 958
sinusoidal regression, SinReg 602
sinusoidale Regression, SinReg 960
Skripts . . . 227, 731, 732, 733, 734, 735
Aktivität 836
Beispiel 836
SLPFLD, Neigungsfeld 451, 491
Smart Graph 340
SocialSt (SozialWiss) Kategorie 36
Software-Version 252, 253
solve( ), lösen 76, 78, 259, 261, 262, 268, ... 273, 276, 281, 282, 487, 960
Sonderzeichen 165, 723, 726
SortA, in aufsteigender Reihenfolge sortieren 963
SortD, in absteigender Reihenfolge sortieren 963
sortieren
in absteigender Reihenfolge, SortD . 963
in aufsteigender Reihenfolge, SortA . 963
Speicher 151, 764–793
archivieren, Archive . . 655, 785, 868
aus Archiv holen, Unarchiv . 785, 974
aus Archiv, Unarchiv 656

nicht genügend Display-Speicher, <<...>> 248
überprüfen 764, 765
VARLINK Bildschirm .767, 769, 770, 771, ... 778, 785
zurücksetzen 764, 765
speichern
Bild, StoPic 692, 965
Graphik-Einstellungen, StoGDB . 552
in Einstellungen, StoGDB 692
in Graphik-Einstellungen, StoGDB . . 964
Symbol, → 655, 995
Sperrung aufheben, Unlock . . . 656, 974
Spinnennetz-Plot. *Siehe* Netz-Plots
Split-Screen-Modus
Aktive Grafik 43
Anzeige 22
Befehl Wechsel der aktiven App . 20
Status 41
Status und offene Apps 3
zurück aus einer App 35
Sprachmodus
Ändern von Moduseinstellungen . 23
Anzeige 22
Staffelung, rref( ) 282, 830
Standardabweichung, stdDev( ) 964
Standardrente 854
startTmr( ), Timer starten 963
Statistik 592
Siehe auch Regressionen
Box Plot 612
Calculation Type 595, 596, 600
Category 595, 596
Ergebnisse anzeigen, ShowStat 603, 957
Ergebnisse mit zwei Variablen, TwoVar 600, 973
Fakultät, ! 73, 988
Freq 595, 596
Häufigkeit 619
Histogram-Plots 613
Kategorien 621
Kombinationen, nCr( ) 925
mit einer Variablen, OneVar 930
Mittelelement, median( ) 922
Mittelwert, mean( ) 922
neu plot, NewPlot 612
neuer Plot, NewPlot 927
Operationen 865
Permutationen, nPr( ) 929
Plots . 605, 606, 609, 610, 611, 612, 613, . . .614, 615, 616, 617, 618
Plots aus, PlotsOff . . .331, 691, 935
Plots ein, PlotsOn 331, 691, 935
Scatter-Plots 611
Standardabweichung, stdDev( ) . 964
Statistik mit einer Variablen, OneVar 600
Überblick 592
Variable 603
Variablen 598

