

## First Steps

| $\square$ On/Off | $\square$ Menus |
| :--- | :--- |
| $\square$ Selecting applications | $\square$ Modes |
| $\square$ Graphing a function | $\square$ Entering numbers |

## Creating...

- Tables
$\square$ Matrices
- Statistic Data Plots
$\square$ Split Screens


## Beyond the Basics

$\square$ Activities
$\square$ Programming
$\square$ Archiving / Unarchiving
$\square$ Functions \& Instructions

## More Information

$\square$ Connectivity
$\square$ Text Editor
$\square$ Number Bases
$\square$ Support \& Service

## Important Information

Texas Instruments makes no warranty, either express or implied, including but not limited to any implied warranties of merchantability and fitness for a particular purpose, regarding any programs or book materials and makes such materials available solely on an "as-is" basis. In no event shall Texas Instruments be liable to anyone for special, collateral, incidental, or consequential damages in connection with or arising out of the purchase or use of these materials, and the sole and exclusive liability of Texas Instruments, regardless of the form of action, shall not exceed the purchase price of this product. Moreover, Texas Instruments shall not be liable for any claim of any kind whatsoever against the use of these materials by any other party.

## USA FCC Information Concerning Radio Frequency Interference

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation.

If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, you can try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/television technician for help.

Caution: Any changes or modifications to this equipment not expressly approved by Texas Instruments may void your authority to operate the equipment.

## Getting Started

## Initial start-up

## Installing the AAA Batteries

The TI-89 Titanium uses four AAA alkaline batteries and a backup silver oxide battery (SR44SW or 303). The backup batteries are already installed, and the AAA batteries are provided with the product.

1. Remove the battery cover from the back of the calculator.
2. Unwrap the four AAA batteries provided with your product and insert them in the battery compartment. Arrange the batteries according to the polarity (+ and -) diagram in the battery compartment.

3. Replace the battery cover on the calculator. The cover should snap into place.

## Turning on your Tl-89 Titanium for the first time

After installing the batteries included with the calculator, press ON. The Apps desktop appears.

Note: If your calculator initializes the preinstalled Apps, a progress bar will appear with the message "Installation in progress . . . Do not interrupt!" instead of the Apps desktop. To avoid losing Apps, do not remove the batteries during initialization. (You can re-install Apps from either the Product CD-ROM or education.ti.com.)

## Progress bar

## Adjusting the contrast

- To lighten the display, press and hold $\square$ and tap $\square$.
- To darken the display, press and hold $\square$ and tap +.



## The Apps desktop

The Apps desktop is the starting point for operating your TI-89 Titanium. Your installed Apps appear on the Apps desktop as icons organized in categories for easy access. From the Apps desktop, you can:

- Open Apps.
- Select and edit categories of Apps.
- View all of the Apps installed on your calculator.
- View the full name of the highlighted App.
- View and edit the time and date.
- Check status line information.
- View split-screen mode information.


TI-89 Titanium Apps desktop
(1) View full name of highlighted App.
(2) View time and date.
(3 Press ENTER to open highlighted App.
(4) Scroll down to view additional Apps.
© Check status line information.
(6) Edit categories.

To return to the Apps desktop at any time, press APPS. The last category selected appears with the last open App highlighted.

## Turning off the calculator

Press 2nd [0FF]. The next time you turn on the calculator, the Apps desktop appears with the same settings and memory contents retained. (If you turned off the Apps desktop, the calculator Home screen appears.)

You can use either of the following keys to turn off the TI-89 Titanium.

| Press: | Description |
| :---: | :---: |
| 2nd [0FF] (press 2nd and then press [0FF]) | Settings and memory contents are retained by the Constant Memory ${ }^{\text {TM }}$ feature. <br> - You cannot, however, use 2nd [0FF] if an error message is displayed. <br> - When you turn the TI-89 Titanium on again, it displays either the Home screen or the Apps desktop (regardless of the last application you used). |
| $\square$ [0FF] (press and then press [OFF]) | Similar to 2nd [0FF] except: <br> - You can use $\quad[0 F F]$ if an error message is displayed. <br> - When you turn the TI-89 Titanium on again, it will be exactly as you left it. |

Note: [ $0 F F]$ is the second function of the 0 ON key.
The calculator's Automatic Power Down ${ }^{\text {TM }}$ (APD ${ }^{\text {TM }}$ ) feature prolongs battery life by turning the calculator off automatically following several minutes of inactivity. When you turn on the calculator after APD:

- The display, cursor, and any error conditions are exactly the same as before APD.
- All settings and memory contents are retained.

Note: APD does not function when a calculation or program is in progress, unless a pause is specified in the calculation or program.

## TI-89 Titanium keys



## TI-89 Titanium keys

(1) Function keys (ㅍ1- F8) open toolbar menus, access Apps, and edit categories of Apps.
(2) Cursor keys $(\uparrow,(1), \odot, \odot)$ move the cursor.
(3) Numeric keypad performs math and scientific functions.
(4) Modifier keys ( $2 n \mathrm{nd}, \square, \pm$ ) add features by increasing the number of key commands.

## Entering special characters

Use the CHAR (Character) menu and key commands to enter special characters. The CHAR menu lets you access Greek, math, international, and other special characters. An on-screen keyboard map shows the locations of shortcuts used to enter other commonly used characters.

To select characters from the CHAR menu:

1. Press 2nd [CHAR]. The CHAR menu appears.
2. Use the cursor keys to select a category. A submenu lists the characters in that category.
3. Use the cursor keys to select a character, and press ENTER.

Example: Enter the right arrow symbol $(\rightarrow)$ in the Text Editor.


To open the keyboard map, press $\quad[\mathrm{KEY}]$. The keyboard map appears.

To type most characters, press 2nd and the corresponding key. Press ESC to close the map.

Example: Use the keyboard map to find the "not equal to" symbol ( $\neq$ ) shortcut and enter the symbol in the Program Editor.

| Press | Result |
| :---: | :---: |
| - [KEY] |  |
| $\square$ |  |

## Modifier keys

Modifier keys add features by increasing the number of keyboard operations at your fingertips. To access a modifier function, press a modifier key and then press the key for the corresponding operation.

| Keys | Description |
| :--- | :--- |
| 2nd | Accesses Apps, menu options, and other <br> operations. Second functions are printed above <br> their corresponding keys in the same color as <br> the 2nd key. |
| (Decond) | Accesses Apps, menu options, and other <br> operations. Diamond functions are printed <br> above their corresponding keys in the same <br> color as the |
| (Shift) | Typey. |
| key you press. Also character for the next letter and © to to |  |
| highlight characters when editing. |  |

Example: Access the VAR-LINK [All] screen, where you can manage variables and Apps.

## Press

Result
2nd [VAR-LINK]


## Function keys

Use the function keys to perform the following operations:

- On the Apps desktop, open Apps and select or edit Apps categories.
- On the calculator Home screen, open toolbar menus to select math-related operations.
- Within Apps, open toolbar menus to select App options.


## Numeric keypad

The numeric keypad lets you enter positive and negative numbers.
To enter a negative number, press $-(-)$ before typing the number.
Note: Don't confuse the negation key ( $(-)$ ) with the subtraction key ( $\square$ ).

To enter a number in scientific notation:

1. Type the numbers that precede the exponent. (This value can be an expression.)
2. Press 2nd 匡. The exponent symbol ( E ) follows the numbers you entered.
3. Type the exponent as an integer with up to three digits. (As the following example shows, you can use a negative exponent.)

Example: On the calculator Home screen, enter 0.00685 using scientific notation.

| Press | Result |
| :---: | :---: |
| $6 \square 85$ | ( |
| EE |  |
| (-) 3 | $\underbrace{6.85 \mathrm{E}^{-3}}_{\text {Kill }}$ |
| ENTER |  |
|  | $\frac{10685}{-05}$ $\frac{6.85 \mathrm{E}^{-3}}{\text { ल. }}$ |


| Key Command | Description |
| :---: | :---: |
| - [Y=] | Displays the Y= Editor. |
| $\square$ [WINDOW] | Displays the Window Editor. |
| - [GRAPH] | Displays the Graph screen. |
| - [TBLSET] | Sets parameters for the Table screen. |
| - [TABLE] | Displays the Table screen. |
| $\rightarrow$ [CUT] <br> - [COPY] <br> $\rightarrow$ [PASTE] | These keys let you edit entered information by performing a cut, copy, or paste operation. |
| APPS | Displays the Apps desktop. |
| $\square$ APPS | With the Apps desktop off, displays the FLASH APPLICATIONS menu. |
| 2nd [ $\boxplus$ ] | Switches between the last two chosen Apps. |
| 2nd [CUSTOM] | Turns the custom menu on and off. |
| 2nd [ $\bullet$ ] | Converts measurement units. |
| -[-] | Designates a measurement unit. |
| $\square$ | Deletes the character to the left of the cursor (backspace). |


| Key Command | Description |
| :---: | :---: |
| $\bullet$ - ${ }^{\text {- }}$ [L] | Deletes the character to the right of the cursor. |
| 2nd [INS] | Switches between insert and overwrite modes. |
| 2nd [MEM] | Displays the MEMORY screen. |
| CATALOG | Displays a list of commands. |
| 2nd [RCL] | Recalls the contents of a variable. |
| STO- | Stores a value to a variable. |
| 2nd [CHAR] | Displays the CHAR menu, which lets you select Greek letters, international accented characters, and other special characters.. |
| 2nd [QUIT] | - In full-screen mode, displays the Apps desktop. <br> - In split-screen mode, displays the full-screen view of the active App. <br> - With the Apps desktop off, displays the calculator Home screen. |

## Mode settings

Modes control how the TI-89 Titanium displays and interprets information. All numbers, including elements of matrices and lists, are displayed according to the current mode
settings. When the TI-89 Titanium is turned off, the Constant Memory ${ }^{\text {TM }}$ feature retains all of the mode settings you have selected.

To view the TI-89 Titanium mode settings:

1. Press MODE. Page 1 of the MODE dialog box appears.
2. Press F2 or F3 to display the modes listed on Page 2 or Page 3.

Note: Modes that are grayed out are available only if other required mode settings are selected. For example, the Custom Units mode listed on Page 3 is available only if the Unit System mode is set to CUSTOM.

## Viewing mode settings

| Press | Result |
| :---: | :---: |
| MODE |  |
| F2 |  |


| MODE |  |
| :---: | :---: |
| [Fage 1 Fage 2 Fage ${ }^{\text {F1 }}$ |  |
| - Unit System....... SIt <br> Angual |  |
|  |  |
| (Enter=SANE) | ESC=CA |

## Changing mode settings

Example: Change the Language mode setting to Spanish (Español).

| Press | Result |
| :---: | :---: |
| MODE |  |
| F3 |  |



To return the Language mode setting to English, repeat the steps, selecting 1:English in the Language field.

## Using the Catalog to access commands

Use the Catalog to access a list of TI-89 Titanium commands, including functions, instructions, and user-defined programs. Commands are listed alphabetically.
Commands not beginning with a letter are found at the end of the list (\&, $/,+,-$, etc.).
The Catalog Help App includes details about each command.
Options not currently valid are grayed out. For example, the Flash Apps (F36) menu option is grayed out if no Flash applications are installed on your $\mathrm{Tl}-89$ Titanium; the User-Defined (F4) menu option is grayed out if you have not created a function or program.

Note: Typing a letter takes you to the first command in the list starting with the same letter.

| Press | Result |
| :---: | :---: |
| CATALOG <br> (displays Built-in commands) |  |

## Result

F3
(displays Flash Apps
 commands, if any)

AHOUR $2 \omega 4$ - : - TI
6al (
birombide - - - TISt
Gingmpat =-:TIEt.Et
O.7i 22 w.

Gh 2 Cdf
-hi 20 BF
Ghi2Fdf
GlrListc
-:- -
-...-TI完

F4
(displays User-Defined commands, if any)


Select commands from the Catalog and insert them onto the calculator Home screen entry line or paste them to other Apps, such as the $\mathrm{Y}=$ Editor, Text Editor, or CellSheet ${ }^{\mathrm{TM}}$ Apps.

Example: Insert the comDenom( command on the calculator Home screen entry line.
Note: Before selecting a command, position the cursor where you want the command to appear.

Pressing 2nd $\Theta$ advances the Catalog list one page at a time.

| Press | Result |
| :---: | :---: |
| CATALOG alpha C <br> 2nd <br> Then press $\odot$ until the pointer is at the comDenom( function. |  |
| ENTER | (emer |

The status line displays any required and optional parameters for the selected command. Optional parameters appear in square brackets.

Note: Pressing F1 also displays the parameters for the selected command.


To exit the Catalog without selecting a command, press ESC.

## Calculator Home screen

The calculator Home screen is the starting point for math operations, including executing instructions, evaluating expressions, and viewing results.

To display the calculator Home screen, press $\quad$ HOME.
You can also display the calculator Home screen from the Apps desktop by highlighting the Home icon and pressing ENTER.

(1) History area lists the entry/answer pairs entered.
(2) Tabs display menus for selecting lists of operations. Press F1, F2, and so on to display menus.

3 Result of last entry is displayed here. (Note that results are not displayed on the entry line.)
(4) Status line shows the current state of the calculator.
© Entry line displays your current entry.
© Your last entry is displayed here.
To return to the Apps desktop from the calculator Home screen, press APPS.

## About the history area

The history area displays up to eight entry/answer pairs, depending on the complexity and height of the expressions. When the display is filled, information scrolls off the top of the screen. Use the history area to:

- Review previous entries and answers. Use the cursor keys to view entries and answers that have scrolled off the screen.
- Recall or auto-paste a previous entry or answer onto the entry line to reuse or edit. (For more information, see the electronic Operating the Calculator chapter.)

The cursor, which normally rests on the entry line, can be moved into the history area. The following table shows you how to move the cursor around in the history area.

| To | Do this |
| :---: | :---: |
| View entries/answers scrolled off the screen | From the entry line, press $\Theta$ to highlight the last answer. <br> Continue using $\Theta$ to move the cursor from answer to entry through the history area. |
| Go to the oldest or newest entry/answer pair | If the cursor is in the history area, press $\square$ $\odot$ or $\bullet \Theta$. |
| View an entry or answer too long for one line ( is displayed at the end of the line) | Move the cursor to the entry or answer. Use (1) or (1) to scroll left or right and 2nd (1) or 2nd (1) to go to the beginning or end. |
| Return cursor to the entry line | Press ESC, or press $\odot$ until the cursor is back on the entry line. |

## Interpreting history information on the status line

Use the history indicator on the status line for information about the entry/answer pairs. For example:

If the cursor is on the entry line:

| Total number of pairs |
| ---: |
| currently saved |$\quad 8 / 30 \longrightarrow$| Maximum number of |
| :--- |
| pairs that can be saved |

If the cursor is in the history area:

| Pair number of the |
| ---: |
| highlighted |
| entry/answer |$\quad 8 / 30 \longrightarrow$| Total number of pairs |
| :--- |
| currently saved |

## Modifying the history area

To change the number of pairs that can be saved:

1. From the calculator Home screen, press $F 1$ and select 9 :Format.
2. Press $(1)$ and use $\odot$ or $\odot$ to highlight the new number.
3. Press ENTER ENTER.

To clear the history area and delete all saved pairs:

- From the calculator Home screen, press F1 and select 8:Clear Home.
- or -
- Enter CIrHome on the calculator Home screen entry line.

To delete an entry/answer pair, move the cursor to either the entry or answer, and press $\square$ or CLEAR.

## Working with Apps

The TI-89 Titanium organizes Apps by category on the Apps desktop. To select a category, press a function key (F2] through [2nd [F8] ). The App icons for the selected category appear on the Apps desktop.

Note: If the name under an Apps desktop icon is truncated, use the cursor keys to highlight the icon. Now view the full name at the top of the Apps desktop.

## Opening Apps

Use the cursor keys to highlight the Apps icon on the Apps desktop and press ENTER. The App either opens directly or displays a dialog box. The most common dialog box lists these options for the App:

Note: The TI-89 Titanium uses the general term variable to refer to the App data files that you create.

| Option | Description |
| :--- | :--- |
| Current | Returns the screen displayed when you last viewed <br> the App. If no current App variable exists, the New <br> dialog box appears. |
| Open | Lets you open an existing file. |

New
Creates a new file with the name typed in the field.
Select an option, enter any required information, and press ENTER. The App appears.
Example: Create a new program using the Program Editor.

| Press | Result |
| :---: | :---: |
| Use cursor keys to highlight <br>  <br> Frosrom Ed... |  |
| ENTER |  |
| 3 | Frosrom Editor <br> $2:$ Gurrorert. <br> 2 inem... |
| ENTER |  |




ENTER ENTER


The newly created program variable, program1, is saved to the Main folder.

## Returning to the Apps desktop from within an App

Press APPS. The icons for the last Apps category selected appear on the Apps desktop with the icon for the last App opened highlighted.

You can also return to the Apps desktop by pressing 2nd [QUIT] in full-screen mode. In split-screen mode, press 2nd [QUIT] twice.

To return to the last open App from the Apps desktop, press $2 n d$ [ $\omega$ ].

## Selecting an Apps category

On the TI-89 Titanium, the Apps category names appear only in the F1 Menu. To select an Apps category, press F1 2:Select Category and use the cursor keys to highlight an Apps category, and then press ENTER to select the highlighted category. You can also use the function key shortcuts to select a category from the keypad (use the 2nd key if necessary). The App icons for the selected category appear on the Apps desktop.

The App icons for the selected category appear on the Apps desktop.

| Key | Description |
| :--- | :--- |
| F2 All | Icons for all installed Apps displayed. Not <br> customizable. |
| F63 English | Customizable category. English is the default. |
| F4] SocialSt | Customizable category. SocialSt (social studies) <br> is the default. |
| F5] Math | Customizable category. Math is the default. |
| 2nd [F6] Graphing | Customizable category. Graphing is the default. |
| 2nd [F7] Science | Customizable category. Science is the default. <br> 2nd [F8] Organizr |

Example：Select the All category．

| Press | Result |  |  |
| :---: | :---: | :---: | :---: |
| F2 | ${ }^{\text {Hewi }}$ | carri semestry | \％${ }^{\text {a }}$ |
|  | ant | \＆非圊 |  |
|  |  | arsemmi censket |  |
|  | Matif |  |  |
|  | Sirsing | $\mathrm{x}_{1}=$ erpm | A｜b 罣 |
|  |  |  |  |

If you select an Apps category containing no Apps，a message appears to confirm that the category is empty and point you to the F1 1：Edit Categories menu，where you can add App shortcuts to the category．（See＂Customizing the Apps categories＂on page 29．）

Press ENTER or ESC to clear the message and return to the Apps desktop．

## Customizing the Apps categories

The TI－89 Titanium organizes your Apps into seven categories，six of which you can customize to fit your individual needs．（The All category contains every installed App and cannot be edited．）

To customize the F3 through［2nd［F8］Apps categories：
1．Select F1 1：Edit Categories．A submenu displays the six customizable Apps category names．（The All category is not listed．）
2. Highlight an Apps category and press ENTER. The Edit Categories dialog box appears with a list of installed Apps and a text box with the category name highlighted.
3. To change the Apps category name, type the desired name.

Note: Enter a name of up to eight characters, including letters with or without capitalization, numbers, punctuation, and accented characters.
4. To add or remove an App shortcut from the category, press $\Theta$ as required to highlight the box next to the App, then press © to add or remove the check mark ( $\checkmark$ ).
5. To save the changes and return to the Apps desktop, press ENTER.

Example: Replace the Social Studies category with the Business category and add the CellSheet ${ }^{\text {TM }}$ and Finance App shortcuts.




## Open Apps and split-screen status

Your TI-89 Titanium lets you split the screen to view two Apps simultaneously. For example, view the $Y=$ Editor and Graph screens simultaneously to see the list of functions and how they are graphed.

Select the Split Screen mode from Page 2 of the MODE screen. The TI-89 Titanium displays the selected Apps in the split-screen view as shown. Split the screen horizontally (top-bottom) or vertically (left-right).

Top-bottom split screen


To return to the Apps desktop, press APPS. The split-screen status appears at the top of the Apps desktop with the names of the open Apps and the portions of the screen in which each is displayed. The highlighted numeral indicates the split-screen portion where the next App you open will appear.

Note: The Apps desktop always appears in the full-screen view.

Split-screen status (highlight indicates the portion where the next App selected will open.) Names of open Apps


More information is available about using split screens. (For more information, see the electronic Split Screens chapter.)

## Checking status information

Look to the status line, located at the bottom of the screen, for information about the current state of your TI-89 Titanium.


| Indicator | Meaning |
| :--- | :--- |
| $\boldsymbol{0}$ Entry/Answer pairs | 22/30-Number of entry/answer pairs (default is <br> 30, maximum is 99) in the history area of the <br> calculator Home screen. |
| 8 Replace batteries | Displayed when batteries are low (BATT). If <br> BATT is highlighted with a black background, <br> change the batteries as soon as possible <br> (EATT). |
| B Busy/Pause, | BUSY-Calculation or graph is in progress <br> Locked/Archived <br> variable |
| PAUSE-You paused a graph or program <br> B-Variable opened in the current editor is <br> locked or archived and cannot be modified |  |

## Turning off the Apps desktop

You can turn off the Apps desktop from the MODE dialog box. If you do, open Apps from the APPLICATIONS menu. To open the APPLICATIONS menu, press APPS.

Example: Turn off the Apps desktop.

| Press | Result |
| :---: | :---: |
| MODE |  |



To turn on the Apps desktop, repeat the procedure, selecting ON in the Apps Desktop mode field. To return to the Apps desktop from the calculator Home screen, press APPS.

## Using the clock

Use the CLOCK dialog box to set the time and date, select the clock display format, and turn the clock off and on.

The clock is turned on by default. If you turn off the clock, all Clock dialog box options except Clock ON/OFF are grayed out.


## Displaying the CLOCK dialog box

1. Use the cursor keys to highlight the Clock icon on the Apps desktop.
2. Press ENTER. The CLOCK dialog box appears with the Time Format field highlighted.

Note: Because the CLOCK dialog box displays the settings current at the time you open the dialog box, you might need to update the time before exiting.

## Setting the time

1. Press © to open the list of time formats.
2. Press $\Theta$ or $\Theta$ to highlight an option, then press ENTER. The selected format appears in the Time Format field.
3. Press $\Theta$ to highlight the Hour field.
4. Type the hour, then press $\Theta$ to highlight the Minute field.
5. Type the minute(s).
6. If the time format is 24 hours, proceed to step 9.

- or -

If the time format is 12 hours, press $\Theta$ to highlight the AM/PM field.
7. Press $(1)$ to open the list of $A M / P M$ options.
8. Press $\Theta$ or $\odot$ to highlight an AM/PM option, then press ENTER. The selected AM/PM option appears.
9. Set the date (for procedures, see Setting the date).

- or -

To save your settings and exit, press ENTER. The time is updated in the top right corner of the Apps desktop.

## Setting the date

1. Press $\Theta$ or $\Theta$ as required to highlight the Date Format field.
2. Press (1) to open the list of date formats.
3. Press $\odot$ or $\odot$ to highlight an option, then press ENTER. The selected format appears in the Date Format field.
4. Press $\Theta$ to highlight the Year field.
5. Type the year, then press $\odot$ to highlight the Month field.
6. Press (1) to open the list of months.
7. Press $\odot$ or $\Theta$ to highlight an option, then press ENTER. The selected month appears in the Month field.
8. Press $\Theta$ to highlight the Day field.
9. Type the day, then press ENTER ENTER to save your settings and exit. The date is updated in the top right corner of the Apps desktop.