Varianz, variance( ) 974
Werkseinstellungen für Zufallszahlengenerator, RandSeed 943
xyline-Plots 611
Zufallsnorm, randNorm( ) 943
Zufallszahl, rand( ) 943
Zufallszahl, RandSeed 830
Statistik mit einer Variablen, OneVar 600
Statistikergebnisse anzeigen, ShowStat 603, ... 957
Status
auf Apps-Arbeitsfläche 3, 21
Batterie schwach 69
Split-Screen 41
Statuszeile 216, 217, 325
Befehlspaprameter 27
Protokollinfo 31
stdDev( ), Standardabweichung 964
Stil, Style 334, 691, 965
StoGDB, Graphik-Einstellungen speichern 552
StoGDB, in Einstellungen variablen speichern 692
StoGDB, in Graphik-Einstellungen speichern 964
Stop, Stop 637, 964
StoPic, Bild speichern 692, 965
Store (Speichern) (() Taste 21
String eingeben, InputSt 682, 912
string( ), Ausdruck in String . . . 663, 965
Strings
anhängen, & 661, 988
Ausdruck in String, string( ) . 663, 965
drehen, rotate( ) 662
Eingeben, InputSt 810
eingeben, InputSt 661, 682
Format, format( ) 662, 683, 903
im String, mid( ) 662
in Ausdruck, expr( ) 682
in String, InString 662
konvertieren, # 661
Konvertierung, # 990
links, left( ) 662, 914
Operationen 660, 661, 865
rechts, right( ) 662, 946
String in Ausdruck, expr( ) 661, 662, 899
tauschen, rotate( ) 946
Teil-String, mid( ) 923
umlagern, shift( ) 663
verlagern, shift( ) 956
Zeichen-Code, ord( ) 662, 931
Zeichen-String, char( ) 661, 871
Zeichenzahl, dim( ) 661
strings
innerhalb, InString 912
Style, Stil 691, 965
subMat( ), Teilmatrix 965
subtrahieren, − 983
Subtraktionstaste (() 17
sum( ), Summenbildung 966

Summe, Σ( ) 286
Summe, cumSum( ) 584, 586
Summe, Σ( ) 990
Summenbildung, sum( ) 966
switch( ), aktives Fenster 966
switch( ), Fenster 679
Systemdaten, sysData 512, 513

T

T, transponieren 967
t0 (Fenstervariable) 452
Tabellen 496–511
  Δtbl 498
  Anfang, tblStart 498
  anzeigen, DispTbl . . . 683, 690, 892
  ASK 507
  automatische Erstellung 502
  Differentialgleichungen 495
  erzeugen, Table 690, 967
  festlegen, setTable( ) . 679, 690, 953
  Graph, Graph<->Table 499
  Independent AUTO/ASK 499
  Intervall, Δtbl 498
  konfigurieren 502
  löschen, ClrTable 873
  manuell erstellen 507
  Parameter 498
  Programme 690
  setTable( ) 502
  tblStart 498
  Übersicht 496
  Zellenbreite 504
Tabellen-Graph, Graph<->Table 499
Table, Tabelle erzeugen 690, 967
tan( ), Tangens 968
tan⁻¹( ), Arcustangens 968
Tangens, tan( ) 968
Tangent (math. Tool für Graphen) . . 356, 361, ... 373, 381
tanh( ), Tangens hyperbolicus 968
tanh⁻¹( ), Arcustangens hyperbolicus . 969
Tastatur 153, 155
  [⎋] taste (hand) 157
  [2nd] taste (zweite) 156
  [⇧] taste (shift) 156
  [◆] taste (karo) 157
  Belegung 11, 13
  [alpha] taste 157
  QWERTY 11
  Tasten-Codes 681
Tastaturbelegung 724, 726
Tasten
  Änderung 11, 15
  Cursor 11, 17
  Funktion 11, 16
  weitere 19
Tastenbefehle
  Sonderzeichen 11
  Tastaturbelegung 13
tauschen, rotate( ) 946
taylor( ), Taylor-Polynom . . 286, 289, 969
Taylor-Polynom, taylor( ) . . 286, 289, 969

tblStart, Tabellenanfang 498
tCollect( ), Trigonometrie 277
tCollect( ), trigonometrische Komprimierung 970
Teil, part( ) 932
Teilmatrix, subMat( ) 965
Teil-String, mid( ) 923
Temperatur konvertieren, tmpCnv( ) 971
Temperaturangaben, ° 849
Temperaturbereichskonvertierung, ΔtmpCnv( ) 307
Temperaturekonvertierung, tmpCnv( ) . . 306
Test, Try 701, 973
tExpand( ), Trigonometrie 277
tExpand( ), trigonometrische Erweiterung 970
Text markieren 719
Text, Text 683, 684, 970
Textbearbeitung 711
ausschneiden, kopieren, einfügen . . 229, ... 230, 231, 232, 720
markieren 719
suchen 721
Then, Then 666, 668, 669, 911
TI Connect-Software 64
TI ViewScreen OHP Anschließen 67
TI Connect Software 815
TI-Connectivity-Kabel 795, 815, 820
TIME, Zeit-Plots 385, 392, 474, 476
timeCnv( ), Uhrzeit umwandeln 971
Timer prüfen, checkTmr( ) 872
Timer starten, startTmr( ) 963
TI-Presenter Videoadapter Anschließen 67
Titel, Title 971
Title, Titel 971
tmax (Fenstervariable) 379, 453
tmin (Fenstervariable) 379
tmpCnv( ), Temperatur konvertieren . 971
tmpCnv( ), Temperaturkonvertierung 306
Toolbar, Menüleiste 684, 972
tplot (Fenstervariable) 453
Trace 343, 344, 345, 346, 347, 372, 391, ... 413, 457
Trace, Trace . . 691, 834, 845, 847, 851, 973
Tracing 381
transmitting. *Siehe* Iverbinden und übertragen
transponieren, T 967
Trig (Menü) 277
Trigonometrie, tCollect( ) 277
Trigonometrie, tExpand( ) 277
trigonometrische Erweiterung expansion, tExpand( ) 970
trigonometrische Komprimierung, tCollect( ) 970
Try, Test 701, 973
tstep (Fenstervariable) 379, 453

U

Überschreiben-Modus ([2nd] [INS]) 21
Uhr
abschalten 51
Betrieb 45
Uhr ausschalten, ClockOff 872
Uhr einschalten, ClockOn 872
Uhrzeit umwandeln, timeCnv( ) 971
Uhrzeitformat abrufen, getTmFmt( ) . 907
Uhrzeitzeichenfolge abrufen, getTmStr( ) 907
umbenennen, Rename 656, 945
umlagern, shift( ) 663
Umschalt-Modifikatortaste ([⇧])
Beschreibung 15
Status 43
Unarchiv, Variablen aus Archiv 656
Unarchiv, Variablen aus Archiv holen . . . 785, ... 974
und (Boole), and 270, 665, 758, 866
und Bild, AndPic 692, 867
undef (undefiniert) (Meldung) 298
unendlich, ∞ 298
ungleich, ≠, /= 664, 987
unitV( ), Einheitenvektor 974
Unlock, Sperrung aufheben . . . 656, 974
Untermenüs 190
Unterstrich, _ 992
Unterstrich, floor( ) 833

V

Value (math. Tool für Graphen) 355, 357, 373, ... 381, 413
Variable 212, 214, 215, 216
aus Archiv holen, Unarchiv 974
aus Archiv, Unarchiv 656
Daten 567, 569, 570, 572, 573
definiert 255, 741, 742, 743
kopieren, CopyVar 655, 875
Liste 565, 569, 570, 572, 573
lokal, Local 647, 654, 656, 657, 658, 659, ... 919
löschen 752
löschen, DelVar . 257, 291, 656, 660, 887
Matrix 568, 569, 570, 572, 573
sperren, Lock 656
sperren/Sperrung aufheben 220
Sperrung aufheben, Unlock 656
statistisch 598, 603
Text 227
überschreiben 258
unbekannt, auswerten nach 746, 747
undefiniert 255, 256, 743
verschieben, MoveVar 656
verzögerte Vereinfachung 267
Variable kopieren, CopyVar 875
Variable sperren, Lock 919
Variable verschieben, MoveVar 924
Variablen 43, 772
aktivieren, Archive 868