Example: Set the time and date to 19/10/02 (October 19, 2002) at 1:30 p.m.

| Press | Result |
| :---: | :---: |
| Use cursor keys to highlight |  |
| ENTER |  |
| $\ominus 1 \ominus$ |  |


| Press | Result |
| :---: | :---: |
| $30 \ominus$ |  |
| (1) $\odot$ |  |
| ENTER $\odot$ |  |
| (1) $\odot$ |  |


| ENTER $\odot$ |  |
| :---: | :---: |
| 2002 |  |
| $\bigcirc(1)$ |  |
| Scroll down to October and press ENTER |  |


| Press | Result |
| :---: | :---: |
| $\bigcirc 19$ |  |
| ENTER ENTER |  |

## Turning off the clock

From the Apps desktop, open the CLOCK dialog box and select OFF in the Clock field.

Example: Turn off the clock.

| Press | Result |
| :---: | :---: |
| Use cursor keys to highlight |  |
| ENTER <br> Scroll down to the Clock field. |  |
| (1) ( $)$ ENTER |  |



To turn on the clock, repeat the procedure, selecting ON in the Clock field. Remember to reset the time and date.

## Using menus

To select most TI-89 Titanium menus, press the function keys corresponding to the toolbars at the top of the calculator Home screen and most App screens. Select other menus using key commands.

## Toolbar menus

The starting point for TI-89 Titanium math operations, the calculator Home screen displays toolbar menus that let you choose math-related options.

Toolbar menus also appear at the top of most App screens. These menus list common functions of the active App.

## Other menus

Use key commands to select the following menus. These menus contain the same options regardless of the screen displayed or the active App.

| Press | To display |
| :--- | :--- |
| 2nd [CHAR] | CHAR menu. Lists characters not available on the <br> keyboard; characters are organized by category <br> (Greek, math, punctuation, special, and <br> international). |
| 2nd [MATH] | MATH menu. Lists math operations by category. |
| APPS | APPLICATIONS menu. Lists the installed Apps. <br> (Menu is available only when the Apps desktop is <br> turned off; Apps are normally accessed from the <br> Apps desktop.) |
| APPS | FLASH APPLICATIONS menu. Lists the installed <br> Flash Apps. (Menu is available only when Apps <br> desktop is turned off; Flash Apps are normally <br> accessed from the Apps desktop.) |

## Selecting menu options

- Press the number or letter to the left of the option you want to select.
- Or -
- Press $\Theta$ or $\Theta$ to select the option, and press ENTER.

Note: If the first menu option is selected, press $\Theta$ to select the last option on the menu. If the last menu option is selected, press $\odot$ to select the first option on the menu.

Example: Select factor( from the Algebra menu on the calculator Home screen.

| Press | Result |
| :---: | :---: |
| Press: <br> HOME <br> - or - <br> From the Apps desktop, use the cursor keys to highlight <br> and press ENTER |  |
| F2 |  |
| $2$ <br> - Or - ENTER |  |

## Selecting submenu options

A small arrow symbol $(\stackrel{)}{ }$ to the right of a menu option indicates that selecting the option will open a submenu.

| Math |  |  |
| :---: | :---: | :---: |
| 1: dumber |  |  |
| 2:Angle | 1-sprg |  |
| Strit | 2:minc |  |
| 5:Complex | $4: 50 r^{-t}$ |  |
| G: Statistics | 5: Sortb |  |
| F: Frobability | G: sum |  |
| 8:Test. | 7: сыmsme |  |
| 9:Alpobrs | giproctuct |  |
| A: ${ }^{\text {Rejperbolic }}$ | A mide |  |
| EtSEring | Biright, | $\downarrow$ points to |

Example: Select ord( from the MATH menu on the calculator Home screen.

## Press Result

2nd [MATH]

| Finth |  |
| :---: | :---: |
| 1: + |  |
|  | - |
| S:List. |  |
| 4:1.引trix |  |
| 5: ¢0mFle | - |
| 6:'tati |  |
| 7: Frobeb |  |
| 8:TESt. |  |
|  | , |
| H:EEICH1 | - |
| B: Heper |  |
| L.t.rirug | - |


| Press | Result |
| :---: | :---: |
| C <br> - or - $\oplus \odot($ |  |
| B <br> - or - <br> $\Theta$ ENTER |  |

## Using dialog boxes

An ellipsis (...) at the end of a menu option indicates that choosing the option will open a dialog box. Select the option and press ENTER.


Example: Open the SAVE COPY AS dialog box from the Window Editor.

| Press | Result |
| :---: | :---: |
| APPS <br> Use the cursor keys to highlight |  |
| F1 |  |
| $\begin{gathered} \mathbf{2} \\ - \text { or - } \\ \odot \text { ENTER ESC } \end{gathered}$ |  |

Note: Pressing the $\square$ S key shortcut also opens the SAVE COPY AS dialog box in most Apps.

## Canceling a menu

To cancel a menu without making a selection, press ESC.

## Moving among toolbar menus

To move among the toolbar menus without selecting a menu option:

- Press the function key (F1 through (F8) of a toolbar menu.
- Press a function key, then press (1) or © $(1)$ to move from one toolbar menu to the next. Press (1) from the last menu to move to the first menu, and vice versa.

Note: If you press (1) when a menu option with a submenu is selected, the submenu will appear instead of the next toolbar menu. Press (1) again to move to the next menu.

More information is available about menus. (See the electronic Operating the Calculator chapter.)

## Custom menu

The custom menu provides quick access to your most commonly used options. Use the default custom menu or create your own using the Program Editor. You can include any available TI-89 Titanium command or character.

The custom menu replaces the standard toolbar menu on the calculator Home screen. (For details on creating a custom menu, see the electronic Programming chapter.) More information is available about custom menus. (See the electronic Operating the Calculator chapter.)

Example: Turn on and turn off the custom menu from the calculator Home screen.

| Press | Result |
| :---: | :---: |
| 2nd [CUSTOM] | Default custom menu |
|  |  |
| 2nd [CUSTOM] | Normal toolbar menu |
|  |  |

Example: Restore the default custom menu.

Note: Restoring the default custom menu erases the previous custom menu. If you created the previous custom menu with a program, you can run the program again to reuse the menu.

| Press | Result |
| :---: | :---: |
| 2nd [CUSTOM] <br> (to turn off the custom menu and turn on the standard toolbar menu) | (100 |
|  | $\xrightarrow{\text { Hinl }}$ |
| 2nd [F6] | Clear UF |
|  | 1:Clear <br> 2BNewrob <br> 3: Restore custom default |
| 3 <br> - or - <br> $\odot \odot$ ENTER | (120] |
|  |  |



## Opening Apps with the Apps desktop turned off

If you turn off the Apps desktop, use the APPLICATIONS menu to open Apps. To open the APPLICATIONS menu with the Apps desktop off, press APPS.

Note: If you press APPS with the Apps desktop turned on, the Apps desktop will appear instead of the APPLICATIONS menu.

Example: With the Apps desktop turned off, open the Window Editor from the APPLICATIONS menu.



To access Apps not listed on the APPLICATIONS menu, select 1:FlashApps.

## Using split screens

The TI-89 Titanium lets you split the screen to show two Apps at the same time. For example, display both the $\mathrm{Y}=$ Editor and Graph screens to compare the list of functions and how they are graphed.

## Setting split-screen mode

You can split the screen either top to bottom or left to right from the MODE dialog box. The split-screen setting stays in effect until you change it.

1. Press MODE to display the MODE dialog box.
2. Press F2 to display the Split Screen mode setting.
3. Press (1) to open the Split Screen mode menu.
4. Press $\ominus$ as required to highlight either TOP-BOTTOM or LEFT-RIGHT.
5. Press ENTER. The Split Screen mode setting displays the option you selected.

Example: Set split-screen mode to TOP-BOTTOM.

| Press | Result |
| :---: | :---: |
| MODE |  |
| F2 |  |
| (1) $\odot$ |  |
| ENTER |  |



## Setting the initial Apps for split screen

After you select either TOP-BOTTOM or LEFT-RIGHT split-screen mode, additional mode settings become available.

Full-screen mode


Split-screen mode

Mode Description

Split 2 App
Lets you specify the App displayed in the bottom or right portion of the split screen. Works together with Split 1 App, which lets you specify the App displayed in the top or left portion of the split screen.

Number of Graphs Lets you set up and display two independent graphs.
To set the initial App for each split-screen portion:

1. Select the Split 1 App mode setting and press ( $(\operatorname{t)}$ to display a menu of available Apps. (See "Setting split-screen mode" on page 54.)
2. Press $\Theta$ or $\Theta$ to highlight the App and press ENTER.
3. Repeat steps 1 and 2 for the Split 2 App mode setting.

Example: Display the $\mathrm{Y}=$ Editor in the top screen and the Graph App in the bottom screen.



If you set Split 1 App and Split 2 App to the same nongraphing App or to the same graphing App with Number of Graphs set to 1, the TI-89 Titanium exits split-screen mode and displays the App in full-screen mode.

## Selecting the active App

In split-screen mode, only one App can be active at a time.

- To switch between active Apps, press 2nd [ $\boxplus$ ].
- To open a third App, press APPS and select the App. This App replaces the active split-screen App.


## Exiting split-screen mode

Exit split-screen mode in any of the following ways:

- Press 2nd [quit] to close the active App and display the full-screen view of the other open App.
- If the Apps desktop is turned off, pressing [2nd [QUIT] replaces the active split-screen App with the calculator Home screen. Pressing [2nd [QUIT] again turns off the splitscreen mode and displays the calculator Home screen in full-screen mode.
- Select Split Screen on Page 2 of the MODE dialog box, set split-screen mode to FULL, and press ENTER.
- Press 2nd [QUIT] twice to display the Apps desktop

More information is available about using split screens. (See the electronic Split Screens chapter.)

## Managing Apps and operating system (OS) versions

Using the Tl-89 Titanium connectivity features, you can download Apps from:

- The TI Educational \& Productivity Solutions (E\&PS) Web site at: education.ti.com/latest
- The CD-ROM included with your TI-89 Titanium.
- A compatible graphing calculator.

Adding Apps to your TI-89 Titanium is like loading software on a computer. All you need is TI Connect ${ }^{\text {TM }}$ software and the USB computer cable that came with your $\mathrm{TI}-89$ Titanium.

For system requirements and instructions to link to compatible calculators and download TI Connect software, Apps, and OS versions, see the TI E\&PS Web site.

Before downloading Apps to your TI-89 Titanium, please read the license agreement on the CD-ROM or TI Web site.

## Finding the OS version and identification (ID) numbers

If you purchase software from the TI E\&PS Web site or call the customer support number, you will be asked to provide information about your TI-89 Titanium. You will find this information on the ABOUT screen.

To display the ABOUT screen, press F1 3:About from the Apps desktop. The ABOUT screen displays the following information about your TI-89 Titanium:

© OS version
(2) Hardware version

3 Unit ID (required to obtain certificates for installing purchased Apps). Similar to a serial number. Write this number down and keep it in a safe place in case the calculator is ever lost or stolen.
(4) Apps certificate revision number (Cert. Rev.)
© Product identifier (Product ID). Similar to a model number.
Note that your screen will be different than the one shown above.

## Deleting an Application

Deleting an application removes it from the $\mathrm{TI}-89$ Titanium and increases space for other applications. Before deleting an application, consider storing it on a computer for reinstallation later.

1. Quit the application.
2. Press 2nd [VAR-LINK] to display the VAR-LINK (AII) screen.
3. Press [2nd [F7] to display the list of installed applications.
4. Select the application you want to delete by pressing F4). (Press F4 again to deselect.)
5. Press F1 1:Delete. The VAR-LINK delete confirmation dialog box displays.
6. Press ENTER to delete the application.

Note: Only Flash Apps can be deleted.

## Connecting your TI-89 Titanium to other devices

The TI-89 Titanium includes both a mini-USB port and a standard I/O port. Ports are used to link two compatible graphing calculators or connect to a computer or peripheral device.

In addition, the teacher model of the TI-89 Titanium includes an accessory port. This port is used to output visual data so that a classroom can view the calculator's display on a video device or overhead screen.

To connect your calculator to a computer - Connect your TI-89 Titanium using the USB port and the included USB computer cable.

To connect your calculator to another calculator - Use the USB unit-to-unit cable or an I/O unit-to-unit cable to connect the TI-89 Titanium to a compatible graphing calculator or peripheral device, such as a TI-89 or TI-92 Plus graphing calculator or the CBL $2^{\text {TM }}$ and CBR ${ }^{\text {TM }}$ systems.

To show your calculator's display to the classroom - Use the accessory port to connect the TI-Presenter ${ }^{T M}$ video adapter to the teacher model of the TI-89 Titanium. The TI-Presenter video adapter provides a video interface between the calculator and video display or recording devices. Or use the accessory port to connect the TI ViewScreen ${ }^{\text {TM }}$ overhead panel to your calculator. The TI ViewScreen overhead panel enlarges and projects the display so an entire class can view it. For more information about the TI-Presenter video adapter and TI ViewScreen panel, see the TI E\&PS Web site at education.ti.com.


TI-89 Titanium ports


TI-89 Titanium ports (teacher model)

## Batteries

The TI-89 Titanium uses four AAA alkaline batteries and a backup silver oxide battery (SR44SW or 303). The backup battery is already installed, and the AAA batteries are provided with your product.

## Installing the AAA Batteries

1. Remove the battery cover from the back of the calculator.
2. Unwrap the four AAA batteries provided with your product and insert them in the battery compartment. Arrange the batteries according to the polarity (+ and -) diagram in the battery compartment.

3. Replace the battery cover on the calculator. The cover should snap into place.

## Replacing the AAA (alkaline) batteries

As the batteries lose power, the display begins to dim, especially during calculations. If you find yourself increasing the contrast frequently, replace the AAA alkaline batteries.

The status line also gives battery information.

| Indicator | Meaning |
| :--- | :--- |
| EATT | Batteries are low. |
| EATT | Replace batteries as soon as possible. |

Before replacing the batteries, turn off the TI-89 Titanium by pressing 2nd [0FF] to avoid losing information stored in memory. Do not remove both the back-up battery and the AAA alkaline batteries at the same time.

## Replacing the backup (silver oxide) battery

1. To replace the silver oxide backup battery, remove the battery cover and unscrew the tiny screw holding the BACK UP BATTERY cover in place.

2. Remove the old battery and install a new SR44SW or 303 battery, positive (+) side up. Replace the cover and the screw.

## Important OS download information

New batteries should be installed before beginning an OS download.
When in OS download mode, the APD ${ }^{\text {TM }}$ feature does not function. If you leave your calculator in download mode for an extended time before you actually start the download, your batteries may become depleted. You will then need to replace the depleted batteries with new batteries before downloading.

You can also transfer the OS to another TI-89 Titanium using a USB unit-to-unit cable . If you accidentally interrupt the transfer before it is complete, you will need to reinstall the OS via a computer. Again, remember to install new batteries before downloading.

Please contact Texas Instruments as described in Service \& Support Information, if you experience a problem.

## Battery Precautions

Take these precautions when replacing batteries:

- Do not leave batteries within the reach of children.
- Do not mix new and used batteries. Do not mix brands (or types within brands) of batteries.
- Do not mix rechargeable and non-rechargeable batteries.
- Install batteries according to polarity (+ and -) diagrams.
- Do not place non-rechargeable batteries in a battery recharger.
- Properly dispose of used batteries immediately.
- Do not incinerate or dismantle batteries.


## Previews

## Performing Computations

This section provides several examples for you to perform from the Calculator Home screen that demonstrate some of the computational features of the TI-89 Titanium. The history area in each screen was cleared by pressing F1 and selecting 8:Clear Home, before performing each example, to illustrate only the results of the example's keystrokes.

## Showing Computations

| Steps and keystrokes | Display |
| :---: | :---: |
| Compute $\boldsymbol{\operatorname { s i n }}(\pi / 4)$ and display the result in | Frict |
| symbolic and numeric format. To clear the history area of previous calculations, press | $\cdots \sin \left(\frac{\pi}{4}\right] \quad \frac{\sqrt{2}}{2}$ |
| and select 8:Clear Home. | $\cdots \sin \left(\frac{\pi}{4}\right) \quad .797107$ |
|  |  |

## Finding the Factorial of Numbers

| Steps and keystrokes | Display |
| :---: | :---: |
| Compute the factorial of several numbers to see how the TI-89 Titanium handles very large integers. To get the factorial operator (!), press 2nd [MATH], select 7:Probability, and then select 1:!. |  |

: 5 2nd [MATH] 71 ENTER 20 2nd [MATH] 71 EENTER 30 2nd [MATH] 71 ENTER

## Expanding Complex Numbers

| Steps and keystrokes | Display |
| :---: | :---: |
| Compute ( $3+5 i)^{3}$ to see how the TI-89 |  |
| Titanium handles computations involving complex numbers. |  |
|  |  |

## Finding Prime Factors

| Steps and keystrokes | Display |
| :---: | :---: |
| Compute the factors of the rational number |  |
| 2634492. You can enter "factor" on the entry line by typing FACTOR on the keyboard, or by pressing F2 and selecting 2 :factor(. |  |
| Press F2 22634492 [ ENTER |  |
| (Optional) Enter other numbers on your own. |  |

## Expanding Expressions

## Steps and keystrokes Display

Expand the expression $(x-5)^{3}$. You can enter "expand" on the entry line by typing EXPAND on the keyboard, or by pressing F2 and selecting 3 :expand.

## 

(Optional) Enter other expressions on your own.

## Reducing Expressions

## Steps and keystrokes

Display
Reduce the expression $\left(x^{2}-2 x-5\right) /(x-1)$ to its simplest form. You can enter "propFrac" on the entry line by typing PROPFRAC on the keyboard, or by pressing [F2 and selecting 7:propFrac(.


Press F2 7

$1 \square \square$ ENTER

## Factoring Polynomials

## Steps and keystrokes

Display
Factor the polynomial ( $x^{2}-5$ ) with respect to $x$. You can enter "factor" on the entry line by typing FACTOR on the keyboard or by pressing (F2) and selecting 2 :factor(.



## Solving Equations



## Solving Equations with a Domain Constraint

| Steps and keystrokes | Display |
| :---: | :---: |
| Solve the equation $x^{2}-2 x-6=2$ with respect to x where x is greater than zero. The "with" (I) operator provides domain constraint. | - |
|  |  |

## Finding the Derivative of Functions

| Steps and keystrokes | Display |
| :---: | :---: |
| Find the derivative of $(x-y)^{3} /(x+y)^{2}$ with respect to x . |  |
| This example illustrates using the calculus differentiation function and how the function is displayed in "pretty print" in the history area. |  |
|  <br>  |  |

## Finding the Integral of Functions

| Steps and keystrokes | Display |
| :---: | :---: |
| Find the integral of $\mathbf{x} * \boldsymbol{\operatorname { s i n }}(\mathbf{x})$ with respect to x . | Frisim |
| This example illustrates using the calculus integration function. | - $f(x=\operatorname{in}(x) d x$ $\begin{aligned} & \frac{f(x)}{f(x) \sin (x), x)}=\sin (x)-x \cdot \cos (x) \end{aligned}$ |
|  |  |

## Symbolic Manipulation

Solve the system of equations $2 x-3 y=4$ and $-x+7 y=-12$. Solve the first equation so that $x$ is expressed in terms of $y$. Substitute the expression for $x$ into the second
equation, and solve for the value of $y$. Then substitute the $y$ value back into the first equation to solve for the value of $x$.

| Steps and keystrokes | Display |
| :---: | :---: |
| 1. Display the Home screen and clear the entry line. Solve the equation $2 x-3 y=4$ for $x$. |  |
|  |  |
| F2 1 selects solve( from the Algebra | $\begin{aligned} & 4, x=\frac{3}{3 \cdot y+} \\ & x=\frac{3}{2} \end{aligned}$ |
| menu. You can also type solve( directly from the keyboard or select it from the |  |
| Catalog. |  |
| $\begin{aligned} & \text { [:- HOME CLEAR CLEAR E2 } 12 \times \square 3 \mathrm{Y} \\ & \square 4 \square \times \square \text { ENTER } \end{aligned}$ |  |

2. Begin to solve the equation $-x+7 y=-12$ for $y$, but do not press ENTER yet.

3. Use the "with" operator to substitute the expression for $x$ that was calculated from the first equation. This gives the value of y.

The "with" operator is displayed as | on
 the screen.

Use the auto-paste feature to highlight the last answer in the history area and paste it to the entry line.
[- $1 \odot$ ENTER ENTER
4. Highlight the equation for x in the history area.

5. Auto-paste the highlighted expression to the entry line. Then substitute the value of $y$ that was calculated from the second equation.
E ENTER $1 \odot$ ENTER ENTER


The solution is:
$x=-8 / 11$ and $y=-20 / 11$
This example is a demonstration of symbolic manipulation. A one-step function is available for solving systems of equations.

## Constants and Measurement Units

Using the equation $\mathrm{f}=\mathrm{m} * a$, calculate the force when $\mathrm{m}=5$ kilograms and $\mathrm{a}=20$ meters/second ${ }^{2}$. What is the force when $a=9.8$ meters $^{2} /$ second ${ }^{2}$. (This is the
acceleration due to gravity, which is a constant named
g). Convert the result from newtons to kilograms of force.

## Steps and keystrokes

Display

1. Display the MODE dialog box, Page 3. For Unit System mode, select SI for the metric system of measurements.
Results are displayed according to these
 default units.

## Press MODE F3 (1) 1 ENTER

2. Create an acceleration unit for meters/second ${ }^{2}$ named _ms2.
The UNITS dialog box lets you select units from an alphabetical list of categories. You can use 2nd $\Theta$ and 2nd $\Theta$ to scroll one page at a time through the categories.

If you use the UNITS dialog box to select a
 unit, the _ is entered automatically. Now, instead of re-entering _m/_s ${ }^{2}$ each time you need it, you can use _ms2. Also, you can now use the UNITS dialog box to select_ms2 from the Acceleration category.
[- 2nd [UNITS] $\odot$ (1) M ENTER $\div$ 2nd [UNITS] $\odot \odot \odot \odot(1)$ S ENTER $\triangle 2$ STO• [-] [2nd [a-lock] MS alpha 2 ENTER

## Display

3. Calculate the force when $\mathrm{m}=5$ kilograms (_kg) and $\mathrm{a}=20$ meters $/$ second $^{2}$ (_ms2).

| $\begin{aligned} & \frac{-m}{\mathrm{~s}^{2}}+-\mathrm{ms} 2 \\ & 5 \cdot-\mathrm{kg} \cdot 20 \cdot-\mathrm{ms} 2 \end{aligned}$ | $\begin{array}{r} \frac{-m}{-s^{2}} \\ 100 .-\mathrm{H} \end{array}$ |  |
| :---: | :---: | :---: |
|  |  |  |
| 5_kg*20_ris2 |  |  |
| Milk | FUNAC | 2130 |

If you know the abbreviation for a unit, you can type it from the keyboard.
( 5 [-] 2nd [a-lock] KG alpha $\otimes 20$ $\rightarrow$ [-] [2nd [a-lock] MS alpha 2 ENTER
4. Using the same $m$, calculate the force for an acceleration due to gravity (the constant _g).

For _g, you can use the pre-defined constant available from the UNITS dialog box or you can type _g.
© $5 \square$ [_] 2nd [a-lock] KG alpha 区 2nd [UNITS] (1) alpha G ENTER ENTER
5. Convert to kilograms of force (_kgf).

2nd [ $\downarrow$ ] displays the conversion operator.
( (1) 2nd [ $\downarrow$ ] [-] 2nd [a-lock] KGF alpha ENTER

learn how to enter a function, produce a graph of the function, trace a curve, find a minimum point, and transfer the minimum coordinates to the Home screen.

Explore the graphing capabilities of the TI-89 Titanium by graphing the function $y=\left(\left|x^{2}-3\right|-10\right) / 2$.