archivieren, Archive . . 655, 785, 868
archivieren, aus Archiv holen . . . 784
aus Archiv holen, Unarchiv 785, 974
aus Archiv, Unarchiv 656
einfügen mit Namen 782, 783
in Anwendungen 782, 783
löschen 806
rückholen 21
speichern 21
System 511
übertragen 795, 798, 804
unabhängige 498, 502, 507
unarchive, Unarchiv 785
VARLINK . 767, 769, 770, 771, 778, 785
Verweis auf App-Dateien 33
Variablenstatus Locked/Archived (Gesperrt/Archiviert) 43
variance( ), Varianz 974
Varianz, variance( ) 974
Vektoren
Einheit, unitV( ) 974
Produkt, crossP( ) 878
Punktprodukt, dotP( ) 892
zylindrische Vektoranzeige, ▸Cylind . 882
Vektor-Format-Modus 22
verbinden und übertragen 697, 950
abbrechen 805
An Rechner senden, SendCalc . 809, 810
an Taschenrechner senden, SendCalc 698
CBL/CBR-Wert holen/ermitteln , Get 699
CBL/CBR-Wert holen/ermitteln, Get . 624, ... 904
Chat senden, SendChat . . 698, 809, 810
Fehler 806, 817, 818
Flash-Anwendungen . 798, 799, 802, 808, ... 809
Listenvariable senden, Send . . . 699, 949
Ordner 798, 799, 804, 806
Programm 809, 810
Rechner an Rechner . 795, 798, 802, 809, ... 810, 812, 813
Variablen 798, 799, 802, 804
zwischen Taschenrechnern 697
Verbindungs- und Übertragungstabelle . . 823
verborgene Fläche 429
verdeckte Oberfläche 416
Vereinfachung
Ende 267
Regeln 264
verzögert 267
verlagern, shift( ) 956
Verlauf
Status 43
Verzeichnis

einfügen mit Namen 782, 783
festlegen, setFold( ) . . 678, 775, 951
holen/ermitteln, getFold( ) 906
löschen 770
löschen, DelFold 656, 782, 887
neu, NewFold 656, 775, 926
VARLINK 770, 771, 778
Verzeichnisse 772
Vollbildmodus
2nd [QUIT] 21
Apps-Arbeitsfläche 41
Vorzeichen, sign( ) 957
Vorzeichentaste (-) 17

W

Wählen von Kategorien 36
Wahrscheinlichkeitsrechnung 859
WEB, Netz-Plots 385, 392, 393
Web-Plots
Konvergenz 395
Oszillation 398
Wert (Graph-Tool) 457
when( ), when 108, 975
when, when( ) 520, 975
While, while 674, 975
while, While 674, 975
willkürlich
Matrix, randMat( ) 830
number seed, RandSeed 830
Window Editor 63
Winkel, ∠ 991
Winkel, angle( ) 867
Winkelmodus 22
Status 43
Wissenschaft Kategorie 37
Wissenschaftliche Notation 18
Wissenschaftliches Zahlenformat . . . 166
with, | 77, 78, 269, 270, 994
Wochentag, dayOfWk( ) 885

X

x^{-1}, reziprok 994
xgrid (Fenstervariable) 410
x-Koordinate, P▶Rx( ) 932
xmax (Fenstervariable) . . 335, 370, 379, 388, ... 409, 453
xmin (Fenstervariable) . . . 335, 370, 379, 388, ... 409, 453
xor, (Boole) exklusives oder . . . 759, 976
xor, Boole ausschließendes oder . . . 665
XorPic, Bild mit ausschließendem oder . . 693
XorPic, Bild mit exklusives oder 976
xres (Fenstervariable) 335
xscl (Fenstervariable) 335, 370, 379, 388, ... 453
xyline-Plots 611

Y

Y= Editor . 326, 369, 376, 385, 408, 448, 514
umgehen 506

ygrid (Fenstervariable) 410
y-Koordinate, P▸Ry( ) 932
ymax (Fenstervariable) . . .335, 370, 379, 388, ... 409, 453
ymin (Fenstervariable) . . .335, 370, 379, 388, ... 409, 453
yscl (Fenstervariable) 335, 370, 379, 388, ... 453

Z

Zahlen
irrationale 259, 260
komplexe 506
Lösung, nSolve( ) 277
negative 165
numerische Ableitung, nDeriv( ) . 286
numerisches Integral, nInt( ) . . . 286
rationale 259, 260, 261
Zahlen ermitteln, getNum( ) 278
Zehnersystem
Anzeige von Ganzzahlen, ▸Dec . 754
Zeichen
Akzentzeichen 723, 727
Griechisch 11, 21
griechisch 725, 728, 730
Groß-/Kleinbuchstaben . . . 162, 716
international/Intonation 11, 21
Mathe 21
Mathematik 11
numerischer Ccode, ord( ) 662
Sonder- 11, 21
Sonderzeichen 165, 723, 726
String, char( ) 661, 871
Symbole 726
Zahlen-Code, ord( ) 931
Zeichen (Menü) 188
Zeichen-String, char( ) 661, 871
Zeichenzahl, dim( ) 661
Zeichnungen
auf Graph 693
Freihand 536
Funktion, DrawFunc . . 532, 696, 892
Horizontallinie, LineHorz . . . 696, 916
invers, DrawInv 533, 696, 892
Kontur, DrwCtour 696, 894
Kreis, Circle 695, 872
Kreise 539
Linie, Line 696, 916
Linien 538, 540
löschen 537
löschen, ClrDraw 694, 872
Neigung, DrawSlp 542, 696, 893
parametrisch 696
parametrisch, DrawParm . . 532, 893
Pencil 536
polar, DrawPol 532, 696, 893
Tangente, LineTan 696, 916
Vertikallinie, LineVert 696, 917
Zeilen in Staffeln anzeigen, ref( ) . . . 945
Zeilen reduzieren, rref( ) 948
Zeit
Einstellung 45

Rücksetzen 53
Zeitformat einstellen, setTmFmt( ) . . 953
Zeit-Plots, TIME 385, 392, 474, 476
Zeitzone abrufen, getTmZn( ) 908
Zeitzone einstellen, setTmZn( ) 954
Zero (math. Tool für Graphen) . 355, 357
zeros( ), Nullstellen . 259, 276, 283, 826, 976
Zertifikat . 806, 813, 814, 815, 816, 817, 818, ... 819
zmax (Fenstervariable) 409
zmin (Fenstervariable) 409
Zoom
Data, ZoomData 350
Daten, ZoomData 979
dezimal, ZoomDec 349, 979
Faktoren 350, 352
Ganzzahl, ZoomInt 349, 980
memory 350, 354
passend, ZoomFit 350, 980
Quadrat, ZoomSqr 349, 981
Rechteck, ZoomBox . . 349, 351, 978
speichern, ZoomSto 354, 982
Standard, ZoomStd 349, 982
Trig, ZoomTrig 349, 982
vergrößern, ZoomIn . . 349, 352, 980
verkleinern, ZoomOut . 349, 352, 981
vorher, ZoomPrev 354, 981
zurückholen, ZoomRcl 354, 981
Zoom (Menü) 348
Zoom Factors (Zoom) 352
ZoomBox, Zoom Rechteck 978
ZoomData, Zoom Daten 979
ZoomDec, Zoom dezimal 979
ZoomFit, Zoom passend 980
ZoomIn, vergrößern 980
ZoomInt, Zoom Ganzzahl 980
ZoomOut, verkleinern 981
ZoomPrev, Zoom vorher 981
ZoomRcl, Zoom zurückholen 981
ZoomSqr, Zoom Quadrat 981
ZoomStd, Zoom Standard 982
ZoomSto, Zoom speichern 982
ZoomTrig, Zoom Trig 982
Zubehöranschluss 66
Zufall
Matrix, randMat( ) 943
Norm, randNorm( ) 943
Polynom, randPoly( ) 943
Werkseinstellungen für Zufallszahlengenerator, RandSeed 943
Zufallszahl, rand( ) 943
zurückholen
aus Graphik-Einstellungen, RclGDB . 692, ... 943
Bild, RclPic 692, 944
Graphik-Einstellungen, RclGDB . 552
zweite Ableitung, ... 992
Zwischenablage 229, 230, 231, 720
Zyklus, Cycle 882
zylindrische Vektoranzeige, ▸Cylind . 882