## Steps and keystrokes

1. Display the $\mathbf{Y}=$ Editor.

$$
\text { Press } \rightarrow[Y=]
$$

## Display


2. Enter the function $\left(\operatorname{abs}\left(x^{2}-3\right)-10\right) / 2$.

The screen shot shows the "pretty print" display at $\mathbf{y} \mathbf{1}=$.
© $\quad$ CATALOG A ENTER $\times$ ® $2 \square 3 \square$ $\square 10 \square \div 2$ ENTER
3. Display the graph of the function.

Select 6:ZoomStd by pressing 6 or by moving the cursor to 6:ZoomStd and pressing ENTER.

Press F2 6



## Display

4. Turn on Trace.

The tracing cursor, and the $x$ and $y$ coordinates are displayed.

Press F3

5. Open the MATH menu and select 3:Minimum.

Press $\operatorname{F5} \odot \odot$ ENTER

6. Set the lower bound.

Press ( (1) (right cursor) to move the tracing cursor until the lower bound for x is just to the left of the minimum node before pressing ENTER the second time.


Press (1) ... (1) ENTER
7. Set the upper bound.

Press ( (1) (right cursor) to move the tracing cursor until the upper bound for x is just to the right of the minimum node.

Press (1) ... (1)


## Display

8. Find the minimum point on the graph between the lower and upper bounds.

Press ENTER

9. Transfer the result to the Home screen, and then display the Home screen.
[- HOME

| FIT |  |  |  |
| :---: | :---: | :---: | :---: |
| $-\left[\begin{array}{ll}1.7320508075682 & -4.9995\end{array}\right.$ $[1.73205-5$. |  |  |  |
|  |  |  |  |
| Milk | Bind AUTO | Funt | $1 \times 3$ |

## Basic Function Graphing II

Graph a circle of radius 5 , centered on the origin of the coordinate system. View the circle using the standard viewing window (ZoomStd). Then use ZoomSqr to adjust the viewing window.

| Steps and keystrokes | Display |
| :---: | :---: |
| 1. Display the MODE dialog box. For Graph mode, select FUNCTION. <br> Press MODE <br> (1) 1 $\square$ |  |

## Display

2. Display the Home screen. Then store the radius, 5 , in variable $r$.
```
5->r
```

( HOME 5 STO』 alpha R ENTER
3. Display and clear the $Y=$ Editor. Then
define $\mathrm{y} 1(\mathrm{x})=\sqrt{\left(r^{2}-x^{2}\right)}$, the top half of a circle.

In function graphing, you must define separate functions for the top and bottom halves of a circle.
(-i) [ $\mathrm{Y}=]$ F1 8 ENTER ENTER 2nd [ v$]$ alpha R ง $2 \square \times \wedge 2 \square$ ENTER
4. Define $\mathrm{y} 2(\mathrm{x})=-\sqrt{r^{2}-x^{2}}$, the function for the bottom half of the circle.
The bottom half is the negative of the top half, so you can define $\mathrm{y} 2(\mathrm{x})=-\mathrm{y} 1(\mathrm{x})$.


Use the full function name $\mathbf{y} 1(\mathbf{x})$, not simply y1.
Press ENTER ( -1$)$ Y 1 X 1 ENTER

## Display

5. Select the ZoomStd viewing window, which automatically graphs the functions.

In the standard viewing window, both the $x$ and $y$ axes range from -10 to 10 .
However, this range is spread over a longer distance along the $x$ axis than the


Notice slight gap between top and bottom halves. $y$ axis. Therefore, the circle appears as an ellipse.

Press F2 6
6. Select ZoomSqr.

ZoomSqr increases the range along the $x$ axis so that circles and squares are shown in correct proportion.


Press F2 5

Note: There is a gap between the top and bottom halves of the circle because each half is a separate function. The mathematical endpoints of each half are $(-5,0)$ and $(5,0)$. Depending on the viewing window, however, the plotted endpoints for each half may be slightly different from their mathematical endpoints.

## Parametric Graphing

Graph the parametric equations describing the path of a ball kicked at an angle ( $\theta$ ) of $60^{\circ}$ with an initial velocity $\left(\mathrm{v}_{0}\right)$ of 15 meters $/ \mathrm{sec}$. The gravity constant $\mathrm{g}=9.8$ meters $/ \mathrm{sec}^{2}$.

Ignoring air resistance and other drag forces，what is the maximum height of the ball and when does it hit the ground？

## Steps and keystrokes <br> Display

1．Display the MODE dialog box．For Graph mode，select PARAMETRIC．

Press MODE（1） 2 ENTER


2．Display and clear the $\mathbf{Y}=$ Editor．Then define the horizontal component $\mathrm{xt} 1(\mathrm{t})=\mathrm{v}_{0} \mathrm{t} \cos \theta$ ．

```
xt1(t)=15t*\operatorname{cos}(6\mp@subsup{0}{}{\circ})
```

Enter values for $v_{0}$ and $\theta$ ．
［ei［ $\mathrm{Y}=]$ F1 8 ENTER ENTER 15T $ख$ 2nd［cos］ 60 2nd［ ${ }^{\circ}$ ］$\square$ ENTER

Type T 区 2nd［cos］，not T 2nd［cos］．
Enter a ${ }^{\circ}$ symbol by typing either［2nd［ ${ }^{\circ}$ ］ or［2nd［MATH］ 2 1．This ensures a number is interpreted as degrees，regardless of the angle mode．
3．Define the vertical component $\mathrm{yt} 1(\mathrm{t})=\mathrm{v}_{0} \mathrm{t} \sin \theta-(\mathrm{g} / 2) \mathrm{t}^{2}$ ．

Enter values for $\mathrm{v}_{0}, \theta$ ，and g ．
瞱 ENTER 15T 区 2nd［SIN］ 60 2nd［ ${ }^{\circ}$ ］$\square$



## Display

4. Display the Window Editor. Enter Window variables appropriate for
t.mirtor
tmax=3. this example.

You can press either $\Theta$ or ENTER to enter a value and move to the next variable.

$$
\begin{aligned}
& \text { Press } \oplus \text { [window] } 0 \odot 3 \odot .02 \odot \leftrightarrow 2 \\
& \ominus 25 \ominus 5 \ominus \Theta 2 \ominus 10 \ominus 5
\end{aligned}
$$

5. Graph the parametric equations to model the path of the ball.

Press [GRAPH]

6. Select Trace. Then move the cursor along the path to find the:

- y value at maximum height.
- $t$ value where the ball hits the ground.


Press (F3) (1) or © 1 as necessary

## Polar Graphing

The graph of the polar equation $\mathrm{r} 1(\theta)=\mathrm{A} \sin \mathrm{B} \theta$ forms the shape of a rose. Graph the rose for $A=8$ and $B=2.5$. Then explore the appearance of the rose for other values of $A$ and $B$.

| Steps and keystrokes | Display |
| :---: | :---: |
| 1. Display the MODE dialog box. For Graph mode, select POLAR. For Angle mode, select RADIAN. <br> Press MODE <br> (1) 3 <br> $\ominus \ominus \odot$ <br> 1 ENTER |  |

2. Display and clear the $\mathbf{Y}=$ Editor. Then define the polar equation $\mathrm{r} 1(\theta)=\mathrm{A} \sin \mathrm{B} \theta$.
Enter 8 and 2.5 for $A$ and $B$, respectively. - $\rightarrow$ [ $\mathrm{Y}=]$ F1 8 [ENTER ENTER 8 2nd [SIN] $2.5-[\theta] \square$ ENTER

3. Select the ZoomStd viewing window, which graphs the equation.

- The graph shows only five rose petals.

- In the standard viewing window, the Window variable $\theta$ max $=2 \pi$. The remaining petals have $\theta$ values greater than $2 \pi$.
- The rose does not appear symmetrical.
- Both the $x$ an $y$ axes range from -10 to 10. However, this range is spread over a longer distance along the $x$ axis than the $y$ axis.


## Press F2 6

4. Display the Window Editor, and change $\theta$ max to $4 \pi$.
$4 \pi$ will be evaluated to a number when you leave the Window Editor.

$\theta \mathrm{m} \times \mathrm{x}=4 \pi$
0st.eF= 1301996939957 xMir=-10. x $\mathrm{m} \times 1 \mathrm{~B}=1 \mathrm{~B}$ $\times \operatorname{xc} 1=1$
니․ $\times 1$ =首: $1=1$ 。 Press $\bullet[$ window $\odot 4$ [2nd $[\pi]$
5. Select ZoomSqr, which regraphs the equation.

ZoomSqr increases the range along the $x$ axis so that the graph is shown in
 correct proportion.

## Press F2 5

You can change values for $A$ and $B$ as necessary and regraph the equation.

## Sequence Graphing

A small forest contains 4000 trees. Each year, 20\% of the trees will be harvested (with $80 \%$ remaining) and 1000 new trees will be planted. Using a sequence, calculate the number of trees in the forest at the end of each year. Does it stabilize at a certain number?

| Initially | After 1 Year | After 2 Years | After 3 Years | $\cdots$ |
| :--- | :--- | :--- | :--- | :--- |
| 4000 | $.8 \times 4000$ | $.8 \times(.8 \times 4000+$ | $.8 \times(.8 \times(.8 \times$ | $\cdots$ |
|  | +1000 | $1000)+1000$ | $4000+1000)+$ |  |
|  |  |  | $1000)+1000$ |  |

1. Display the MODE dialog box. For Graph mode, select SEQUENCE.

Press MODE (1) 4 ENTER

2. Display and clear the $Y=$ Editor. Then define the sequence as
 $\mathrm{u} 1(\mathrm{n})=\operatorname{iPart}(.8 * u 1(\mathrm{n}-1)+1000)$.

Use iPart to take the integer part of the result. No fractional trees are harvested.

FLDTS ETant

$41=4000$

- $-12=$
$-12=$
- 

$14=$
4
-1 (n)=iF.artく, B+!

- Molk

To access iPart(, you can use 2nd [MATH], simply type it, or select it from the CATALOG.
> :- [Y=] [F1 8 [ENTER ENTER 2nd [MATH] 14.8 alpha U1 $\square$ alpha $\mathrm{NG} 1 \square \square$ $1000 \square$ ENTER
3. Define ui 1 as the initial value that will be used as the first term.

Press ENTER 4000 ENTER
4. Display the Window Editor. Set the n and plot Window variables.
nmin=0 and nmax=50 evaluate the size of the forest over 50 years.

Press $\oplus[$ window $0 \ominus 50 \ominus 1 \odot 1 \odot$
nmin= ทㅍ․ $=50$ Plotstrt=1. Plotster=1 xMir1= $\mathrm{xMax}=50$. $x=1=10$.
 Max=60ㅂ․ -
5. Set the $x$ and $y$ Window variables to appropriate values for this example.

Press $0 \ominus 50 \ominus 10 \odot 0 \ominus 6000 \ominus 1000$
6. Display the Graph screen.

Press [GRAPH]

7. Select Trace. Move the cursor to trace year by year. How many years (nc) does it take the number of trees (yc) to stabilize?

Trace begins at $\mathrm{nc}=0$. nc is the number of years.
$x c=n c$ since $n$ is plotted on the $x$ axis.
$\mathrm{yc}=\mathrm{u} 1(\mathrm{n})$, the number of trees at year $n$.


By default, sequences use the Square display style.

Press (F3) (1) and (1) as necessary

## 3D Graphing

Graph the 3D equation $z(x, y)=\left(x^{3} y-y^{3} x\right) / 390$. Animate the graph by using the cursor to interactively change the eye Window variable values that control your viewing angle. Then view the graph in different graph format styles.
Steps and keystrokes Display

1. Display the MODE dialog box. For Graph mode, select 3D.

Press MODE (1) 5 ENTER

2. Display and clear the $\mathbf{Y}=$ Editor. Then define the 3D equation $z 1(x, y)=\left(x^{3} y-y^{3} x\right) / 390$.

Notice that implied multiplication is used in the keystrokes.


Press $\bullet[\mathrm{Y}=]$ F1 8 ENTER ENTER $1 \mathrm{X} \wedge 3$
Y -Y 囚 $3 \times \square \div 390$ ENTER
3. Change the graph format to display and label the axes. Also set Style = WIRE FRAME.

You can animate any graph format style,
 but WIRE FRAME is fastest.
目 $\rightarrow$ ( $\odot 2 \odot(1) 2 \ominus$ (1) 1 ENTER
4. Select the ZoomStd viewing cube, which automatically graphs the equation.

As the equation is evaluated (before it is graphed), "evaluation percentages" are shown in the upper-left part of the screen.
$\overline{[F 1}]$


Press F2 6
Note: If you have already used 3D graphing, the graph may be shown in expanded view. When you animate the graph, the screen returns to normal view automatically. (Except for animation, you
 can do the same things in normal and expanded view.)

Press $\mathbb{\text { ® }}$ (press $\boxtimes$ to switch between expanded and normal view)
5. Animate the graph by decreasing the eyeф Window variable value.
$\Theta$ or $\Theta$ may affect eye $\theta$ and eye $\psi$, but to a lesser extent than eye $\phi$.

To animate the graph continuously, press
 and hold the cursor for about 1 second and then release it. To stop, press ENTER.

Press $\odot$ eight times

## Display

6. Return the graph to its initial orientation. Then move the viewing angle along the "viewing orbit" around the graph.


Press 0 (zero, not the letter O) (1) (1) (1)
7. View the graph along the $x$ axis, the $y$ axis, and then the $z$ axis.
Press X


This graph has the same shape along the $y$ axis and $x$ axis.

Press Y


Press Z

8. Return to the initial orientation.

Press 0 (zero)

Steps and keystrokes
9. Display the graph in different graph format styles.
[ $\square$ (press $\square$ to switch from each style to the next)

Display


HIDDEN SURFACE


CONTOUR LEVELS
(may require extra time to calculate contours)


WIRE AND CONTOUR


WIRE FRAME

Note: You can also display the graph as an implicit plot by using the GRAPH FORMATS dialog box ( $\square \square$ ). If you press: $\square$ to switch between styles, the implicit plot is not displayed.

## Differential Equation Graphing

Graph the solution to the logistic 1st-order differential equation y' = .001y*(100-y). Start by drawing only the slope field. Then enter initial conditions in the $\mathbf{Y =}$ Editor and interactively from the Graph screen.
Steps and keystrokes Display

1. Display the MODE dialog box. For Graph mode, select DIFF EQUATIONS.

Press MODE (1) 6 ENTER

2. Display and clear the $\mathbf{Y}=$ Editor. Then define the 1st-order differential equation:
$\mathrm{y} 1^{\prime}(\mathrm{t})=.001 \mathrm{y} 1 *(100-\mathrm{y} 1)$
Press $\boxtimes$ to enter the * shown above. Do not use implied multiplication between the
 variable and parentheses. If you do, it is treated as a function call.

Leave the initial condition yi1 blank.
Note: With y1' selected, the device will graph the y1 solution curve, not the derivative y 1 '.

Press $-[\mathrm{Y}=]$ F1 8 ENTER ENTER . 001 Y 1
® 100 Y Y1 1 ENTER
3. Display the GRAPH FORMATS dialog box. Then set Axes = ON, Labels = ON, Solution Method = RK, and Fields = SLPFLD.

Note: To graph one differential equation, Fields must be set to SLPFLD or FLDOFF.
 If Fields=DIRFLD, an error occurs when you graph.
 (1) 1 ENTER

## Display

4．Display the Window Editor，and set the Window variables as shown to the right．

| $\Theta 10 \ominus 110 \ominus 10 \ominus \Theta 10 \odot 120 \odot 10$$\Theta 0 \odot .001 \odot 20$ |
| :---: |
|  |  |
|  |  |

た回日，
tmax＝10．
$t=\mathrm{tep}=1$
t－plot＝0．
xMin＝－10．
$\mathrm{x} \dot{\mathrm{m}} \mathrm{x}=110$ ，
$x=c 1=10$.




fiftel＝201
5．Display the Graph screen．
Because you did not specify an initial condition，only the slope field is drawn（as specified by Fields＝SLPFLD in the GRAPH FORMATS dialog box）．


Press［GRAPH］
6．Return to the $\mathbf{Y}=$ Editor and enter an initial condition：
yi1 $=10$


7．Return to the Graph screen．
Initial conditions entered in the $\mathbf{Y}=$ Editor always occur at $\mathrm{t}_{0}$ ．The graph begins at the initial condition and plots to the right． Then it plots to the left．

Press［GRAPH］


The initial condition is marked with a circle．
8. Return to the $\mathbf{Y}=$ Editor and change yi1 to enter two initial conditions as a list:
yi1=\{10,20\}

```
*)
*FLDTS
t.0=0.
* 1'=.001-y1 (1000-91)
yi1=<10 20
```

Press $\bullet[\mathrm{Y}=] \oplus$ ENTER 2nd [ [ ] $10 \square 20$
2nd [ ] ] ENTER
9. Return to the Graph screen.

Press [GRAPH]

10. To select an initial condition interactively, press:
[ 2nd [F8]
When prompted, enter $\mathrm{t}=40$ and $\mathrm{y} 1=45$.


When selecting an initial condition interactively, you can specify a value for $t$ other than the $t_{0}$ value entered in the Y= Editor or Window Editor.

Instead of entering $\mathbf{t}$ and $\mathbf{y 1}$ after pressing

[2nd [F8]
you can move the cursor to a point on the screen and then press ENTER.

You can use F3 to trace curves for initial conditions specified in the $Y=$ Editor.
However, you cannot trace the curve for an initial condition selected interactively.
(2) 2nd [F8] 40 ENTER 45 ENTER

## Additional Graphing Topics

From the Home screen, graph the piecewise defined function: $y=-x$ when $x<0$ and $y=5 \cos (x)$ when $x \geq 0$. Draw a horizontal line across the top of the cosine curve. Then save a picture of the displayed graph.

| Steps and keystrokes | Display |
| :---: | :---: |
| 1. Display the MODE dialog box. For Graph mode, select FUNCTION. For Angle mode, select RADIAN. <br> Press MODE <br> (1) 1 <br> $\odot \ominus \ominus(1$ <br> 1 ENTER |  |

2. Display the Home screen. Use the Graph command and the when function to specify the piecewise defined function.
```
Graph when(x<0,-x,
5*\operatorname{cos(x))}
```

F4 2 selects Graph from the Other toolbar menu and automatically adds a space.
[ HOME FF4 2 2nd [a-lock] WHEN alpha
 [cos] $\times \square \square$
3. Execute the Graph command, which automatically displays the Graph screen.

The graph uses the current Window variables, which are assumed to be their
 standard values (F2 6) for this example.
Press ENTER

Display
4. Draw a horizontal line across the top of the cosine curve.

The calculator remains in "line" mode until you select a different operation or press
 ESC.
[-2nd [F7] $5 \odot$ (until the line is positioned) ENTER
5. Save a picture of the graph. Use PIC1 as the variable name for the picture.

Be sure to set Type = Picture. By default, it is set to GDB.

[- F1 $2 \oplus 2 \odot \odot$ PIC alpha 1 ENTER ENTER
6. Clear the drawn horizontal line.

You can also press F4 to regraph.
[2nd [F6] 1

7. Open the saved picture variable to redisplay the graph with the line.

Be sure to set Type = Picture. By default, it is set to GDB.

Press F1 1 (1) 2 (if not already shown, also set Variable = pic1) ENTER


## Tables

Evaluate the function $y=x^{3}-2 x$ at each integer between -10 and 10. How many sign changes are there, and where do they occur?

| Steps and keystrokes | Display |
| :---: | :---: |
| 1. Display the MODE dialog box. For the Graph mode, select FUNCTION. <br> Press MODE (1) 1 ENTER |  |

2. Display and clear the $Y=$ Editor. Then
define $y 1(x)=x^{3}-2 x$.
Press $\bullet[\mathrm{Y}=]$ F1 8 ENTER ENTER X 囚 $3 \square$ $2 \times$ ENTER

3. Set the table parameters to:
tbIStart $=-10$
$\Delta$ tbl $=1$
Graph $<->$ Table $=$ OFF Independent = AUTO


Press $\bullet[$ TBLSET $\Theta 10 \odot 1 \odot(1) 1 \odot(1) 1$ ENTER

## Display

4. Display the Table screen.

Press [TABLE]

5. Scroll through the table. Notice that $\mathbf{y} 1$ changes sign at $x=-1,1$, and 2 .

To scroll one page at a time, use 2nd $\Theta$ and $2 n d$.

Press $\Theta$ and $\Theta$ as necessary
6. Zoom in on the sign change between $x=-2$ and $x=-1$ by changing the table parameters to:
tblStart $=-2$
$\Delta$ tbl $=.1$


Press (F2) (-1) $2 \odot .1$ ENTER ENTER

## Split Screens

Split the screen to show the $\mathbf{Y}=$ Editor and the Graph screen. Then explore the behavior of a polynomial as its coefficients change.


Press MODE (1) 1 F2 (1) $3 \oplus(1) 2 \oplus$ (1) 4
ENTER
2. Clear the $\mathbf{Y}=$ Editor and turn off any stat data plots. Then define
$y 1(x)=.1 x^{3}-2 x+6$.
A thick border around the $\mathbf{Y}=$ Editor
 indicates it is active. When active, its entry line goes all the way across the display.
Press F1 8 ENTER F5 5 ENTER . $1 \times \Delta 3 \square$ $2 \times \square 6$ ENTER
3. Select the ZoomStd viewing window, which switches to the Graph screen and graphs the function.

The thick border is now around the Graph screen.

Fiz TET Fs


## Press F2 6

4. Switch to the $\mathbf{Y}=$ Editor. Then edit $\mathbf{y} \mathbf{1}(\mathbf{x})$ to change $.1 x^{3}$ to $.5 x^{3}$.

2nd $[\boxplus]$ is the second function of APPS.
The thick border is around the $Y=$ Editor.
Press 2nd [ $\rightarrow$ ] © $\odot$ ENTER (1) (1) (1) $\square 5$
ENTER
5. Switch to the Graph screen, which regraphs the edited function.

The thick border is around the Graph screen.


Press 2nd [ $\boxplus$ ]
6. Switch to the $\mathbf{Y}=$ Editor. Then open the Window Editor in its place.

Press 2nd [ $\boxplus$ ] [WINDOW]


## 7. Open the Home screen. Then exit to a

 full-sized Home screen.Press 2nd [QuIT] twice.

## Data/Matrix Editor

Use the Data/Matrix Editor to create a one-column list variable. Then add a second column of information. Notice that the list variable (which can have only one column) is automatically converted into a data variable (which can have multiple columns).

## Steps and keystrokes <br> Display

1. Use APPS to display the Data/Matrix Editor. Create a new list variable named TEMP.

Press $3 \oplus 3 \odot \odot$ TEMP ENTER ENTER

2. Enter a column of numbers. Then move the cursor up one cell (just to see that a highlighted cell's value is shown on the entry line).

LIST is shown in the upper-left corner to
 indicate a list variable.

You can use $\Theta$ instead of ENTER to enter information in a cell.

Press 1 ENTER 2 ENTER 3 ENTER 4 ENTER 5 ENTER 6 ENTER $\Theta$
3. Move to column 2, and define its column header so that it is twice the value of column 1.

DATA is shown in the upper-left corner to indicate that the list variable was converted to a data variable.
( (1) F4 2 区 alpha C 1 [ENTER

I. means the cell is in a defined column.
4. Move to the column 2 header cell to show its definition in the entry line.

When the cursor is on the header cell, you do not need to press [F4] to define it. Simply begin typing the expression.


Press 2 nd $\Theta \odot$
5. Clear the contents of the variable.

Simply clearing the data does not convert the data variable back into a list variable.

Press F1 8 ENTER

Note: If you don't need to save the current variable, use it as a scratchpad. The next time you need a variable for temporary data, clear the current variable and re-use it. This lets you enter temporary data without creating a new variable each time, which uses up memory.

## Statistics and Data Plots

Based on a sample of seven cities, enter data that relates population to the number of buildings with more than 12 stories. Using Median-Median and linear regression calculations, find and plot equations to fit the data. For each regression equation, predict how many buildings of more than 12 stories you would expect in a city of 300,000 people.


## Display

2. Use APPS to display the Data/Matrix Editor. Create a new data variable named BUILD.

Press $3 \ominus \ominus$ BUILD ENTER ENTER

3. Using the sample data below, enter the population in column 1.

| Pop. (in 1000s) | Bldgs $>12$ stories |
| :---: | :---: |
| 150 | 4 |
| 500 | 31 |
| 800 | 42 |
| 250 | 9 |
| 500 | 20 |
| 750 | 55 |
| 950 | 73 |



Press 150 ENTER 500 ENTER 800 ENTER 250 ENTER 500 ENTER 750 ENTER 950 ENTER

## Display

4. Move the cursor to row 1 in column 2 (r1c2). Then enter the corresponding number of buildings.
$\square$ moves the cursor to the top of the page. After typing data for a cell, you can
 press ENTER or $\odot$ to enter the data and move the cursor down one cell. Pressing $\Theta$ enters the data and moves the cursor up one cell.
圈
(1) $\bullet \odot 4$ ENTER 31 ENTER 42 ENTER
9 ENTER 20 ENTER 55 ENTER 73 ENTER
5. Move the cursor to row 1 in column 1 (r1c1). Sort the data in ascending order of population.

This sorts column 1 and then adjusts all other columns so that they retain the same order as column 1. This is critical for maintaining the relationships between columns of data.

To sort column 1, the cursor can be anywhere in column 1. This example has
 you press

so that you can see the first four rows.

6. Display the Calculate dialog box. Set Calculation Type = MedMed $\mathrm{x}=\mathrm{C} 1$
$\mathrm{y}=\mathrm{C} 2$
Store RegEQ to $=\mathbf{y 1}(\mathbf{x})$

[
F5 ( (1) $7 \odot C$ alpha $1 \odot$ alpha C2
(1) $\odot$ ENTER
7. Perform the calculation to display the MedMed regression equation.

As specified on the Calculate dialog box, this equation is stored in $\mathbf{y 1}(\mathbf{x})$.

Press ENTER

8. Close the STAT VARS screen. The Data/Matrix Editor displays.

Press ENTER
9. Display the Calculate dialog box. Set:

Calculation Type $=$ LinReg
$\mathrm{x}=\mathrm{C} 1$
$\mathrm{y}=\mathrm{C} 2$
Store RegEQ to $=\mathbf{y 2}(\mathbf{x})$


Press $\operatorname{F5}(1) 5 \odot \odot \odot \odot \odot$ ENTER
10. Perform the calculation to display the LinReg regression equation.

This equation is stored in $\mathbf{y 2}(\mathbf{x})$.
Press ENTER

| STAT YAFS |  |
| :---: | :---: |
| $y=a \cdot x+b$ |  |
| d |  |
| $\square$ | =-12.012431 |
| Edr | =.950317 |
| Fiz | =. 916457 |
| Entsr=近 |  |

11. Close the STAT VARS screen. The Data/Matrix Editor displays.

Press ENTER
12. Display the Plot Setup screen.

Plot 1 is highlighted by default.
F3 lets you clear highlighted Plot settings.
Press F2
13. Define Plot 1 as:

Plot Type = Scatter
Mark = Box
$\mathrm{x}=\mathrm{C} 1$
$\mathrm{y}=\mathrm{C} 2$


Notice the similarities between this and the Calculate dialog box.
 C2
14. Save the plot definition and return to the Plot Setup screen.

Notice the shorthand notation for Plot 1's definition.

Press ENTER twice
15. Display the $\mathbf{Y}=$ Editor. For $\mathbf{y 1}(\mathbf{x})$, the MedMed regression equation, set the display style to Dot.

Note: Depending on the previous contents of your $\mathrm{Y}=$ Editor, you may need to move
 the cursor to $\mathbf{y 1}$.

PLOTS 1 at the top of the screen means that Plot 1 is selected.

Notice that $\mathbf{y 1}(\mathbf{x})$ and $\mathbf{y 2}(\mathbf{x})$ were selected when the regression equations were stored.

芭 [Y=] [2nd [F6] 2
16. Scroll up to highlight Plot 1.

The displayed shorthand definition is the same as on the Plot Setup screen.

Press $\Theta$

17. Use ZoomData to graph Plot 1 and the regression equations $\mathbf{y 1}(\mathbf{x})$ and $\mathbf{y 2} \mathbf{2 ( x )}$.

ZoomData examines the data for all selected stat plots and adjusts the viewing
 window to include all points.

Press F2 9
18. Return to the current session of the Data/Matrix Editor.

Press 2nd [ $\boxplus$ ]
19. Enter a title for column 3. Define column 3's header as the values predicted by the MedMed line.

To enter a title, the cursor must highlight the title cell at the very top of the column.

F4 lets you define a header from anywhere in a column. When the cursor is on a header cell, pressing F4 is not required.
© © (1) (1) © $\odot$ 2nd [a-lock] MED alpha ENTER F4 Y1 1 alpha C1 1 ENTER
20. Enter a title for column 4. Define column 4's header as the residuals (difference between observed and predicted values) for MedMed.
© © © © 2nd [a-lock] RESID alpha ENTER
 alpha C2 alpha C3 ENTER
21. Enter a title for column 5. Define column 5's header as the values predicted by the LinReg line.
© $(1) \odot \odot$ 2nd [a-lock] LIN alpha ENTER F4 Y2 1 alpha C1 1 ENTER
22. Enter a title for column 6. Define column 6's header as the residuals for LinReg.

园
(1) © [2nd [a-lock] RESID alpha ENTER F4 alpha C2 -1 alpha C5 ENTER

|  |  |  |  |
| :---: | :---: | :---: | :---: |
| ${ }^{\text {DiSti }}$ | resid | lin | resid |
|  | C. 4 | -5 |  |
| 4 | -66667 | . 22169 | 3.7783 |
|  | -1.889 | 8.3778 | . 622224 |
|  | 1.2222 | 28.768 | 2.232 |
|  | -9.778 | 28.768 | -8.768 |
| $0 \cdot 6=c \cdot 2-5$ |  |  |  |
|  |  |  |  |

23. Display the Plot Setup screen and deselect Plot 1.

Press F2 F4
24. Highlight Plot 2 and define it as:

Plot Type = Scatter
Mark = Box
$\mathrm{x}=\mathrm{C} 1$
$\mathbf{y}=\mathbf{C 4}$ (MedMed residuals)

: $\odot$ F1 $\odot \odot$ C alpha $1 \odot$ alpha C 4 ENTER ENTER
25. Highlight Plot 3 and define it as:

Plot Type = Scatter
Mark = Plus
$\mathrm{x}=\mathrm{C} 1$
$\mathbf{y}=\mathbf{C 6}$ (LinReg residuals)

© $\odot$ F1 $\odot(1) 3 \odot C$ alpha $1 \odot$ alpha C6 ENTER ENTER
26. Display the $\mathbf{Y}=$ Editor and turn all the $\mathbf{y}(\mathbf{x})$ functions off.

From F5, select 3:Functions Off, not
1:All Off.
Plots 2 and 3 are still selected.


Press [Y=] F5 3
27. Use ZoomData to graph the residuals.marks the MedMed residuals;

+ marks the LinReg residuals.
Press F2 9


28. Display the Home screen.
[ HOME
29. Use the MedMed ( $\mathrm{y} 1(\mathrm{x})$ ) and LinReg (y2(x)) regression equations to calculate values for $x=300(300,000$ population).
The round function (2nd [MATH] 13) ensures that results show an integer number of buildings.

After calculating the first result, edit the entry line to change y1 to y2.
Press [2nd [MATH] 13 Y1 $0300 \square 0 \square$ ENTER (1) (1) (eight times) $\square 2$ ENTER

## Programming

Write a program that prompts the user to enter an integer, sums all integers from 1 to the entered integer, and displays the result.
Steps and keystrokes Display

1. Use APPS to display the Program Editor. Create a new program.

Press 3

|  | Frosrim Editor |
| :---: | :---: |
|  | 1: Eurrert. |
|  | 2\%pern... |

2. Type PROG1 (with no spaces) as the name of the new program variable.
琎 $\ominus \ominus$ PROG alpha 1

3. Display the "template" for a new program. The program name, Prgm, and EndPrgm are shown automatically.
After typing in an input box such as Variable, you must press ENTER twice.


Press ENTER twice

## Display

4．Type the following program lines．
Request＂Enter an integer＂，n
Displays a dialog box that prompts
＂Enter an integer＂，waits for the user to enter a value，and stores it（as a

Progl（）
：Pr＂ヨM
－Realdest＂Enter＂ari integer＂ ＂， H
 －首terpr
Fario 1 ，ri， 1
－temp＋i $\rightarrow$ temp
－EradF $\mathrm{or}^{\circ}$
－DisF t巨円F
EradProm
string）to variable n ．

## $\operatorname{expr}(\mathrm{n}) \rightarrow_{\mathrm{n}}$

Converts the string to a numeric expression．

Creates a variable named temp and initializes it to 0 ．

```
For i,1,n,1
```

Starts a For loop based on variable i．
First time through the loop，$i=1$ ．At end of loop，$i$ is incremented by 1.
Loop continues until $\mathrm{i}>\mathrm{n}$ ．
temp $+i \rightarrow$ temp
Adds current value of $i$ to temp．

## EndFor

Marks the end of the For loop．

## Disp temp

Displays the final value of temp．
Type the program lines as shown．
Press ENTER at the end of each line．
5. Go to the Home screen. Enter the program name, followed by a set of parentheses.

```
prog1()
```

You must include ( ) even when there are no arguments for the program.

The program displays a dialog box with the prompt specified in the program.
[ HOME 2nd [a-lock] PROG alpha 1 [ ENTER
6. Type 5 in the displayed dialog box.

Press 5

7. Continue with the program. The Disp command displays the result on the Program I/O screen.

The result is the sum of the integers from 1 through 5.

Although the Program I/O screen looks similar to the Home screen, it is for program input and output only. You cannot perform calculations on the Program I/O screen.
Press ENTER twice
8. Leave the Program I/O screen and return to the Home screen.

You can also press [ESC, 2nd [QUIT], or [- HOME
to return to the Home screen.


## Press $F 5$

## Text Operations

Start a new Text Editor session. Then practice using the Text Editor by typing whatever text you want. As you type, practice moving the text cursor and correcting any typos you may enter.
Steps and keystrokes Display

1. Start a new session of the Text Editor.

Press 3

|  | Frosiom Editor |
| :---: | :---: |
|  | 1: Eurrert |
|  | 2:0pron.. |

2. Create a text variable called TEST, which will automatically store any text you enter in the new session.

| HEW |  |
| :---: | :---: |
| TrFe: Text |  |
| Folder: main ${ }^{\text {a }}$ |  |
| Wariable test |  |
| Entrr= | ESC=CANCEL |

Use the MAIN folder, shown as the default on the NEW dialog box.

After typing in an input box such as Variable, you must press ENTER twice.

Press $\odot$ TEST ENTER ENTER
3. Type some sample text.

- To type a single uppercase letter, press 1 and then the letter.
- To type a space, press alpha [_] (alpha function of the $(--)$ key).

- To type a period, press alpha to turn alpha-lock off, press $\square$, and then press 2nd [a-lock] to turn alpha-lock on again.
Practice editing your text by using:
- The cursor pad to move the text cursor.
- $\quad$ or $\square$ [DEL] to delete the character to the left or right of the cursor, respectively.
[2. 2nd [a-lock] type anything you want

4. Leave the Text Editor and display the Home screen.

Your text session was stored automatically as you typed. Therefore, you do not need to save the session manually before exiting the Text Editor.
(HOME
5. Return to the current session on the Text Editor. Notice that the displayed session is exactly the same as you left it.
Press 2nd [ $\boxplus$ ]

## Numeric Solver

Consider the equation $\mathrm{a}=(\mathrm{m} 2-\mathrm{m} 1) /(\mathrm{m} 2+\mathrm{m} 1) * \mathrm{~g}$, where the known values are $\mathrm{m} 2=10$ and $\mathrm{g}=9.8$. If you assume that $\mathrm{a}=1 / 3 \mathrm{~g}$, find the value of m 1 .

| Steps and keystrokes | Display |
| :---: | :---: |
| 1. Use APPS to display the Numeric Solver. |  |

2. Enter the equation.

When you press ENTER or $\Theta$, the screen lists the variables used in the equation.

瞱

```
alpha A # O alpha M2 - alpha M1
```



```
alpha G ENTER
```

3. Enter values for each variable, except the unknown variable m1.

Define m 2 and g first. Then define a . (You must define $g$ before you can define a in terms of g.) Accept the default for bound. If a variable has been defined previously, its value is shown as a default.

4. Move the cursor to the unknown variable m1.

Optionally, you can enter an initial guess for m1. Even if you enter a value for all variables, the Numeric Solver solves for the variable marked by the cursor.

Press $\odot \odot$

Fit Enter* Equation ean : $\quad=(m 2-m 1)(m 2+m 1) * g$


```
g=(m2-m1)/(m2+m1)*g
    g=9/3
    m2=10.
    m1=
    M1=
    bound={-1,E14,1,E14}
```



``` \(a=(m 2-m 1) /(m 2+m 1) * g\) コ=3. 266666666667
    m2=10.
    m1=
    g=9.8
    bound={-1.E14,1.E14}
```

$\mathrm{g} / 3$ is evaluated when you move the cursor off the line.

## Display

5. Solve for the unknown variable.

To check the solution's accuracy, the left and right sides of the equation are evaluated separately. The difference is shown as left-rt. If the solution is precise, left-rt=0.

Press F2

```
FF1T
G=(m2-m1)\(m2+m1):+g
\Xi=3: 26666666666%
M2=1曰
m1=5.
    g=9.8
```



```
*left-r*t=0
```

marks the calculated values.
6. Graph the solution using a ZoomStd viewing window.

The graph is displayed in a split screen. You can explore the graph by tracing, zooming, etc.


The variable marked by the cursor (unknown variable $m 1$ ) is on the $x$ axis, and left-rt is on the $y$ axis.

Press F3 3
7. Return to the Numeric Solver and exit the split screen.

You can press ENTER or $\odot$ to redisplay the list of variables.

Press 2nd [円] F3 2

## Number Bases

Calculate 10 binary (base 2) + F hexadecimal (base 16) + 10 decimal (base 10). Then, use the operator to convert an integer from one base to another. Finally, see how changing the Base mode affects the displayed results.

## Steps and keystrokes <br> Display

1. Display the MODE dialog box, Page 2. For Base mode, select DEC as the default number base.

Integer results are displayed according to the Base mode. Fractional and floating-
 point results are always displayed in decimal form.

Press MODE F2 (use $\odot$ to move to Base mode) (1) 1 ENTER
2. Calculate $0 \mathrm{~b} 10+0 \mathrm{hF}+10$.

To enter a binary or hex number, you

| - bulo + OhF + 10 |  |  | 27 |
| :---: | :---: | :---: | :---: |
|  |  |  |  |
| Mill\| | Rind illo | FUNIC | 1/30 | must use the Ob or Oh prefix (zero and the letter B or H). Otherwise, the entry is treated as a decimal number.

Note: The 0b or Oh prefix is a zero, not the letter O, followed by B or H.
[- 0 alpha B 10 - 0 2nd [a-lock] HF alpha + 10 ENTER
3. Add 1 to the result and convert it to binary.
[2nd [ $\bullet$ ] displays the conversion operator.
 ENTER
4. Add 1 to the result and convert it to hexadecimal.

园 1 2nd [ $\bullet$ ] 2nd [a-lock] HEX alpha ENTER
5. Add 1 to the result and leave it in the default decimal base.

Results use the Ob or Oh prefix to identify the base.


Press $\dagger 1$ ENTER
6. Change the Base mode to HEX.

When Base $=$ HEX or BIN, the magnitude of a result is restricted to certain size limitations.

Press MODE F2 (use $\Theta$ to move to Base mode) (1) 2 ENTER
7. Calculate $0 \mathrm{~b} 10+0 \mathrm{hF}+10$.
[-0 alpha B 10 - 0 2nd [a-lock] HF alpha


+ 10 ENTER

8. Change the Base mode to BIN.

Press MODE F2 (use $\odot$ to move to Base mode) (1) 3 ENTER
9. Re-enter 0b10+0hF+10.

Press ENTER

```
- Ob1日 + OhF + 10
ᄇํา1B
Ob10 + 0hF + 10 回 11011
```



```
MAlk rablol
Fin\ illTD FUNT
```


## Memory and Variable Management

Assign values to a variety of variable data types. Use the VAR-LINK screen to view a list of the defined variables. Then move a variable to the user data archive memory and explore the ways in which you can and cannot access an archived variable. (Archived
variables are locked automatically.) Finally, unarchive the variable and delete the unused variables so that they will not take up memory.

| Steps and keystrokes |
| :---: |
| 1. From the Home screen, assign variables with the following variable types. |
| Expression: $5 \rightarrow x 1$ |
| Function: $\quad x^{2}+4 \rightarrow f(x)$ |
| List: $\quad\{5,10\} \rightarrow 11$ |
| Matrix: $\quad[30,25] \rightarrow \mathrm{m} 1$ |
| - HOME CLEAR 5 ST0^ X1 ENTER X ${ }_{\text {¢ }}$ |
|  |
| [1] 5 - 10 2nd [ ] ] ST0^ alpha L1 |
| ENTER 2nd [c] $30 \square 25$ 2nd []] STO* |
| alpha M1 ENTER |

2. Suppose you start to perform an operation using a function variable but can't remember its name.
```
5*
```

Display

|  |  |  |
| :---: | :---: | :---: |
| - $5 \cdot \times 1$ |  | 5 |
| - $x^{2}+4 \rightarrow f(x)$ |  | Done |
|  |  | 103 |
|  | [30] | 25] |
| [30, 25] ${ }^{\text {a }}$, 1 |  |  |
| MAIN | FUIN: | 4,30 |

## Display

4. Change the screen's view to show only function variables.

Although this may not seem particularly
 useful in an example with four variables, consider how useful it could be if there were many variables of all different types.
Press ©2 $\odot \odot(1) 5$ ENTER

5. Highlight the function variable, and view its contents.

Notice that the function was assigned using $f(x)$ but is listed as $f$ on the screen.

[-0 $\Theta$ 2nd [F6]
6. Close the Contents window.

Press ESC
7. With the f variable still highlighted, close VAR-LINK and paste the variable name to the entry line. Notice that "(" is pasted.
Press ENTER
8. Complete the operation.

Press $2 \square$ ENTER
5*f(2)

## Archiving a variable

## Steps and keystrokes <br> Display

1. Redisplay VAR-LINK, and highlight the variable you want to archive.

The previous change in view is no longer in effect. The screen lists all defined
 variables.

Press 2nd [VAR-LINK] (use $\odot$ to highlight $\mathrm{x} 1)$
2. Use the F1 Manage toolbar menu to archive the variable.
andicates the variable is archived.
Press F1 8


3. Return to the Home screen and use the archived variable in a calculation.

| - 5•f(2) |  |  | 40 |
| :---: | :---: | :---: | :---: |
| -6. $\times 1$ |  |  | 30 |
| $6 * \times 1$ |  |  |  |
| Hill | Rifl illto | FUNC | 6, 30 |

[ HOME 6 区 X1 ENTER
4. Attempt to store a different value to the archived variable.

Press 10 STOD X1 ENTER

5. Cancel the error message.

Press ESC
6. Use VAR-LINK to unarchive the variable.

Press [2nd [VAR-LINK] (use $\odot$ to highlight x1) F1 9
7. Return to the Home screen and store a different value to the unarchived variable.
[- HOME ENTER

| -5.f(2) |  |  | 40 |
| :---: | :---: | :---: | :---: |
| - $6 \cdot \times 1$ |  |  | 30 |
| - $6 \cdot \times 110 \rightarrow \times 1$ |  |  |  |
| Error: Variable is lockep |  |  |  |
| -10 ${ }^{\text {a a }}$ |  |  | 10 |
| 1093 |  |  |  |
| - $\mathrm{Minl\mid m}$ | Fint illta | FINIC | Br 30 |

## Deleting variables

Steps and keystrokes Display

1. Display VAR-LINK, and use the F5 All toolbar menu to select all variables.

A $\checkmark$ mark indicates items that are selected. Notice that this also selected the MAIN folder.

Note: Instead of using F5 (if you don't want to delete all your variables), you can select individual variables. Highlight each variable to delete and press (F4).

## Press 551

2. Use F1 to delete.

Note: You can press $\square$ (instead of $F 11$ 1) to delete the marked variables.

Press F1 1
3. Confirm the deletion.

Press ENTER
UAFi-LIAK
4. Because F5 $\mathbf{1}$ also selected the MAIN folder, an error message states that you cannot delete the MAIN folder.
Acknowledge the message.


CTEIT

When VAR-LINK is redisplayed, the deleted variables are not listed.

Press ENTER
5. Close VAR-LINK and return to the current application (Home screen in this example).

When you use ESC (instead of ENTER) to close VAR-LINK, the highlighted name is not pasted to the entry line.

Press ESC

## Operating the Calculator

## Turning the Calculator On and Off

You can turn your graphing calculator on and off manually by using the ON and 2nd [OFF] (or $\square[0 F F]$ ) keys. To prolong battery life, the APD ${ }^{\text {TM }}$ (Automatic Power Down ${ }^{\text {TM }}$ ) feature lets the calculator turn itself off automatically.

## Turning the Calculator On

Press ON.

- If you turned the unit off by pressing 2nd [0FF], the unit returns to either the Apps desktop or the Home screen.
- If you turned the unit off by pressing $\square 0 \mathrm{FF}]$ or if the unit turned itself off through APD, the unit returns to whichever application you used last.


## Turning the Calculator Off

You can use either of the following keys to turn off your graphing calculator.

| Press: | Description |
| :---: | :---: |
| 2nd [0FF] (press 2nd and then press [0FF]) | Settings and memory contents are retained by the Constant Memory ${ }^{\text {TM }}$ feature. However: <br> - You cannot use 2nd [0FF] if an error message is displayed. <br> - When you turn the calculator on again, it displays either the Home screen or the Apps desktop (regardless of the last application you used). |
| - [OFF] <br> (press $\square$ and then press [0FF]) | Similar to 2nd [0FF] except: <br> - You can use [off] if an error message is displayed. <br> - When you turn the calculator on again, it will be exactly as you left it. |

Note: [OFF] is the second function of the $[0 N$ key.

## APD (Automatic Power Down)

After several minutes without any activity, the calculator turns itself off automatically. This feature is called APD.

When you press ON, the calculator will be exactly as you left it.

- The display, cursor, and any error conditions are exactly as you left them.
- All settings and memory contents are retained.

APD does not occur if a calculation or program is in progress, unless the program is paused.

## Setting the Display Contrast

The brightness and contrast of the display depend on room lighting, battery freshness, viewing angle, and the adjustment of the display contrast. The contrast setting is retained in memory when the graphing calculator is turned off.

## Adjusting the Display Contrast

You can adjust the display contrast to suit your viewing angle and lighting conditions.

| To: | Press and hold both: |
| :--- | :--- |
| Decrease (lighten) the contrast | $\square$ and $\square$ |
| Increase (darken) the contrast | $\square$ and $\square$ |



Contrast keys
If you press and hold $\square \square$ or $\square$ too long, the display may go completely black or blank. To make finer adjustments, hold $\square$ and then tap $\square$ or $\square$.

## When to Replace Batteries

As the batteries get low, the display begins to dim (especially during calculations) and you must increase the contrast. If you have to increase the contrast frequently, replace the four alkaline batteries.

Note: The display may be very dark after you change batteries. Use $\square$ to lighten the display.

The status line along the bottom of the display also gives battery information.

| Indicator in status line | Description |
| :--- | :--- |
| EHTT | Batteries are low. |
| EHTT | Replace batteries as soon as possible. |

## The Tl-89 Titanium Keyboard

Most keys can perform two or more functions, depending on whether you first press a modifier key.

(1) F1 - 2nd [F8] open toolbar menus. Select applications (when used with $\quad$ )
(2) 2nd, $\uparrow$, $\uparrow$, and alpha add functionality by increasing the available key commands.
(3) $\mathrm{X}, \mathrm{Y}$, and Z are often used in symbolic calculations.
(4) $(1),(1), \ominus$, and $\odot$ move the cursor.
(5) APPS lets you select an application.
( ENTER evaluates an expression, executes an instruction, selects a menu item, etc.

## Modifier Keys

## Modifier Keys

| Modifier | Description |
| :--- | :--- |
| 2nd | Accesses the second function of the next key you <br> press. On the keyboard, these are printed in the same <br> color as the 2nd key. |
| (diamond) | Activates keys that select certain applications, menu <br> items, and other operations from the keyboard. On the <br> keyboard, these are printed in the same color as the <br> key. |
| (shift) | Types an uppercase character for the next letter key <br> you press.t is also used with © and © to highlight <br> characters in the entry line for editing purposes. <br> alphaUsed to type alphabetic letters, including a space <br> character. On the keyboard, these are printed in the <br> same color as the alpha key. |

Note: Information is available about using 1 and alpha.

## Examples of [2nd] and [diamond] Modifiers

The ESC key is one of several keys that can perform three operations, depending on whether you first press 2nd or $\square$.

The following TI-89 Titanium example shows using the 2nd or $\square$ modifier key with the ESC key.

2nd [QUIT] accesses QUIT, which is the same color as the 2nd key.


ESC accesses the key's primary function.

Some keys perform only one additional operation, which may require either 2nd or $\square$, depending on the color in which the operation is printed on the keyboard and where it is positioned above the key.

| CUT | On the TI-89 Titanium, |
| :--- | :--- |
| 2nd | [CUT] accesses CUT, which is |
|  | the same color as the - key. |

When you press a modifier such as 2nd or $\bullet$, a 2ND or indicator appears in the status line at the bottom of the display. If you press a modifier by accident, press it again (or press ESC) to cancel its effect.

## Other Important Keys You Need to Be Familiar With

| Key | Description |
| :--- | :--- |
| $\square[\mathrm{Y}=]$ | Displays the $\mathrm{Y}=$ Editor. |
| $\square[$ WINDOW $]$ | Displays the Window Editor. |
| $[$ GRAPH $]$ | Displays the Graph screen. |
| $[$ TBLSET $]$ | Sets parameters for the Table screen. |


| Key | Description |
| :---: | :---: |
| $\square$ [TABLE] | Displays the Table screen. |
| 目: <br> - [CUT] <br> - [COPY] <br> - [PASTE] | These keys let you edit entered information by performing a cut, copy, or paste operation. |
| 2nd [ [] | Toggles between the last two chosen Apps or between split screen portions. |
| 2nd [CUSTOM] | Toggles the custom menu on and off. |
| 2nd [ $\bullet$ ] | Converts measurement units. |
| - [-] | Designates a measurement unit. |
| $\square$ | Deletes the character to the left of the cursor (backspaces). |
| 2nd [INS] | Toggles between insert and overtype mode for entering information. |
| $\square$ [DEL] | Deletes the character to the right of the cursor. |
| [ | Enters the "with" operator, which is used in symbolic calculations. |
| $\begin{aligned} & \text { 2nd [ }[\mathrm{J}], \\ & \text { 2nd }[d] \end{aligned}$ | Performs integrations and derivatives. |
| 2nd [ $\angle$ ] | Designates an angle in polar, cylindrical, and spherical coordinates. |
| 2nd [MATH] | Displays the MATH menu. |


| Key | Description |
| :--- | :--- |
| 2nd [MEM] | Displays the MEMORY screen. |
| 2nd [VAR-LINK] | Displays the VAR-LINK screen for managing variables <br> and Flash applications. |
| 2nd [RCL] | Recalls the contents of a variable. |
| [2nd [UNITS] | Displays the UNITS dialog box. |
| 2nd [CHAR] | Displays the CHAR menu, which lets you select Greek <br> letters, international accented characters, etc. |
| 2nd [ENTRY], | Recalls the previous entry and the last answer, <br> respectively. |
| 2nd [ANS] |  |

## Entering Alphabetic Characters

Alphabetic characters are used in expressions such as $\mathbf{x}^{2}+\mathbf{y}^{2}$ to enter variable names and in the Text Editor (Text Editor module).

## Entering a Letter Character on the TI-89 Titanium

The letters $x, y, z$, and $t$ are commonly used in algebraic expressions. So that you can type them quickly, these letters are primary keys on the TI-89 Titanium keyboard.
$\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \mathbf{T}$

Other letters are available as the alpha function of another key, similar to the 2nd and modifiers described in the previous section. For example:
2nd $\left[{ }^{\prime}\right]$ types ${ }^{\prime}$, which is
the same color as the
2nd key.

Typing Alphabetic Characters on the TI-89 Titanium

| To: | Press: |
| :---: | :---: |
| Type a single lowercase alpha character. | alpha and then the letter key (status line shows $\mathbf{l}$ ) |
| Type a single uppercase alpha character. | $\dagger$ and then the letter key (status line shows *) |
| Type a space. | alpha [-] (alpha function of the (-) key) |
| Turn on lowercase alpha-lock. | 2nd [a-lock] <br> (status line shows |
| Turn on uppercase ALPHA-lock. | T [a-lock] <br> (status line shows |
| Turn off alpha-lock. | alpha (turns off upperand lowercase lock) |

## Notes:

- On the TI-89 Titanium, you do not need alpha or alpha-lock to type $x, y, z$, or $t$. But you must use 1 or uppercase ALPHA-lock for X, Y, Z, or T.
- On the TI-89 Titanium, alpha-lock is always turned off when you change applications, such as going from the Text Editor to the Home screen.

On the $\mathrm{TI}-89$ Titanium, while either type of alpha-lock is on:

- To type a period, comma, or other character that is the primary function of a key, you must turn alpha-lock off.
- To type a second function character such as 2nd [i], you do not need to turn alphalock off. After you type the character, alpha-lock remains on.


## Automatic Alpha-Lock in TI-89 Titanium Dialog Boxes

There are certain times when you do not need to press alpha or [2nd [a-lock] to type alphabetic characters on the TI-89 Titanium. Automatic alpha-lock is turned on whenever a dialog box is first displayed. The automatic alpha-lock feature applies to these dialog boxes:

| Dialog box | Alpha-lock |
| :--- | :--- |
| Catalog dialog box | All commands are listed in alphabetical order. <br> Press a letter to go to the first command that <br> begins with that letter. |
| Units dialog box | In each unit category, type the first letter of a <br> unit or constant. See Constants and <br> Measurement Units for more information. |
| Dialog boxes with entry <br> fields | These include, but are not limited to: Create <br> New Folder, Rename, and Save Copy As. |

Note: To type a number, press alpha to turn alpha-lock off. Press alpha or 2nd [a-lock] to resume typing letters.

Alpha-lock is not turned on in dialog boxes that require numeric-only entries. The dialog boxes that accept only numeric entries are: Resize Matrix, Zoom Factors, and Table Setup.

## For Special Characters

Use the [2nd [CHAR] menu to select from a variety of special characters. For more information, refer to "Entering Special Characters" in the Text Editor module.

## Entering Numbers

The keypad lets you enter positive and negative numbers for your calculations. You can also enter numbers in scientific notation.

## Entering a Negative Number

1. Press the negation key $(-)$. (Do not use the subtraction key $\square$.)
2. Type the number.

To see how your graphing calculator evaluates a negation in relation to other functions, refer to the Equation Operating System (EOS ${ }^{\text {TM }}$ ) hierarchy in the Technical Reference
module. For example, it is important to know that functions such as $x^{2}$ are evaluated before negation.

Use $\square$ and $\square$ to include parentheses if you


If you use $\square$ instead of $-($ (or vice versa), you may get an error message or you may get unexpected results. For example:

- 9 区 ( $-7=-63$
- but -
$9 \boxtimes \square 7$ displays an error message.
- $6 \square 2=4$
- but -
$6(-1) 2=-12$ since it is interpreted as 6(-2), implied multiplication.
- $-(-)$ - $4=2$
- but -
$\square 2 \oplus 4$ subtracts 2 from the previous answer and then adds 4 .
Important: Use $\square$ for subtraction and use $-(-)$ for negation.


## Entering a Number in Scientific Notation

1. Type the part of the number that precedes the exponent. This value can be an expression.
2. Press:

EE
E appears in the display.
3. Type the exponent as an integer with up to 3 digits. You can use a negative exponent.

Entering a number in scientific notation does not cause the answers to be displayed in scientific or engineering notation.

The display format is determined by the mode settings and the magnitude of the number.


## Entering Expressions and Instructions

You perform a calculation by evaluating an expression. You initiate an action by executing the appropriate instruction. Expressions are calculated and results are displayed according to the mode settings.

## Definitions

| Expression | Consists of numbers, variables, operators, functions, and their arguments that evaluate to a single answer. For example: $\pi r^{2}+3$. <br> - Enter an expression in the same order that it normally is written. <br> - In most places where you are required to enter a value, you can enter an expression. |
| :---: | :---: |
| Operator | Performs an operation such as,,,$+-^{\wedge}$. <br> - Operators require an argument before and after the operator. For example: $4+5$ and $5^{\wedge} 2$. |
| Function | Returns a value. <br> - Functions require one or more arguments (enclosed in parentheses) after the function. For example: $\sqrt{(5)}$ and $\min (5,8)$. |

Instruction Initiates an action.

- Instructions cannot be used in expressions.
- Some instructions do not require an argument. For example: ClrHome.
- Some require one or more arguments. For example: Circle 0,0,5.
Note: For instructions, do not put the arguments in parentheses.


## Notes:

- The Technical Reference module describes all of the built-in functions and instructions.
- This guidebook uses the word command as a generic reference to both functions and instructions.


## Implied Multiplication

The graphing calculator recognizes implied multiplication, provided it does not conflict with a reserved notation.

If you enter: The calculator interprets it as:

| Valid | $2 \pi$ | $2 * \pi$ |
| :--- | :--- | :--- |
|  | $4 \sin (46)$ | $4 * \sin (46)$ |
|  | $5(1+2)$ or $(1+2) 5$ | $5 *(1+2)$ or $(1+2) * 5$ |
|  | $[1,2] \mathrm{a}$ | $[\mathrm{a} 2 \mathrm{a}]$ |
|  | $2(\mathrm{a})$ | $2 * \mathrm{a}$ |
| Invalid | xy | Single variable named xy |
|  | $\mathrm{a}(2)$ | Function call |
|  | $\mathrm{a}[1,2]$ | Matrix index to element $\mathrm{a}[1,2]$ |

## Parentheses

Expressions are evaluated according to the Equation Operating System (EOS ${ }^{\text {TM }}$ ) hierarchy described in the Technical Reference module. To change the order of evaluation or just to ensure that an expression is evaluated in the order you require, use parentheses.

Calculations inside a pair of parentheses are completed first. For example, in 4(1+2), EOS first evaluates (1+2) and then multiplies the answer by 4.

## Entering an Expression

Type the expression, and then press ENTER to evaluate it. To enter a function or instruction name on the entry line, you can:

- Press its key, if available. For example, press: 2nd [sin]
- or -
- Select it from a menu, if available. For example, select 2:abs from the Number submenu of the MATH menu.
- or -
- Type the name letter-by-letter from the keyboard. (On the TI-89 Titanium, use alpha and [2nd [a-lock] to type letters.) You can use any mixture of uppercase or lowercase letters. For example, type $\boldsymbol{\operatorname { s i n }}$ ( or $\mathbf{S i n}($.


## Example

Calculate $3.76 \div(-7.9+\sqrt{5})+2$ log 45 . $\begin{aligned} & \text { Type the function name in } \\ & \text { this example. }\end{aligned}$

## TI-89 Titanium

| Press | Display |
| :---: | :---: |
| $\begin{aligned} & 3.76 \div \\ & 10[(-) 7.9 \square \\ & \text { 2nd }[\mathrm{v}] \end{aligned}$ | $\square$ <br> 3.76/( $-7.9+\sqrt{ }($ <br> 2nd $[v]$ inserts $\sqrt{ }$ ( because its argument must be in parentheses. |
| $5 \square \square$ | $3.76 /(-7.9+\sqrt{(5))}$ <br> Use 1 once to close $\sqrt{ }(5)$ and again to close $(-7.9+\sqrt{5})$. |
| + 2 <br> 2nd [a-lock] LOG <br> alpha $\square$ 145 $\square$ | $3.76 /(-7.9+\sqrt{(5)})+2 \log (45)$ <br> log requires ( ) around its argument. |
| ENTER |  |

Note: You can also select log by using
CATALOG

## Entering Multiple Expressions on a Line

To enter more than one expression or instruction at a time, separate them with a colon by pressing 2nd [:].

(1) Displays last result only.
(2) $\rightarrow$ is displayed when you press STOD to store a value to a variable.

## If an Entry or Answer Is Too Long for One Line

In the history area, if both the entry and its answer cannot be displayed on one line, the answer is displayed on the next line.

If an entry or answer is too long to fit on one line, is displayed at the end of the line.


To view the entire entry or answer:

1. Press $\Theta$ to move the cursor from the entry line up into the history area. This highlights the last answer.
2. As necessary, use $\Theta$ and $\Theta$ to highlight the entry or answer you want to view. For example, $\odot$ moves from answer to entry, up through the history area.
3. Use (1) and © 1 or 2nd (1) and $2 n d$ (1) to scroll right and left.

Note: When you scroll to the right, 4 is
 displayed at the beginning of the line.
4. To return to the entry line, press ESC.

## Continuing a Calculation

When you press ENTER to evaluate an expression, the graphing calculator leaves the expression on the entry line and highlights it. You can continue to use the last answer or enter a new expression.

| If you press: | The calculator: |
| :---: | :---: |
|  | Replaces the entry line with the variable ans(1), which lets you use the last answer as the beginning of another expression. |
| Any other key | Erases the entry line and begins a new entry. |

## Example

Calculate $3.76 \div(-7.9+\sqrt{5})$. Then add $2 \log 45$ to the result.

## TI-89 Titanium

| Press | Display |
| :---: | :---: |
| 3.76 [田 $7.9 \pm$ | - 3.76 -. 66 |
| 2nd [ r ] $5 \square \square$ | $-\frac{-7.9+\sqrt{5}}{-.66}$ |
| ENTER | -.6638497522033+2.204258 |
| $\pm 2$ 2nd [a-lock] LOG |  |
| alpha 1045 | When you press $\dagger$, the |
| ENTER | entry line is replaced with |
|  | the variable ans(1), which contains the last answer. |

## Stopping a Calculation

When a calculation is in progress, BUSY appears on the right end of the status line. To stop the calculation, press 0 N .

There may be a delay before the Break message is displayed.


Press ESC to return to the current application.

## Formats of Displayed Results

A result may be calculated and displayed in any of several formats. This section describes the modes and their settings that affect the display formats. You can check or change your current mode settings.

## Pretty Print Mode

By default, Pretty Print = ON. Exponents, roots, fractions, etc., are displayed in the same form in which they are traditionally written. You can use MODE to turn pretty print off and on.

| ON | Pretty Print |
| :---: | :---: |
| $\pi^{2}, \frac{\pi}{2}, \sqrt{\frac{x-3}{2}}$ | OFF |

The entry line does not show an expression in pretty print. If pretty print is turned on, the history area will show both the entry and its result in pretty print after you press ENTER.

## Exact/Approx Mode

By default, Exact/Approx = AUTO. You can use MODE to select from three settings.

Because AUTO is a combination of the other two settings, you should be familiar with all

## 19RUTO 3: APFRDMIMATE

 three settings.EXACT - Any result that is not a whole number is displayed in a fractional or symbolic form ( $1 / 2, \pi, \sqrt{2}$, etc.).


Note: By retaining fractional and symbolic forms, EXACT reduces rounding errors that could be introduced by intermediate results in chained calculations.

APPROXIMATE - All numeric results, where possible, are displayed in floating-point (decimal) form.

Note: Results are rounded to the precision of your graphing calculator and displayed according to current mode settings.


Because undefined variables cannot be evaluated, they are treated algebraically. For example, if the variable $r$ is undefined, $\pi r^{2}=3.14159 \cdot r^{2}$.

AUTO - Uses the EXACT form where possible, but uses the APPROXIMATE form when your entry contains a decimal point. Also, certain functions may display APPROXIMATE results even if your entry does not contain a decimal point.

| - $2 \cdot \pi$ | $2 \cdot \pi$ | A decimal in the entry forces a floating-point result. |
| :---: | :---: | :---: |
| - $2 . \cdot \pi$ | 6.28319 |  |
| - $\sqrt{4 / 7}$ | $\frac{2 \cdot \sqrt{7}}{7}$ |  |
| $\sqrt{\frac{4 .}{7}}$ | . 755929 |  |
| $\frac{5}{5(4.1}$ | (16 $\quad 4 \times 30$ |  |

Note: To retain an EXACT form, use fractions instead of decimals. For example, use 3/2 instead of 1.5.

The following chart compares the three settings.

| Entry | Exact <br> Result | Approximate <br> Result | Auto <br> Result |
| :--- | ---: | ---: | ---: |
| $8 / 4$ | 2 | 2. | 2 |
| $8 / 6$ | $4 / 3$ | 1.33333 | $4 / 3$ |
| $8.5 * 3$ | $51 / 2$ | 25.5 | 25.5 |

Note: To evaluate an entry in APPROXIMATE form, regardless of the current setting, press ENTER.

## Display Digits Mode

By default, Display Digits = FLOAT 6, which means that results are rounded to a maximum of six digits. You can use MODE to select different settings. The settings apply to all exponential formats.

Internally, the calculator calculates and retains all decimal results with up to 14 significant digits (although a maximum of 12 are displayed).

| Setting | Example |  | Description |
| :--- | :--- | :--- | :--- |
| FIX | 123. | (FIX 0) | Results are rounded to the |
| (0-12) | 123.5 | (FIX 1) | selected number of decimal <br>  <br>  <br>  <br>  <br> 123.46 <br> places. <br> (FIX 2) |
| FLOAT | 123.457 | (FIX 3) |  |
|  |  |  | Number of decimal places varies, <br> depending on the result. |
| FLOAT | $1 . E 2$ | (FLOAT 1) | Results are rounded to the total |
| (1-12) | 1.2 E 2 | (FLOAT 2) | number of selected digits. |
|  | 123. | (FLOAT 3) |  |
|  | 123.5 | (FLOAT 4) |  |
|  | 123.46 | (FLOAT 5) |  |
|  | 123.457 | (FLOAT 6) |  |

## Notes:

- Regardless of the Display Digits setting, the full value is used for internal floatingpoint calculations to ensure maximum accuracy.
- A result is automatically shown in scientific notation if its magnitude cannot be displayed in the selected number of digits.


## Exponential Format Mode

By default, Exponential Format = NORMAL.
You can use MODE to select from three
 settings.

| Setting | Example | Description |
| :---: | :---: | :---: |
| NORMAL | 12345.6 | If a result cannot be displayed in the number of digits specified by the Display Digits mode, the calculator switches from NORMAL to SCIENTIFIC for that result only. |
| SCIENTIFIC | $\frac{1.23456 E}{1} \frac{4}{1}$ | $1.23456 \times 10^{4}$ |
| ENGINEERING | $\frac{1.23456 E}{\prod_{3}} \frac{3}{4}$ | $12.3456 \times 10^{3}$ |

(1) Always 1 digit to the left of the decimal point.
(2) Exponent (power of 10).
(3) May have 1, 2, or 3 digits to the left of the decimal point.
(4) Exponent is a multiple of 3 .

Note: In the history area, a number in an entry is displayed in SCIENTIFIC if its absolute value is less than 001.

## Editing an Expression in the Entry Line

Knowing how to edit an entry can be a real time-saver. If you make an error while typing an expression, it's often easier to correct the mistake than to retype the entire expression.

## Removing the Highlight from the Previous Entry

After you press ENTER to evaluate an expression, the calculator leaves that expression on the entry line and highlights it. To edit the expression, you must first remove the highlight; otherwise, you may clear the expression accidentally by typing over it.

To remove the highlight, move the cursor toward the side of the expression you want to edit.

(1) moves the cursor to the beginning.
(1) moves the cursor to the end of the expression.

## Moving the Cursor

After removing the highlight, move the cursor to the applicable position within the expression.

| To move the cursor: | Press: |  |
| :--- | :--- | :--- |
| Left or right within an <br> expression. | (1) or (1) | Hold the pad to repeat the <br> movement. |
| To the beginning of the <br> expression. | 2nd (1) |  |
| To the end of the expression. | 2nd (1) |  |

Note: If you accidentally press $\Theta$ instead of $(\mathbb{1}$ or ( $(1)$, the cursor moves up into the history area. Press ESC or press $\Theta$ until the cursor returns to the entry line.

## Deleting a Character

| To delete: | Press: |  |
| :--- | :--- | :--- |
| The character to the left of <br> the cursor. | $\boxed{ }$ | Hold $\square$ to delete multiple <br> characters. |
| The character to the right of |  |  |
| the cursor. |  |  |
| All characters to the right of <br> the cursor. | CLEAR | If there are no characters to <br> (once only) <br> the right of the cursor, CLEAR <br> erases the entire entry line. |

## Clearing the Entry Line

To clear the entry line, press:

- CLEAR if the cursor is at the beginning or end of the entry line.
- or -
- CLEAR CLEAR if the cursor is not at the beginning or end of the entry line. The first press deletes all characters to the right of the cursor, and the second clears the entry line.


## Inserting or Overtyping a Character

The calculator has both an insert and an overtype mode. By default, the calculator is in the insert mode. To toggle between the insert and overtype modes, press 2nd [INS].

| If in: | The next character you <br> type: |
| :--- | :--- |
| Inserfe mode |  |
| $\quad$ Thin cursor between characters | Will be inserted at the <br> cursor. |
| Overtbpe mode <br> $L$ Cursor highlights a character | Will replace the highlighted <br> character. |
|  |  |

Note: Look at the cursor to see if you're in insert or overtype mode.

## Replacing or Deleting Multiple Characters

First, highlight the applicable characters. Then, replace or delete all the highlighted characters.

## To highlight multiple characters:

1. Move the cursor to either side of the characters you want to highlight.

| $\underline{\sin \left(\frac{\pi}{4}\right)}$ | $\frac{\sqrt{2}}{2}$ |
| :---: | :---: |
|  |  |
| To replace $\boldsymbol{\operatorname { s i n }}$ ( with cos(, place the cursor beside sin. |  |

2. Hold 1 and press (1) or (1) to highlight characters left or right of the cursor.


To replace or delete the highlighted characters:

1. Type the new characters.
2. Press $\square$.


Note: When you highlight characters to replace, remember that some function keys automatically add an open parenthesis.

## Menus

To leave the keyboard uncluttered, the calculator uses menus to access many operations. This section gives an overview of how to select an item from any menu. Specific menus
are described in the appropriate modules.

## Displaying a Menu

| Press: | To display: |
| :--- | :--- |
| F1, [F2, etc. | A toolbar menu - Drops down from the toolbar at the <br> top of most application screens. Lets you select <br> operations useful for that application. |
| APPS | Apps desktop or APPLICATIONS menu - Lets you <br> select from a list of applications. |
| 2nd [CHAR] | CHAR menu - Lets you select from categories of <br> special characters (Greek, math, etc.). |
| 2nd [MATH] | MATH menu - Lets you select from categories of <br> math operations. |
| CATALOG] | CATALOG menu - Lets you select from a complete, <br> alphabetic list of built-in functions and instructions. <br> Also lets you select user-defined functions or Flash <br> application functions (if any have been defined or <br> loaded). |
| 2nd [CUSTOM] | CUSTOM menu - Lets you access a menu that you <br> can customize to list any available function, <br> instruction, or character. The calculator includes a <br> default custom menu, which you can modify or <br> redefine. Refer to the Calculator Home Screen and/or <br> the Programming module for more information on the <br> custom menu. |

## Selecting an Item from a Menu

To select an item from the displayed menu, either:

- Press the number or letter shown to the left of that item. For a letter on the TI-89 Titanium, press alpha and then a letter key.
- or -
- Use the cursor pad $\Theta$ and $\Theta$ to highlight the item, and then press ENTER. (Note that pressing $\Theta$ from the first item moves the highlight to the last item, and vice versa.)



## Items Ending with $>$ (Submenus)

If you select a menu item ending with $\downarrow$, a submenu is displayed. You then select an item from the submenu.


Because of limited screen size, the TI-89 Titanium overlaps these menus.


For items that have a submenu, you can use the cursor pad as described below.

- To display the submenu for the highlighted item, press (1). (This is the same as selecting that item.)
- To cancel the submenu without making a selection, press © (1). (This is the same as pressing ESC.)
- To wrap to the last menu item directly from the first menu item, press $\Theta$. To wrap to the first menu item directly from the last menu item, press $\Theta$.


## Items Containing ". . ." (Dialog Boxes)

If you select a menu item containing "..." (ellipsis marks), a dialog box is displayed for you to enter additional information.


After typing in an input box such as Variable, you must press ENTER twice to save the information and close the dialog box.

## Canceling a Menu

To cancel the current menu without making a selection, press ESC. Depending on whether any submenus are displayed, you may need to press ESC several times to cancel all displayed menus.

## Moving from One Toolbar Menu to Another

To move from one toolbar menu to another without making a selection, either:

- Press the key (F1, F2, etc.) for the other toolbar menu.
- or -
- Use the cursor pad to move to the next (press (®)) or previous (press (1)) toolbar menu. Pressing (1) from the last menu moves to the first menu, and vice versa.

When using ( $\mathfrak{C}$, be sure that an item with a submenu is not highlighted. If so, (1) displays that item's submenu instead of moving to the next toolbar menu.

## Example: Selecting a Menu Item

Round the value of $\pi$ to three decimal places. Starting from a clear entry line on the Home screen:

1. Press 2nd [MATH] to display the MATH menu.
2. Press 1 to display the Number submenu. (Or press ENTER since the first item is automatically highlighted.)
3. Press 3 to select round. (Or press $\Theta \odot$ and ENTER.)
4. Press 2nd $[\pi] \square 3 \square$ and then ENTER to evaluate the expression.
(1) Selecting the function in Step 3 automatically
 typed round( on the entry line.

## Selecting an Application

The graphing calculator has different applications that let you solve and explore a variety of problems. You can select an application from a menu, the Apps desktop, or you can access commonly used applications directly from the keyboard.

## From the APPLICATIONS Menu

1. If the Apps desktop is off, press APPS to display a menu that lists the applications.

Note: To cancel the menu without making a selection, press ESC.
2. Select an application. Either:

- Use the cursor pad $\odot$ or $\Theta$ to highlight the application and then press ENTER.

| AFFLICOATIGAS |  |
| :---: | :---: |
| 1:F1Esh月PFS | * PPFS |
| $2 \%$ Editor |  |
| S: |  |
| 4: Graph |  |
| 5: Table |  |
|  |  |
| 7: Frogram. Ed |  |
| B+TEXE Editor |  |

- or -
- Press the number or letter for that application.

| Application: | Lets you: |
| :--- | :--- |
| FlashApps | Display a list of Flash applications, if any. |
| Y= Editor | Define, edit, and select functions or equations for <br> graphing. |
| Window Editor | Set window dimensions for viewing a graph. |
| Graph | Display graphs. |


| Application: | Lets you: |
| :--- | :--- |
| Table | Display a table of variable values that correspond to <br> an entered function. |
| Data/Matrix Editor | Enter and edit lists, data, and matrices. You can <br> perform statistical calculations and graph statistical <br> plots. |
| Program Editor | Enter and edit programs and functions. |
| Text Editor | Enter and edit a text session. |
| Numeric Solver | Enter an expression or equation, define values for all <br> but one variable, and then solve for the unknown <br> variable. |
| Home | Enter expressions and instructions, and perform <br> calculations. |

## From the Apps Desktop

Use the cursor keys to highlight an application icon on the Apps desktop and press ENTER. The application either opens directly or displays a dialog box. (Your Apps desktop may vary from the one shown below.)

| $\begin{array}{r} \text { F1 } \\ \text { Menu } \end{array}$ | Cabri Geametry |  |  |
| :---: | :---: | :---: | :---: |
| ${ }_{611} \mathrm{~F}$ | Cellsheet |  | \|c|ce |
| Enslish |  |  |  |
| F4 |  |  | [natarkistri. |
| sobialst | 14 |  |  |
| $\mathrm{Moth}^{\mathrm{FS}}$ | 0 |  | $f(x)=0$ |
| F Grofhins | Finumise Grafh | Home | Humaris St... |
| sciente | $x_{1}=\quad \text { - } 1$ | $\boldsymbol{A} \\|$ | $\begin{array}{\|l\|l\|} \hline \mathrm{H}_{1} & \mathrm{l} \\ \hline \hline \\ \hline \hline \end{array}$ |
| $\text { Dr }{ }_{\text {Finfizs }}$ | Folvnomidl ... Frosram Ed. | +10 | mratist E... |
| F-TH\|N | Fifl illla | FUWH: |  |

The most common dialog box lists these options for the application:

| Option | Description |
| :--- | :--- |
| Current | Returns the screen displayed when you last viewed the <br> App. (If there is no current file/variable for the selected <br> App, this option defaults to New if you press ENTER.) |
| Open | Lets you select an existing file. |
| New | Creates a new file with the name typed in the field. |

Select an option and press ENTER. The application appears.
Note: The general term variable is used to refer to the application data files that you create.

Use any of these methods to return to the Apps desktop from within an application:

- Press APPS.
- In full-screen mode, press 2nd [QUIT].
- In split-screen mode, press [2nd [QuIT] to open the full-screen view of the active application, then press [2nd [QUIT] again.

To return to the last open application from the Apps desktop, press 2nd [ $\boxplus$ ].

## From the Keyboard



You can access commonly used applications from the keyboard. On the TI-89 Titanium for example, $\square[\mathrm{Y}=]$ is the same as pressing $\Delta$ and then F 1 . This guidebook uses the notation $\square[\gamma=]$, similar to the notation used in second functions.

| Application: | Press: |
| :--- | :--- |
| Home | $[\mathrm{HOME}$ |
|  | $[\mathrm{CALC} \mathrm{HOME}]$ |
| $\mathrm{Y}=$ Editor | $\square[\mathrm{Y}=]$ |
| Window Editor | $\square[$ WINDOW $]$ |
| Graph | $\square[$ GRAPH $]$ |
| Table Setup | $\square[$ TBLSET $]$ |
| Table Screen | $\square[$ TABLE $]$ |

## Setting Modes

Modes control how numbers and graphs are displayed and interpreted. Mode settings are retained by the Constant Memory ${ }^{\text {TM }}$ feature when the graphing calculator is turned off. All numbers, including elements of matrices and lists, are displayed according to the current mode settings.

## Checking Mode Settings

Press MODE to display the MODE dialog box, which lists the modes and their current settings.

(1) There are three pages of mode listings. Press F1, F2, or F3 to quickly display a particular page.
(2) Indicates you can scroll down to see additional modes.
(3) $\rightarrow$ indicates that you can press (1) or © $(1)$ to display and select from a menu.

Note: Modes that are not currently valid are dimmed. For example, on Page 2, Split 2 App is not valid when Split Screen = FULL. When you scroll through the list, the cursor skips dimmed settings.

## Changing Mode Settings

From the MODE dialog box:

1. Highlight the mode setting you want to change. Use $\Theta$ or $\Theta$ (with F1, F2], or F3) to scroll through the list.
2. Press $(\mathfrak{1})$ or $(\mathbb{1})$ to display a menu that lists the valid settings. The current setting is highlighted.
3. Select the applicable setting. Either:

- Use $\Theta$ or $\Theta$ to highlight the setting and press ENTER.
- or -
- Press the number or letter for that setting.

Note: To cancel a menu and return to the MODE dialog box without making a selection, press ESC.
4. Change other mode settings, if necessary.
5. When you finish all your changes, press ENTER to save the changes and exit the dialog box.

Important: If you press ESC instead of ENTER to exit the MODE dialog box, any mode changes you made will be canceled.

## Overview of the Modes

Note: For detailed information about a particular mode, look in the applicable section of this guidebook.

| Mode | Description |
| :--- | :--- |
| Graph | Type of graphs to plot: FUNCTION, PARAMETRIC, <br> POLAR, SEQUENCE, 3D, or DE. |
| Current Folder | Folder used to store and recall variables. Unless you <br> have created additional folders, only the MAIN folder is <br> available. Refer to "Using Folders to Store Independent <br> Sets of Variables" in Calculator Home Screen. |
| Display Digits | Maximum number of digits (FLOAT) or fixed number of <br> decimal places (FIX) displayed in a floating-point result. <br> Regardless of the setting, the total number of displayed <br> digits in a floating-point result cannot exceed 12. |
| Angle | Units in which angle values are interpreted and displayed: <br> RADIAN or DEGREE. |


| Mode | Description |
| :--- | :--- |
| Exponential <br> Format | Notation used to display results: NORMAL, SCIENTIFIC, <br> or ENGINEERING. |
| Complex | Format used to display complex results, if any: <br> REAL (complex results are not displayed unless you use <br> a complex entry), RECTANGULAR, or POLAR. |
| Vector Format | Format used to display 2- and 3-element vectors: <br> RECTANGULAR, CYLINDRICAL, or SPHERICAL. |
| Pretty Print | Turns the pretty print display feature OFF or ON. |
| Split Screen | Splits the screen into two parts and specifies how the <br> parts are arranged: FULL (no split screen), <br> TOP-BOTTOM, or LEFT-RIGHT. Refer to the Split |
| Screens module. |  |


| Mode | Description |
| :--- | :--- |
| Unit System | Lets you select from three systems of measurement to <br> specify the default units for displayed results: $\mathbf{S I}$ (metric or <br> MKS); Eng/US (feet, pounds, etc.); or Custom. |
| Custom Units | Lets you select custom defaults. The mode is dimmed <br> until you select Unit System, 3:CuSTOM. |
| Language | Lets you localize the calculator into one of several <br> languages, depending on which language Flash <br> applications are installed. |
| Apps Desktop | Turns the Apps desktop ON or OFF. |
| Using the Clean Up Menu to Start a New Problem |  |

On the Home screen, the Clean Up toolbar menu lets you start a new calculation from a cleared state without resetting the memory.

## Clean Up Toolbar Menu

To display the Clean Up menu from the Home screen, press:
2nd [F6]


| Menu Item | Description |
| :--- | :--- |
| Clear a-z | Clears (deletes) all single-character variable names in the <br> current folder, unless the variables are locked or <br> archived. You will be prompted to press ENTER to confirm <br> the action. |
|  | Single-character variable names are often used in <br> symbolic calculations such as: <br> solve(a.x ${ }^{2}$ +b:x+c=0,x) <br> If any of the variables have already been assigned a <br> value, your calculation may produce misleading results. <br> To prevent this, you can select 1:Clear a-z before <br> beginning the calculation. |

## Notes:

- When defining a variable that you want to retain, use more than one character in the name. This prevents it from being deleted inadvertently by 1:Clear a-z.
- For information about checking and resetting memory or other system defaults, refer to Memory and Variable Management.


## Using the Catalog Dialog Box

The CATALOG provides a way to access any built-in command (functions and instructions) from one convenient list. In addition, the CATALOG dialog box lets you select functions used in Flash applications or user-defined functions (if any have been loaded or defined).

## Displaying the CATALOG

To display the CATALOG dialog box, press:
CATALOG

The CATALOG defaults to F2 Built-in, which displays an alphabetic list of all preinstalled commands (functions and instructions).

(1) Defaults to F2 Built-in.
(2) F1 Help displays a command's parameters in a dialog box.
(3) F3 and F4 allow access to Flash application functions and User-Defined functions and programs.

Note: Options that are not currently valid are dimmed. For example, F3 Flash Apps is dimmed if you have not installed a Flash application. F4 User-Defined is dimmed if you have not created a function or a program.

## Selecting a Built-in Command from the CATALOG

When you select a command, its name is inserted in the entry line at the cursor location. Therefore, you should position the cursor as necessary before selecting the command.

1. Press:

CATALOG
2. Press F2 Built-in.


- Commands are listed in alphabetical order. Commands that do not start with a letter ( $+, \%, \sqrt{ }, \Sigma$, etc.) are at the end of the list.
- To exit the CATALOG without selecting a command, press ESC.

Note: The first time you display the Built-in list, it starts at the top of the list. The next time you display the list, it starts at the same place you left it.
3. Move the indicator to the command, and press ENTER.

| To move the indicator: | Press or type: |
| :--- | :--- |
| One function or program at a time | $\Theta$ or $\Theta$ |
| One page at a time | 2nd $\Theta$ or 2nd $\Theta$ |
| To the first function that | The letter key. (On the TI-89 |
| begins with a specified letter | Titanium, do not press alpha first. If <br> you do, you need to press alpha or <br>  <br>  <br>  <br>  <br> 2nd [a-lock] again before you can <br> type a letter.) |

Note: From the top of the list, press $\Theta$ to move to the bottom. From the bottom, press $\Theta$ to move to the top.

## Information about Parameters

For the command indicated by , the status line shows the required and optional parameters, if any, and their type.

(1) Indicated command and its parameters
(2) Brackets [] indicate optional parameters

From the example above, the syntax for factor is:

| factor(expression) | required |
| :--- | ---: |
| - or - |  |
| factor(expression, variable) | optional |

Note: For details about the parameters, refer to that command's description in the Technical Reference module.

## Viewing CATALOG Help

You can display a command's parameters in a dialog box by pressing F1 Help. The parameters are the same as those displayed on the status line.

Indicated command and its parameters.


Some commands, such as CIrDraw, do not require parameters. If you select one of these commands, parameters will not display on the status line and you will see Unavailable if you press F1 Help.

Press ESC to exit the CATALOG Help dialog box.

## Selecting a Flash Application Function

A Flash application may contain one or more functions. When you select a function, its name is inserted in the entry line at the cursor location. Therefore, you should position the cursor as necessary before selecting the function.

1. Press:

CATALOG
2. Press F3 Flash Apps. (This option is dimmed if no Flash applications are installed.)


- The list is alphabetized by function name. The left column lists functions. The right column lists the Flash application that contains the function.
- Information about a function is displayed in the status line.
- To exit without selecting a function, press ESC.

3. Move the indicator to the function, and press ENTER.
To move the indicator: Press or type:

| One function or program at a time | $\Theta$ or $\Theta$ |
| :--- | :--- |
| One page at a time | 2nd $\Theta$ or 2nd $\Theta$ |
| To the first function that | The letter key. (On the TI-89 |
| begins with a specified letter | Titanium, do not press alpha first. If <br> you do, you need to press alpha or |
|  | 2nd [a-lock] again before you can <br> type a letter.) |

## Selecting a User-Defined Function or Program

You can create your own functions or programs and then use F4 User-Defined to access them. For instructions on how to create functions, see "Creating and Evaluating User-Defined Functions" in Calculator Home Screen, and "Overview of Entering a Function" in the Programming module. See Programming for instructions on how to create and run a program.

When you select a function or program, its name is inserted in the entry line at the cursor location. Therefore, you should position the cursor as necessary before selecting the function or program.

1. Press:

CATALOG
2. Press F4 User-Defined. (This option is dimmed if you have not defined a function or created a program.)


- The list is alphabetized by function / program name. The left column lists functions and programs. The right column lists the folder that contains the function or program.
- If the function or program's first line is a comment, the comment text is displayed in the status line.
- To exit without selecting a function or program, press ESC.

Note: Use the VAR-LINK screen to manage variables, folders, and Flash applications. See the Memory and Variable Management module.
3. Move the indicator to the function or program, and press ENTER.

| To move the indicator: | Press or type: |
| :--- | :--- |
| One function or program at a time | $\Theta$ or $\Theta$ |
| One page at a time | 2nd $\Theta$ or 2nd $\Theta$ |
| To the first function or program that <br> begins with a specified letter | The letter key. (On the TI-89 <br>  <br>  <br>  <br>  <br>  <br>  <br>  <br>  <br>  <br> you do, you need to press alphal or <br> 2nd [a-lock] again before you can <br> type a letter.) |

## Storing and Recalling Variable Values

When you store a value, you store it as a named variable. You can then use the name instead of the value in expressions. When the calculator encounters the name in an expression, it substitutes the variable's stored value.

## Rules for Variable Names

A variable name:

- Can use 1 to 8 characters consisting of letters and digits. This includes Greek letters (but not $\pi$ ), accented letters, and international letters.
- Do not include spaces.
- The first character cannot be a digit.
- Can use uppercase or lowercase letters. The names AB22, Ab22, aB22, and ab22 all refer to the same variable.
- Cannot be the same as a name that is preassigned by the calculator. Preassigned names include:
- Built-in functions (such as abs) and instructions (such as LineVert). Refer to the Technical Reference module.
- System variables (such as xmin and xmax, which are used to store graphrelated values). Refer to the Technical Reference module for a list.


## Examples

| Variable | Description |
| :--- | :--- |
| myvar | OK |
| a | OK |
| Log | Not OK, name is preassigned to the log function. |
| Log1 | OK |
| 3rdTotal | Not OK, starts with a digit. |
| circumfer | Not OK, more than 8 characters. |

## Data Types

| DataTypes | Examples |
| :--- | :--- |
| Expressions | $2.54,1.25 \mathrm{E} 6,2 \pi, \mathrm{xmin} / 10,2+3 i,(\mathrm{x}-2)^{2}, \sqrt{2} / 2$ |
| Lists | $\{2468\},\{112\}$ |


| DataTypes | Examples |
| :--- | :--- |
| Matrices | $\left[\begin{array}{lll}1 & 0 & 0\end{array}\right],\left[\begin{array}{lll}1 & 0 & 0 \\ 3 & 4 & 6\end{array}\right]$ |
| Character strings | "Hello", "The answer is:", "xmin/10" |
| Pictures | myfunc(arg), ellipse(x,y,r1,r2) |

## Storing a Value in a Variable

1. Enter the value you want to store, which can be an expression.
2. Press STOص. The store symbol $(\rightarrow)$ is displayed.
3. Type the variable name.

Note: TI-89 Titanium users should use

| - $5+8^{3}+$ num 1 |  |  | 517 |
| :---: | :---: | :---: | :---: |
| 5+8^3*积以1 |  |  |  |
| -1ikl\| | Ríci illita | FUAC | 1, 30 | alpha as necessary when typing variable names.

4. Press ENTER.

To store to a variable temporarily, you can use the "with" operator. Refer to "Substituting Values and Setting Constraints" in Symbolic Manipulation.

## Displaying a Variable

1. Type the variable name.
2. Press ENTER.

FiHLIMUTD FUN: $\quad 1,20$

If the variable is undefined, the variable name is shown in the result.

In this example, the variable a is undefined.
Therefore, it is used as a symbolic variable.


Note: Refer to Symbolic Manipulation for information about symbolic manipulation.

## Using a Variable in an Expression

1. Type the variable name into the expression.
2. Press ENTER to evaluate the expression. Note: To view a list of existing variable names, use 2nd [VAR-LINK] as described in Memory and Variable Management.

If you want the result to replace the variable's previous value, you must store the result.


The variable's value_ did not change.


## Recalling a Variable's Value

In some cases, you may want to use a variable's actual value in an expression instead of the variable name.

1. Press 2nd [RCL] to display a dialog box.
2. Type the variable name.
3. Press ENTER twice.


In this example, the value stored in num1 will be inserted at the cursor position in the entry line.

## Status Line Indicators in the Display

The status line is displayed at the bottom of all application screens. It shows information about the current state of the calculator, including several important mode settings.

## Status Line Indicators


(1) Current Folder
(2) Modifier Key
(3) Angle Mode
(4) Exact/Approx Mode
(5) Graph Number
(6) Graph Mode
(7) Replace Batteries
(8 History Pairs, Busy/Pause, Locked Variable

| Indicator | Meaning |
| :---: | :---: |
| Current Folder | Shows the name of the current folder. Refer to "Using Folders to Store Independent Sets of Variables" in Calculator Home Screen. MAIN is the default folder. |
| Modifier Key | Shows which modifier key is in effect, as described below. |
| 2nd | 2nd - will use the second function of the next key you press. |
| - | $\square$ - will use the diamond feature of the next key you press. |
| * | T - will type the uppercase letter for the next key you press. On the TI-89 Titanium, you can use 1 to type a letter without having to use alpha. |
| 1 | alpha — will type the lowercase letter for the next key you press. |
| T | 2nd [a-lock] — lowercase alpha-lock is on. Until you turn this off, will type the lowercase letter for each key you press. To cancel alpha-lock, press alpha. |
| TI | 1 alpha - uppercase ALPHA-lock is on. Until you turn this off, will type the uppercase letter for each key you press. To cancel ALPHA-lock, press alpha. |


| Indicator | Meaning |
| :---: | :---: |
| Angle Mode | Shows the units in which angle values are interpreted and displayed. To change the Angle mode, use the MODE key. |
| RAD | Radians |
| DEG | Degrees |
| Exact/Approx Mode | Shows how answers are calculated and displayed. To change the Exact/Approx mode, use the MODE key. |
| AUTO | Auto |
| EXACT | Exact |
| APPROX | Approximate |
| Graph Number | If the screen is split to show two independent graphs, this indicates which graph is active - G1 or $\mathbf{G 2}$. |
| Graph Mode | Indicates the type of graphs that can be plotted. To change the Graph mode, use the MODE key. |
| FUNC | $y(x)$ functions |
| PAR | $\mathbf{x}(\mathbf{t})$ and $\mathbf{y}(\mathbf{t})$ parametric equations |
| POL | $\mathbf{r}(\theta)$ polar equations |
| SEQ | $u(n)$ sequences |
| 3D | $\mathbf{z ( x , y )} 3 \mathrm{D}$ equations |
| DE | $\mathbf{y}^{\prime}(\mathrm{t})$ differential equations |
| Battery | Displayed only when the batteries are getting low. If BATT is shown with a black background, change the batteries as soon as possible. |


| Indicator | Meaning |
| :---: | :--- |
| History Pairs, <br> Busy/Pause, <br> Archived | The information shown in this part of the status line <br> depends on the application you are using. |
| $23 / 30$ | Displayed on the Home screen to show the number of <br> entry/answer pairs in the history area. Refer to History <br> Information on the Status Line in the Calculator Home <br> Screen module. |
| BUSY | A calculation or graph is in progress. |
| PAUSE | You paused a graph or program. |
| $\mathbf{E}$ | The variable opened in the current editor (Data/Matrix <br> Editor, Program Editor, or Text Editor) is locked or <br> archived and cannot be modified. |

## Notes:

- To cancel 2 nd , $\bullet$, alpha, or $\downarrow$, press the same key again or press a different modifier key.
- If the next key you press does not have a diamond feature or an associated letter, the key performs its normal operation.


## Calculator Home Screen

## Calculator Home Screen

The calculator Home screen is the starting point for math operations, including executing instructions, evaluating expressions, and viewing results.


A blank calculator Home screen

This module describes the parts of the calculator Home screen, how to scroll through or modify the history area; how to use cut, copy, and paste, and more.

Note: The term "calculator Home screen" is used in this module. Other modules use the term "Home screen." Both terms are interchangeable and refer to the same screen.

## Displaying the Calculator Home Screen

When you first turn on your TI-89 Titanium, the Apps desktop is displayed. To display the calculator Home screen, highlight the Home icon and press ENTER. You can also display the calculator Home screen by pressing HOME (TI-89 Titanium). If you turn off the Apps Desktop mode, the calculator Home screen appears automatically.

## Parts of the Calculator Home Screen

The following example contains previously entered data and describes the main parts of the calculator Home screen. Entry/answer pairs in the history area are displayed in "pretty print." Pretty print displays expressions in the same form in which they are written on the board or in textbooks.


## (1) Toolbar

Lets you display menus for selecting operations applicable to the calculator Home screen. To display a toolbar menu, press F1, F2, etc.

## (2) Pretty Print Display

Shows exponents, roots, fractions, etc., in traditional form.

## (3) Last Entry

Your last entry.

## (4) Entry Line

Where you enter expressions or instructions.

## (5) Status Line

Shows the current state of the calculator, including several important mode settings.

## (6 Last Answer

Result of your last entry. Note that results are not displayed on the entry line. Note: (Approx) was used in this example.

The following example shows an answer that is not on the same line as the expression. Note that the answer is longer than the screen width. An arrow ( $\downarrow$ ) indicates the answer is continued. The entry line contains ellipsis (...). Ellipsis indicates the entry is longer than the screen width.


## (1) Last Entry

"Pretty print" is ON. Exponents, roots, fractions, etc., are displayed in the same form in which they are traditionally written.

## (2) History Area

Lists entry/answer pairs you have entered. Pairs scroll up the screen as you make new entries.

## (3) Answer Continues

Highlight the answer and press (1) to scroll right and view the rest of it. Note that the answer is not on the same line as the expression.

## (4) Expression Continues (...)

Press (1) to scroll right and view the rest of the entry. Press 2nd (1) or 2nd (1) to go to the beginning or end of the entry line.

## History Area

The history area shows up to eight previous entry/answer pairs (depending on the complexity and height of the displayed expressions). When the display is filled, information scrolls off the top of the screen. You can use the history area to:

- Review previous entries and answers. You can use the cursor to view entries and answers that have scrolled off the screen.
- Recall or auto-paste a previous entry or answer onto the entry line so that you can re-use or edit it.


## Scrolling through the History Area

Normally, the cursor is in the entry line. However, you can move the cursor into the history area.

| To: | Do this: |
| :---: | :---: |
| View entries or answers that have scrolled off the screen | - From the entry line, press $\Theta$ to highlight the last answer. <br> - Continue using $\Theta$ to move the cursor from answer to entry, up through the history area. |
| Go to the oldest or newest history pair | If the cursor is in the history area, press <br> $\Theta$ or $\Theta \odot$, respectively. |
| View an entry or answer that is too long for one line ( $\boldsymbol{\square}$ is at end of line) | Move the cursor to the entry or answer. Use (1) and (1) to scroll left and right (or 2nd (1) and 2nd (1) to go to the beginning or end), respectively. |

Return the cursor to the entry line

Press ESC, or press $\odot$ until the cursor is back on the entry line.

Note: An example of viewing a long answer is available.

## History Information on the Status Line

Use the history indicator on the status line for information about the entry/answer pairs. For example:

| If the cursor is | Total number of <br> pairs that are <br> on the entry <br> line: |
| :--- | :--- |
| currently saved. |  |
| If the cursor is  <br> in the history Pair number of <br> the highlighted <br> area: entry or answer. | pairs that can be <br> saved. |
| that number of pairs |  |

By default, the last 30 entry/answer pairs are saved. If the history area is full when you make a new entry (indicated by 30/30), the new entry/answer pair is saved and the oldest pair is deleted. The history indicator does not change.

| To: | Do this: |
| :---: | :---: |
| Change the number of pairs that can be saved | Press F1 and select 9:Format, or press |
|  | Then press $(\mathfrak{\square}$, use $\Theta$ or $\Theta$ to highlight the new number, and press ENTER twice. |
| Clear the history area and delete all saved pairs | Press F1 and select 8:Clear Home, or enter CIrHome on the entry line. |
| Delete a particular entry/answer pair | Move the cursor to either the entry or answer. Press $\square$ or CLEAR. |

## Saving the Calculator Home Screen Entries as a Text Editor Script

To save all the entries in the history area, you can save the calculator Home screen to a text variable. When you want to reexecute those entries, use the Text Editor to open the variable as a command script.

## Saving the Entries in the History Area

From the calculator Home screen:

1. Press F1 and select 2:Save Copy As.

2. Specify a folder and text variable that you want to use to store the entries.

Note: Only the entries are saved, not the answers.


| Item | Description |
| :--- | :--- |
| Type | Automatically set as Text and cannot be changed. |
| Folder | Shows the folder in which the text variable will be <br> stored. To use a different folder, press © $(1)$ to display a <br> menu of existing folders. Then select a folder. |
| Variable | Type a valid, unused variable name. |

Note: For information about folders, see the Memory and Variable Management module.
3. Press ENTER (after typing in an input box such as Variable, press ENTER twice).

## Restoring the Saved Entries

Because the entries are stored in a script format, you cannot restore them from the calculator Home screen. (On the calculator Home screen's F1 toolbar menu, 1:Open is not available.) Instead:

1. Use the Text Editor to open the variable containing the saved calculator Home screen entries.

The saved entries are listed as a series of command lines that you can execute individually, in any order.
2. Starting with the cursor on the first line of the script, press F4 repeatedly to execute the commands line by line.
3. Display the restored calculator Home screen.


This split screen shows the Text Editor (with the command line script) and the restored calculator Home screen.

Note: For complete information on using the Text Editor and executing a command script, refer to the Text Editor module.

## Cutting, Copying, and Pasting Information

Cut, copy, and paste operations let you move or copy information within the same application or between different applications. These operations use the clipboard, which is an area in memory that serves as a temporary storage location.

## Auto-paste vs. Cut/Copy/Paste

Auto-paste is a quick way to copy an entry or answer in the history area and paste it to the entry line.

1. Use $\Theta$ and $\Theta$ to highlight the item in the history area.
2. Press ENTER to auto-paste that item to the entry line.

To copy or move information in the entry line, you must use a cut, copy, or paste operation. (You can perform a copy operation in the history area, but not a cut or paste.)

## Cutting or Copying Information to the Clipboard

When you cut or copy information, that information is placed in the clipboard. However, cutting deletes the information from its current location (used to move information) and copying leaves the information.

1. Highlight the characters that you want to cut or copy.

In the entry line, move the cursor to either side of the characters. Hold 1 and press
(1) or (1) to highlight characters to the left or right of the cursor, respectively.
2. Press F1 and select 4:Cut or 5:Copy.


Note: You can cut, copy or paste without having to use the F1 toolbar menu. Press: $\square \square$ [CUT], $\bullet$ [COPY], or $\bullet$ [PASTE]

Cutting is not the same as deleting. When you delete information, it is not placed in the clipboard and cannot be retrieved.

Note: When you cut or copy information, it replaces the clipboard's previous contents, if any.

## Pasting Information from the Clipboard

A paste operation inserts the contents of the clipboard at the current cursor location on the entry line. This does not change the contents of the clipboard.

1. Position the cursor where you want to paste the information.
2. Press F1 and select 6:Paste, or use the key shortcut:
—— [PASTE]

## Example: Copying and Pasting

Suppose you want to reuse an expression without retyping it each time.

1. Copy the applicable information.
a) Use $\boldsymbol{T}(1)$ or $\boldsymbol{1}(1)$ to highlight the
 expression.
b) Press:

瞱 [COPY]
c) For this example, press ENTER to evaluate the entry.
2. Paste the copied information into a new entry.
a) Begin a new entry and place the cursor where you want to paste the copied information.
b) Press [F3 1 to select the $d$ (differentiate) function.
c) Press:

- [PASTE] to paste the copied expression.

d) Complete the new entry, and press ENTER.

$$
\begin{aligned}
& \text { - solve }\left(x^{4}-3 \cdot x^{3}-6 \cdot x^{2}+8 \cdot b\right. \\
& x=4 \text { or } x=1 \text { or } x=0 \text { or } \\
& -\frac{d}{d x}\left(x^{4}-3 \cdot x^{3}-6 \cdot x^{2}+8 \cdot x\right)
\end{aligned}
$$


Note: You can also reuse an expression by creating a user-defined function.
3. Paste the copied information into a different application.
a) Press $\Delta[\mathrm{Y}=]$ to display the $\mathrm{Y}=$ Editor.
b) Press ENTER to define $\mathbf{y} \mathbf{1}(\mathbf{x})$.
c) Press:
© [PASTE] to paste.

d) Press ENTER to save the new definition.

Note: By copying and pasting, you can easily transfer information from one application to another.

## Reusing a Previous Entry or the Last Answer

You can reuse a previous entry by reexecuting the entry "as is" or by editing the entry and then reexecuting it. You can also reuse the last calculated answer by inserting it into a new expression.

## Reusing the Expression on the Entry Line

When you press ENTER to evaluate an expression, the TI-89 Titanium leaves that expression on the entry line and highlights it. You can type over the entry, or you can reuse it as necessary.

For example, using a variable, find the square of $1,2,3$, etc. As shown below, set the initial variable value and then enter the variable expression. Next, reenter to increment the variable and calculate the square.

| TI-89 Titanium | Display |
| :---: | :---: |
| $0 \text { STO }$ <br> 2nd [a-lock] NUM ENTER |  |
| NUM alpha +1 STO® <br> 2nd [a-lock] NUM <br> 2nd [:] NUM ${ }^{1} 2$ <br> ENTER |  |

## TI-89 Titanium

## Display

ENTER ENTER

| - $9 \rightarrow$ คาแツ |  |  | 0 |
| :---: | :---: | :---: | :---: |
|  |  |  |  |
| - пแm+1-nแm : |  |  |  |
|  |  |  |  |
|  |  |  |  |
| Mill\| | Find illa | FUFIT: | $4 / 30$ |

Note: Reexecuting an entry "as is" is useful for iterative calculations that involve variables.

Using the equation $\mathbf{A}=\pi \mathbf{r}^{2}$, use trial and error to find the radius of a circle that covers 200 square centimeters.

Note: Editing an entry lets you make minor changes without retyping the entire entry.
The example below uses 8 as the first guess and then displays the answer in its approximate floating-point form. You can edit and reexecute using 7.95 and continue until the answer is as accurate as you want.

| TI-89 Titanium | Display |
| :---: | :---: |
| $\begin{aligned} & 8 \text { ST0』 alpha } \mathrm{R} \text { 2nd } \\ & {[:]} \\ & \text { [nd }[\pi] \text { alpha } \mathrm{R} \wedge 2 \\ & \text { ENTER } \end{aligned}$ |  |
| - ENTER |  |


| Tl-89 Titanium | Display |
| :---: | :---: |
|  |  |
|  | $-8+r: \pi \cdot r^{2}$ <br> $-8+r: \pi \cdot r^{2}$ <br> 804.062 |
|  | $\frac{17.95 \rightarrow r: \pi \cdot r^{2} \quad 198.557}{}$ |
|  |  |

Note: When the entry contains a decimal point, the result is automatically displayed in floating-point.

## Recalling a Previous Entry

You can recall any previous entry that is stored in the history area, even if the entry has scrolled off the top of the screen. The recalled entry replaces whatever is currently shown on the entry line. You can then reexecute or edit the recalled entry.

| To recall: | Press: | Effect: |
| :--- | :--- | :--- |
| The last entry <br> (if you've changed <br> the entry line) | 2nd [ENTRY] <br> once | If the last entry is still shown on the <br> entry line, this recalls the entry prior <br> to that. |
| Previous entries | 2nd [ENTRY] <br> repeatedly | Each press recalls the entry prior to <br> the one shown on the entry line. |

Note: You can also use the entry function to recall any previous entry. Refer to entry( ) in the Technical Reference module.

For example:

If the entry line contains the last entry, 2nd [ENTRY] recalls this entry.

If the entry line is edited or cleared, 2nd [ENTRY] recalls this entry.


## Recalling the Last Answer

Each time you evaluate an expression, the TI-89 Titanium stores the answer to the variable ans(1). To insert this variable in the entry line, press [2nd [ANS].

For example, calculate the area of a garden plot that is 1.7 meters by 4.2 meters. Then calculate the yield per square meter if the plot produces a total of 147 tomatoes.

1. Find the area.
1.7 区 4.2 ENTER
2. Find the yield.
$147 \div$ 2nd [ANS] ENTER

| $\square 1.7 .4 .2 \quad 7.14$ |  |  |  |
| :---: | :---: | :---: | :---: |
| 147 20.5882 |  |  |  |
| 7.14 20.5882 |  |  |  |
| 147/ans(1) |  |  |  |
| - MHIN | Find illa | FUNIC | 2 c 30 |
|  |  |  |  |

Variable ans(1) is inserted, and its value is used in the calculation.

Just as ans(1) always contains the last answer, ans(2), ans(3), etc., also contain previous answers. For example, ans(2) contains the next-to-last answer.

Note: Refer to ans( ) in the Technical Reference module.

## Auto-Pasting an Entry or Answer from the History Area

You can select any entry or answer from the history area and "auto-paste" a duplicate of it on the entry line. This lets you insert a previous entry or answer into a new expression without having to retype the previous information.

## Why Use Auto-Paste

The effect of using auto-paste is similar to 2nd [ENTRY] and 2nd [ANS] as described in the previous section, but there are differences.

| For entries: | Pasting lets you: | 2nd [ENTRY] lets you: |
| :---: | :---: | :---: |
|  | Insert any previous entry into the entry line. | Replace the contents of the entry line with any previous entry. |
| For answers: | Pasting lets you: | 2nd [ANS] lets you: |
|  | Insert the displayed value of any previous answer into the entry line. | Insert the variable ans(1), which contains the last answer only. Each time you enter a calculation, ans(1) is updated to the latest answer. |

Note: You can also paste information by using the F1 toolbar menu.

## Auto-Pasting an Entry or Answer

1. On the entry line, place the cursor where you want to insert the entry or answer.
2. Press $\Theta$ to move the cursor up into the history area. This highlights the last answer.
3. Use $\Theta$ and $\Theta$ to highlight the entry or answer to auto-paste.

- $\Theta$ moves from answer to entry up through the history area.
- You can use $\Theta$ to highlight items that have scrolled off the screen


Note: To cancel auto-paste and return to the entry line, press ESC. To view an entry or answer too long for one line (indicated by at the end of the line), use (1) and (1) or 2nd (1) and 2nd (1).
4. Press ENTER.

The highlighted item is inserted in the entry line.


This pastes the entire entry or answer. If you need only a part of the entry or answer, edit the entry line to delete the unwanted parts.

## Creating and Evaluating User-Defined Functions

User-defined functions can be a great time-saver when you need to repeat the same expression (but with different values) multiple times. User-defined functions can also extend your TI-89 Titanium's capabilities beyond the built-in functions.

## Format of a Function

The following examples show user-defined functions with one argument and two arguments. You can use as many arguments as necessary. In these examples, the definition consists of a single expression (or statement).

| cube $(x)=x^{3}$ | $\operatorname{xroot}(x, y)=y^{1 / x}$ |
| :---: | :---: |
| (1) (2) (3) (2) |  |

(1) Function name
(2) Argument list
(3) Definition

When defining functions and programs, use unique names for arguments that will not be used in the arguments for a subsequent function or program call.

Note: Function names follow the same rules as variable names. Refer to "Storing and Recalling Variable Values" in Operating the Calculator.

In the argument list, be sure to use the same arguments that are used in the definition. For example, cube( $\mathbf{n}$ ) $=\mathbf{x}^{3}$ gives unexpected results when you evaluate the function.

Arguments ( $x$ and $y$ in these examples) are placeholders that represent whatever values you pass to the function. They do not represent the variables $x$ and $y$ unless you specifically pass $x$ and $y$ as the arguments when you evaluate the function.

## Creating a User-Defined Function

Use one of the following methods.

| Method | Description |
| :---: | :---: |
| STOD | Store an expression to a function name (including the argument list). |
|  |  |
| Define command | Define a function name (including the argument list) as an expression. |
|  |  |
| Program Editor | Refer to Programming or information on creating a user-defined function. |

## Creating a Multi-Statement Function

You can also create a user-defined function whose definition consists of multiple statements. The definition can include many of the control and decision-making structures (If, Elself, Return, etc.) used in programming.

Note: For information about similarities and differences between functions and programs, refer to Programming.

For example, suppose you want to create a function that sums a series of reciprocals based on an entered integer (n):

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

When creating the definition of a multi-statement function, it may be helpful to visualize it first in a block form.
(1) Func
(2) Local temp,i If fPart (nn) $\neq 0$ or $n n \leq 0$
(3) Return "bad argument" $0 \rightarrow$ temp
(4) For i,nn,1,-1
approx(temp+1/i) $\rightarrow$ temp EndFor
(5) Return temp
(1) EndFunc
(1) Func and EndFunc must begin and end the function.
(2) Variables not in the argument list must be declared as local.
(3) Returns a message if $n n$ is not an integer or if $n n \leq 0$.
(4) Sums the reciprocals.
(5) Returns the sum.

When entering a multi-statement function on the calculator Home screen, you must enter the entire function on a single line. Use the Define command just as you would for a single-statement function.


Note: It's easier to create a complicated multi-statement function in the Program Editor than on the calculator Home screen. Refer to Programming.

## Evaluating a Function

You can use a user-defined function just as you would any other function. Evaluate it by itself or include it in another expression.


## Displaying and Editing a Function Definition

| To: | Do this: |
| :---: | :---: |
| Display a list of all user-defined functions | Press 2nd [VAR-LINK] to display the VAR-LINK screen. You may need to use the F2 View toolbar menu to specify the Function variable type. (Refer to Memory and Variable Management.) <br> - or - <br> Press: <br> CATALOG |

Display a list of Flash Press:
application functions © CATALOG F3

| To: | Do this: |
| :---: | :---: |
| Display the definition of a user-defined function | From the VAR-LINK screen, highlight the function and display the Contents menu. <br> 2nd [F6] <br> - or - <br> From the calculator Home screen, press 2nd [RCL]. Type the function name but not the argument list (such as xroot), and press ENTER twice. <br> - or - <br> From the Program Editor, open the function. (Refer to Programming.) |
| Edit the definition | From the calculator Home screen, use 2nd [RCL] to display the definition. Edit the definition as necessary. Then use STOD or Define to save the new definition. <br> - or - <br> From the Program Editor, open the function, edit it, and save your changes. (Refer to Programming.) |

Note: You can view a user-defined function in the CATALOG dialog box, but you cannot use the CATALOG to view or edit its definition.

## If an Entry or Answer Is "Too Big"

In some cases, an entry or answer may be "too long" and/or "too tall" to be displayed completely in the history area. In other cases, the TI-89 Titanium may not be able to display an answer because there is not enough free memory.

## If an Entry or Answer Is "Too Long"

Move the cursor into the history area, and highlight the entry or answer. Then use the cursor pad to scroll. For example:

- The following shows an answer that is too long for one line.

- The following shows an answer that is both too long and too tall to be displayed on the screen.

Note: This example uses the randMat function to generate a $25 \times 25$ matrix.
(1) Press $\Theta$ or $\boldsymbol{1} \Theta$ to scroll up
(2) Press $\boldsymbol{1} \odot$ to scroll down
(3) Press (1) or 2nd (1) to scroll left
(4) Press (1) or 2nd (1) to scroll right


## If There Is not Enough Memory

A << ...>> symbol is displayed when the TI-89 Titanium does not have enough free memory to display the answer.

For example:

Note: This example uses the seq function to generate a sequential list of integers from 1 to 2500 .

When you see the << ...>> symbol, the answer cannot be displayed even if you highlight it and try to scroll.

In general, you can try to:

- Free up additional memory by deleting unneeded variables and/or Flash applications. Use 2nd [VAR-LINK] as described in Memory and Variable Management.
- If possible, break the problem into smaller parts that can be calculated and displayed with less memory.


## Using the Custom Menu

The TI-89 Titanium has a custom menu that you can turn on and off at any time. You can use the default custom menu or create your own as described in the Programming module.

## Turning the Custom Menu On and Off

When you turn on the custom menu, it replaces the normal toolbar menu. When you turn it off, the normal menu returns. For example, from the calculator Home screen's normal toolbar menu, press 2nd [CATALOG] to toggle the custom menu on and off.


Note: You can also turn the custom menu on and off by entering CustmOn or CustmOff in the entry line and pressing ENTER.

Unless the menu has been modified, the default custom menu appears.

| Menu | Function |
| :--- | :--- |
| F1 Var | Common variable names. |
| F6 $\mathrm{f}(\mathbf{x})$ | Function names such as $\mathrm{f}(\mathrm{x}), \mathrm{g}(\mathrm{x})$, and $\mathrm{f}(\mathrm{x}, \mathrm{y})$. |
| F6 Solve | Items related to solving equations. |
| F4 | Unit |
| F5 Symbol | Common units such as _m,_ft, and _l. |


| Menu | Function |
| :---: | :---: |
| International ［－2nd［F6］ | Commonly accented characters such as è，é，and ê． |
| Tool 2nd［F7］ | ClrHome，NewProb，and CustmOff． |

Note：A custom menu can give you quick access to commonly used items．The Programming module shows you how to create custom menus for the items you use most often．

## Restoring the Default Custom Menu

If a custom menu other than the default is displayed and you want to restore the default：
1．From the calculator Home screen，use 2nd［CATALOG］to turn off the custom menu and display the calculator Home screen＇s normal toolbar menu．

2．Display the Clean Up toolbar menu，and select 3：Restore custom default．
［2nd［F6］

F1T FET
1：Clear シース．．．
2Finewprob custom defent

This pastes the commands used to create the default menu into the entry line．
Note：The previous custom menu is erased．If that menu was created with a program，it can be recreated later by running the program again．

3．Press ENTER to execute the commands and restore the default．

## Finding the Software Version and ID Number

In some situations, you may need to find out information about your TI-89 Titanium, particularly the software version and the unit's ID number.

## Displaying the "About" Screen

1. From either the calculator Home screen or the Apps desktop, press F1 and then select A:About.


Your screen will be different from the one shown to the right.
2. Press ENTER or ESC to close the screen.


## When Do You Need this Information?

The information on the About screen is intended for situations such as:

- If you obtain new or upgraded software or Flash applications for your TI-89 Titanium, you may need to provide your current software version and/or the ID number of your unit.
- If you have difficulties with your TI-89 Titanium and need to contact technical support, knowing the software version may make it easier to diagnose the problem.

The About screen displays the following information about your calculator:

- Hardware version
- OS (Advanced Mathematics Software) version
- Product identifier (Product ID)
- Unit ID
- Apps certificate revision number (Cert. Rev.)

(1) OS version
(2) Product identifier
(3) Apps certificate revision number
(4) Hardware version
(5) Unit ID (required to obtain certificates for installing purchased Apps)

Your screen will be different from the one shown above.

## Symbolic Manipulation

## Using Undefined or Defined Variables

When performing algebraic or calculus operations, it is important that you understand the effect of using undefined and defined variables. Otherwise, you may get a number for a result instead of the algebraic expression that you anticipated.

## How Undefined and Defined Variables Are Treated

When you enter an expression that contains a variable, the TI-89 Titanium treats the variable in one of two ways.

- If the variable is undefined, it is treated as an algebraic symbol.

- If the variable is defined (even if defined as 0 ), its value replaces the variable.


To see why this is important, suppose you want to find the first derivative of $x^{3}$ with respect to $x$.

- If $x$ is undefined, the result is in the form you probably expected.

| - $\frac{d}{d x}\left(x^{3}\right)$ |  |  | $3 \cdot x^{2}$ |
| :---: | :---: | :---: | :---: |
| ( $\left.{ }^{(1)} x^{*} 3, x\right)$ |  |  |  |
| Minl\| | Find illto | FUAC | $1 / 30$ |

- If $x$ is defined, the result may be in a form you did not expect.

Note: When defining a variable, it's a good practice to use more than one character in the name. Leave one-character names undefined for symbolic calculations.


Unless you knew that 5 had been stored to $x$ previously, the answer 75 could be misleading.

## Determining If a Variable Is Undefined

| Method: | Example: |
| :---: | :---: |
| Enter the variable name. | If defined, the variable's value is displayed. <br> If undefined, the variable name is displayed. |
| Use the getType function. | If defined, the variable's type is displayed. <br> If undefined, "NONE" is displayed. |

Note: Use [2nd [VAR-LINK] to view a list of defined variables, as described in Memory and Variable Management.

## Deleting a Defined Variable

You can "undefine" a defined variable by deleting it.

| To delete: | Do this: |
| :---: | :---: |
| One or more specified variables | Use the DelVar function. |
|  |  |
|  | You can also delete variables by using the VAR-LINK screen (2nd [VAR-LINK]) as described in Memory and Variable Management. |

All one-letter variables $(a-z)$ in the From the Home screen current folder.

Clean Up menu, select
1:Clear a-z. You will be prompted to press ENTER to confirm the deletion. module.

Gliar 1-chardrtir uariables a-x in current folder":
Ent 4 =TES

## Temporarily Overriding a Variable

By using the "with" operator ( | ), you can:

- Temporarily override a variable's defined value.

- Temporarily define a value for an undefined variable.


Note: For more information about the | operator, refer to Typing the "With" Operator.
To type the "with" operator ( | ), press:
园 1

## Using Exact, Approximate, and Auto Modes

The Exact/Approx mode settings, which are described briefly in Operating the Handheld, directly affect the precision and accuracy with which the TI-89 Titanium calculates a result. This section describes these mode settings as they relate to symbolic manipulation.

## EXACT Setting

When Exact/Approx = EXACT, the handheld uses exact rational arithmetic with up to 614 digits in the numerator and 614 digits in the denominator. The EXACT setting:

- Transforms irrational numbers to standard forms as much as possible without approximating them. For example, $\sqrt{12}$ transforms to $2 \sqrt{3}$ and $\boldsymbol{\operatorname { l n }}(\mathbf{1 0 0 0})$ transforms to $3 \boldsymbol{\operatorname { l n }}(10)$.
- Converts floating-point numbers to rational numbers. For example, 0.25 transforms to $1 / 4$.

The functions solve, cSolve, zeros, cZeros, factor, $\int$, fMin, and fMax use only exact symbolic algorithms. These functions do not compute approximate solutions in the EXACT setting.

- Some equations, such as $2^{-x}=\mathbf{x}$, have solutions that cannot all be finitely represented in terms of the functions and operators on the handheld.
- With this kind of equation, EXACT will not compute approximate solutions. For example, $2^{-x}=\mathbf{x}$ has an approximate solution $\mathbf{x} \approx \mathbf{0 . 6 4 1 1 8 6}$, but it is not displayed in the EXACT setting.

| Advantages | Disadvantages |
| :--- | :--- |
| Results are exact. | As you use more complicated rational <br> numbers and irrational constants, calculations <br> can: |
|  | - $\quad$ Use more memory, which may exhaust the |
|  | memory before a solution is completed. |
|  | - $\quad$ Take more computing time. |

## APPROXIMATE Setting

When Exact/Approx = APPROXIMATE, the handheld converts rational numbers and irrational constants to floating-point. However, there are exceptions:

- Certain built-in functions that expect one of their arguments to be an integer will convert that number to an integer if possible. For example: $d(\mathbf{y}(\mathbf{x}), \mathbf{x}, \mathbf{2 . 0})$ transforms to $d(\mathbf{y}(\mathbf{x}), \mathbf{x}, \mathbf{2})$.
- Whole-number floating-point exponents are converted to integers. For example: $x^{2.0}$ transforms to $x^{2}$ even in the APPROXIMATE setting.

Functions such as solve and $\int$ (integrate) can use both exact symbolic and approximate numeric techniques. These functions skip all or some of their exact symbolic techniques in the APPROXIMATE setting.

| Advantages | Disadvantages |
| :--- | :--- |
| If exact results are not needed, this | Results with undefined variables or |
| might save time and/or use less | functions often exhibit incomplete |
| memory than the EXACT setting. | cancellation. For example, a |
| Approximate results are sometimes | coefficient that should be $\mathbf{0}$ might be |
| more compact and comprehensible | displayed as a small magnitude such |
| than exact results. | as $\mathbf{1 . 2 3 4 5 7 E - 1 1 .}$ |


| Advantages | Disadvantages |
| :--- | :--- |
| If you do not plan to use symbolic | Symbolic operations such as limits |
| computations, approximate results | and integration are less likely to give |
| are similar to familiar, traditional | satisfying results in the |
| numeric calculators. | APPROXIMATE setting. |
|  | Approximate results are sometimes |
|  | less compact and comprehensible |
|  | than exact results. For example, you |
|  | may prefer to see $1 / 7$ instead of |
|  | .142857. |

## AUTO Setting

When Exact/Approx = AUTO, the handheld uses exact rational arithmetic wherever all of the operands are rational numbers. Otherwise, floating-point arithmetic is used after converting any rational operands to floating-point. In other words, floating-point is "infectious." For example:

1/2-1/3 transforms to $1 / 6$
but
$0.5-1 / 3$ transforms to .16666666666667
This floating-point infection does not leap over barriers such as undefined variables or between elements of lists or matrices. For example:

$$
\begin{aligned}
& (1 / 2-1 / 3) x+(0.5-1 / 3) y \text { transforms to } x / 6+.16666666666667 y \\
& \text { and } \\
& \{1 / 2-1 / 3,0.5-1 / 3\} \text { transforms to }\{1 / 6, .16666666666667\}
\end{aligned}
$$

In the AUTO setting, functions such as solve determine as many solutions as possible exactly, and then use approximate numerical methods if necessary to determine additional solutions. Similarly, ] (integrate) uses approximate numerical methods if appropriate where exact symbolic methods fail.

## Advantages

## Disadvantages

You see exact results when practical, and approximate numeric results when exact results are impractical. You can often control the format of a result by choosing to enter some coefficients as either rational or floating-point numbers.

If you are interested only in exact results, some time may be wasted seeking approximate results. If you are interested only in approximate results, some time may be wasted seeking exact results. Moreover, you might exhaust the memory seeking those exact results.

## Automatic Simplification

When you type an expression on the entry line and press ENTER, the TI-89 Titanium automatically simplifies the expression according to its default simplification rules.

## Default Simplification Rules

All of the following rules are applied automatically. You do not see intermediate results.

- If a variable has a defined value, that value replaces the variable.

If the variable is defined in terms of another variable, the variable is replaced with its "lowest level" value (called infinite lookup).


Default simplification does not modify variables that use path names to indicate a folder. For example, $x+c l a s s \mid x$ does not simplify to $2 x$.

Note: For information about folders, refer to the Calculator Home Screen module.

- For functions:
- The arguments are simplified. (Some built-in functions delay simplification of some of their arguments.)
- If the function is a built-in or user-defined function, the function definition is applied to the simplified arguments. Then the functional form is replaced with this result.
- Numeric subexpressions are combined.
- Products and sums are sorted into order.


Products and sums involving undefined variables are sorted according to the first letter of the variable name.

- Undefined variables $r$ through $z$ are assumed to be true variables, and are placed in alphabetical order at the beginning of a sum.
- Undefined variables a through $q$ are assumed to represent constants, and are placed in alphabetical order at the end of a sum (but before numbers).
- Similar factors and similar terms are collected.

- Identities involving zeros and ones are exploited.


This floating-point number causes numeric results to be shown as floating-point.

If a floating-point whole number is entered as an exponent, it is treated as an integer (and
 does not produce a floating-point result).

- Polynomial greatest common divisors are canceled.

- Polynomials are expanded unless no key cancellation can occur.

- Common denominators are formed unless no key cancellation can occur.
- Functional identities are exploited. For example:
$\ln (2 x)=\ln (2)+\ln (x)$ and
$\boldsymbol{\operatorname { s i n }}(\mathrm{x})^{2}+\boldsymbol{\operatorname { c o s }}(\mathrm{x})^{2}=\mathbf{1}$

| $\cdot \frac{2 \cdot x}{x^{2}-1}-\frac{1}{x-1}$ | $\frac{1}{x+1}$ |
| :---: | :---: |
| - $\frac{1}{x}+\frac{1}{4}$ | $\frac{1}{x}+\frac{1}{4}$ |
|  |  |

No key cancellation


## How Long Is the Simplification Process?

Depending on the complexity of an entry, result, or intermediate expression, it can take a long time to expand an expression and cancel common divisors as necessary for simplification.

To interrupt a simplification process that is taking too long, press ON. You can then try simplifying only a portion of the expression. (Auto-paste the entire expression on the entry line, and then delete the unwanted parts.)

## Delayed Simplification for Certain Built-In Functions

Usually, variables are automatically simplified to their lowest possible level before they are passed to a function. For certain functions, however, complete simplification is delayed until after the function is performed.

## Functions that Use Delayed Simplification

Functions that use delayed simplification have a required var argument that performs the function with respect to a variable. These functions have at least two arguments with the general form:

## function(expression, var [, ... ])

Note: Not all functions that use a var argument use delayed simplification.

For example: solve( $\mathbf{x}^{\wedge} \mathbf{2 - x}-\mathbf{2 = 0 , x}$ )

$$
\begin{aligned}
& d\left(\mathbf{x}^{\wedge} 2-\mathbf{x}-2, \mathbf{x}\right) \\
& \int\left(\mathbf{x}^{\wedge}-\mathbf{x}-2, \mathbf{x}\right) \\
& \operatorname{limit}\left(\mathbf{x}^{2}-\mathbf{x}-\mathbf{2}, \mathbf{x}, 5\right)
\end{aligned}
$$

For a function that uses delayed simplification:

1. The var variable is simplified to the lowest level at which it remains a variable (even if it could be further simplified to a non-variable value).
2. The function is performed using the variable.
3. If $v a r$ can be further simplified, that value is then substituted into the result.

Note: You may or may not want to define a numeric value for var, depending on the situation.

For example:
x cannot be simplified.

| - DelUar x - $\frac{d}{d x}\left(x^{3}\right)$ |  | $\begin{aligned} & \text { Lone } \\ & 3 \cdot x^{2} \end{aligned}$ |
| :---: | :---: | :---: |
|  |  |  |
|  | FUNFC | $2{ }^{2} 30$ |


| x is not simplified. The function uses $\mathrm{x}^{3}$, |
| :--- | :--- | :--- | :--- |
| and then substitutes 5 for x. |

Note: The example to the right finds the derivative of $x^{3}$ at $x=5$. If $x^{3}$ was initially simplified to 75 , you would find the derivative of 75 , which is not what you want.


## Substituting Values and Setting Constraints

The "with" operator ( | ) lets you temporarily substitute values into an expression or specify domain constraints.

## Typing the "With" Operator

To type the "with" operator ( | ), press:
瞱
$\square$

## Substituting for a Variable

For every occurrence of a specified variable, you can substitute a numeric value or an expression.

| -( $(x+2)^{2} \mid x=1$ |  |
| :---: | :---: |
| - $\pi \cdot r^{2} \mid r=5$ | 25 |
| - $\left.\frac{d}{d x}\left(x^{3}\right)^{3} \right\rvert\, x=5$ | 75 |
|  |  |

First derivative of $\mathrm{x}^{3}$ at $x=5$

| - $(\mathrm{x}+2)^{2} \mid \mathrm{x}=\mathrm{a}+1$ | $(a+3)^{2}$ |
| :---: | :---: |
|  |  |

To substitute for multiple variables at the same time, use the Boolean and operator.


## Substituting for a Simple Expression

For every occurrence of a simple expression, you can substitute a variable, numeric value, or another expression.


By replacing a commonly used (or long) term, you can display results in a more compact form.

Note: $\operatorname{acos}(\mathbf{x})$ is different from $\mathbf{a}^{*} \cos (\mathbf{x})$.

## Substituting Complex Values

You can substitute complex values just as you would for other values.

All undefined variables are treated as real numbers in symbolic calculations. To perform complex symbolic analysis, you must define a complex variable. For example:

$$
x+y i \geqslant z
$$

Then you can use $z$ as a complex variable. You can also use $z$ _. For more information see the _ (underscore) topic in the Technical Reference module.

## Note:

- For an overview of complex numbers, refer to the Technical Reference module.
- To get the complex $i$, press 2nd [i]. Do not simply type the letter $\mathbf{i}$ on the keyboard.


## Be Aware of the Limitations of Substitutions

- Substitution occurs only where there is an exact match for the substitution.

$$
\begin{aligned}
& \text { Only } x^{2} \text { was } \\
& \text { replaced, not } x^{4} \text {. } \\
& -x^{4}+3 \cdot x^{2} \mid x^{2}=y \\
& -x^{4}+3 \cdot x^{2} \mid x=y^{1 / 2} \\
& \frac{y^{2}+3 \cdot y}{x^{\wedge} 4+3 x^{\wedge} 21 x=y^{\wedge}(1 / 2)}
\end{aligned}
$$

Define the
substitution in
simpler terms for
a more complete
substitution.

- Infinite recursions can occur when you define a substitution variable in terms of itself.


When you enter a substitution that causes an infinite recursion:

- An error message is displayed.

- When you press ESC, an error is shown in the history area.

- Internally, an expression is sorted according to the automatic simplification rules. Therefore, products and sums may not match the order in which you entered them.
- As a general rule, you should substitute for a single variable.

| $\begin{aligned} & - \text { solve }\left(m \cdot c^{2}=e, m\right) \quad m=\frac{e}{c^{2}} \\ & -\sin \left(2 \cdot m \cdot c^{2}\right) \left\lvert\, m=\frac{e}{c^{2}}\right. \end{aligned}$$\sin (\gamma \cdot e)$ |  |
| :---: | :---: |
|  |  |
| sin $\left.2 * m * c^{\circ} 2\right) 1 \mathrm{~m}=\mathrm{e} / \mathrm{c}^{\circ} 2$ |  |

- Substituting for more general expressions (either $\boldsymbol{m} \cdot \mathbf{c}^{2}=\mathbf{e}$ or $\mathbf{c}^{2} \cdot \mathbf{m}=\mathrm{e}$ ) may not work as you anticipate.

Note: Use the solve function to help determine the single-variable substitution.


## Specifying Domain Constraints

Many identities and transformations are valid for only a particular domain. For example:
$\ln (x * y)=\ln (x)+\ln (y)$
$\operatorname{Sin}^{-1}(\boldsymbol{\operatorname { s i n }}(\theta))=\theta$
only if $\mathbf{x}$ and/or $\mathbf{y}$ is not negative
only if $\theta \geq-\pi / 2$ and $\theta$
$\leq \pi / 2$ radians

Use the "with" operator to specify the domain constraint.

Because $\ln (x * y)=\ln (x)+\ln (y)$ is not always valid, the logarithms are not combined.

With a constraint, the identity is valid and the

$\left.$| $-\ln (x \cdot y)-\ln (x)$ |
| :--- |
| $\ln (x \cdot y)-\ln (x)$ |
| $-\ln (x \cdot y)-\ln (x) \mid x>0$ | $\ln (y) \right\rvert\,$ expression is simplified.

Note: Enter $\operatorname{In}(\mathbf{x} * \mathbf{y})$ instead of $\operatorname{In}(\mathbf{x y})$; otherwise, xy is interpreted as a single variable named xy.

Because $\sin ^{-1}(\sin (\theta))=\theta$ is not always valid, the expression is not simplified.

With a constraint, the expression can be simplified.
$-\sin ^{-1}\left(\sin (\theta) \quad \sin ^{-1}(\sin (\theta)\right.$
$-\sin -\sin (\theta) \left\lvert\, \theta \geq \frac{-\pi}{2}\right.$ and $\theta$.


Note: For $\geq$ or $\leq$, press $\rightarrow$ [ $>$ ] or [ $\rightarrow$ ]. You can also use 2nd [MATH] 8 or 2nd [CHAR] 2 to select them from a menu.

## Using Substitutions vs. Defining a Variable

In many cases, you can achieve the same effect as a substitution by defining the variable.

| $-(x+2)^{2} \mid x=1$ | 9 |
| :--- | ---: |
| $-1+x$ | 1 |
| $-(x+2)^{2}$ | 9 |
| $(x+2)^{\wedge} 2$ |  |
| MAIIN ALTD | FUNC |

However, substitution is preferable for most cases because the variable is defined only for the current calculation and does not accidentally affect later calculations.

Substituting $\mathrm{x}=1$ does not affect the next calculation.


Storing $1 \rightarrow x$ affects the subsequent calculations.calculation.


Caution: After x is defined, it can affect all calculations that involve x (until you delete x ).

## Overview of the Algebra Menu

You can use the F2 Algebra toolbar menu to select the most commonly used algebraic functions.

## The Algebra Menu

From the Home screen, press F2 to display:

This menu is also available from the MATH menu. Press 2nd [MATH] and then select
9:Algebra.

Note: For a complete description of each function and its syntax, refer to the Technical Reference module.

| Menu Item | Description |
| :--- | :--- |
| solve | Solves an equation for a specified variable. This returns <br> real solutions only, regardless of the Complex Format <br> mode setting. Displays answers with "and" and "or" <br> connecting solutions. (For complex solutions, select |
|  | A:Complex from the Algebra menu.) |


| Menu Item | Description |
| :---: | :---: |
| Trig | Displays the submenu: |
|  | 1: texperich <br> 2:tcollecte |
|  | tExpand - Expands trig expressions with angle sums and multiple angles. |
|  | TCollect - Collects the products of integer powers of trig functions into angle sums and multiple angles. tCollect is the opposite of tExpand. |
| Complex | Displays the submenu: |
|  | 1.cgoluec <br> $3: \operatorname{sengog}$ |
|  | These are the same as solve, factor, and zeros; but they also compute complex results. |
| Extract | Displays the submenu: |
|  |  |
|  | getNum - Applies comDenom and then returns the resulting numerator. |
|  | getDenom - Applies comDenom and then returns the resulting denominator. |
|  | left - Returns the left-hand side of an equation or inequality. |

right - Returns the right-hand side of an equation or inequality.

Note: The left and right functions are also used to return a specified number of elements or characters from the left or right side of a list or character string.

## Common Algebraic Operations

This section gives examples for some of the functions available from the F2 Algebra toolbar menu. For complete information about any function, refer to the Technical Reference module. Some algebraic operations do not require a special function.

## Adding or Dividing Polynomials

You can add or divide polynomials directly, without using a special function.


| $\frac{x^{2}+5 \cdot x+6}{x+2}$ |  | $x+3$ |
| :---: | :---: | :---: |
| ( $\left.x^{*} 2+5 x+6\right) /(x+2)$ |  |  |
| Mill\| | FUPIC | 1, 30 |

## Factoring and Expanding Polynomials

Use the factor (F2 2) and expand ( F 2 3 3 ) functions.

expand(expression $[, v a r])$
for partial expansion with respect to a variable

Factor $x^{5}-1$. Then expand the result.
Notice that factor and expand perform opposite operations.


## Finding Prime Factors of a Number

The factor ([22 2) function lets you do more than simply factor an algebraic polynomial.
You can find prime factors of a rational number (either an integer or a ratio of integers).

| factor(1729) <br> factor $\left(\frac{21475}{1548}\right)$ | $\begin{gathered} 7 \cdot 13 \cdot 19 \\ 5^{2} \cdot 859 \\ \hline 2^{2} \cdot 3^{2} \cdot 43 \end{gathered}$ |  |
| :---: | :---: | :---: |
|  |  |  |
| factor (21475/1548) |  |  |
| Millw Eifl illa | FUNC |  |

## Finding Partial Expansions

With the expand (F2 3) function's optional var value, you can do a partial expansion that collects similar powers of a variable.

Do a full expansion of $\left(x^{2}-\mathbf{x}\right)\left(\mathbf{y}^{2}-\mathbf{y}\right)$ with respect to all variables.

Then do a partial expansion with respect to x .

- expand ( $\left.\left(x^{2}-x\right)\left(y^{2}-y\right)\right)$
$x^{2} \cdot y^{2}-x^{2} \cdot y-x \cdot y^{2}+x \cdot y$
- expand ( $\left.\left(x^{2}-x\right)\left(y^{2}-y\right), x\right)$
$x^{2} \cdot y \cdot(y-1)-x \cdot y \cdot(y-1)$



## Solving an Equation

Use the solve (ㅍ2 1) function to solve an equation for a specified variable.
solve(equation, var)

Solve $\mathbf{x + y} \mathbf{~ - ~} \mathbf{5}=\mathbf{2 x} \mathbf{- 5} \mathbf{y}$ for $\mathbf{x}$.
Notice that solve displays only the final result.


To see intermediate results, you can manually solve the equation step-by-step.


Note: An operation such as $\square 2$ 区 subtracts $2 x$ from both sides.

## Solving a System of Linear Equations

Consider a set of two equations with two
unknowns:

$$
\begin{aligned}
& 2 x-3 y=4 \\
& -x+7 y=-12
\end{aligned}
$$

To solve this system of equations, use any of the following methods.

| Method | Example |
| :--- | :--- |
| Use the solve function for a one-step <br> solution. | solve $(2 x-3 y=4$ and $-x+7 y=-12,\{x, y\})$ |
| Use the solve function with <br> substitution ( $\mid$ ) for step-by-step <br> manipulation. | Substitutions are in the form of an <br> equality, such as $x=3$ or $y=\sin (x)$. To <br> be most effective, the left side should <br> be a simple variable. <br>  <br>  <br>  <br>  <br>  <br>  <br> See "Symbolic Manipulation" in the <br> Previews chapter, which solved for <br> $\mathbf{x = - 8 / 1 1}$ and $\mathbf{y}=-\mathbf{2 0 / 1 1}$. |
|  |  |

Method Example

Use the simult function with a matrix.

Enter the coefficients as a matrix and the results as a constant column matrix.

Use the rref function with a matrix. Enter the coefficients as an augmented matrix.


Note: The simult and rref matrix functions are not on the F2 Algebra menu. Use 2nd [MATH] 4 or the Catalog.

## Finding the Zeros of an Expression

Use the zeros (ㅌ24) 4) function.

## zeros(expression, var)

Use the expression $\mathbf{x}$
$\boldsymbol{\operatorname { s i n }}(\mathrm{x})+\boldsymbol{\operatorname { c o s }}(\mathrm{x})$.
Find the zeros with respect to $x$ in the interval $\mathbf{0} \leq \mathrm{x}$ and $\mathrm{x} \leq \mathbf{3}$.


Note: For $\geq$ or $\leq$, type [ $\quad$ ] or [<]. You can also use 2nd [MATH] 8 or 2nd [CHAR] 2 to select them from a menu.

## Finding Proper Fractions and Common Denominators

## 

propFrac(rational expression [,var])
_for proper fractions with respect to a variable
comDenom(expression [,var])
-for common denominators that collect similar powers of this variable

Find a proper fraction for the expression $\left(x^{4}-2 x^{2}+x\right) /\left(2 x^{2}+x+4\right)$.

Then transform the answer into a ratio of a fully expanded numerator and a fully expanded denominator.

Notice that propFrac and comDenom perform opposite operations.

Note: You can use comDenom with an expression, list, or matrix.

$$
\begin{aligned}
& \text { - PropFras }\left(\frac{x^{4}-2 \cdot x^{2}+x}{2 \cdot x^{2}+x+4}\right) \\
& \frac{31 \cdot x+60}{8 \cdot\left(2 \cdot x^{2}+x+4\right)}+\frac{x^{2}}{2}-\frac{x}{4} \rightarrow \\
& \text { - comDenom }\left[\frac{31 \cdot x+60}{8 \cdot\left(2 \cdot x^{2}+x+4\right)}\right. \text {, } \\
& \frac{\frac{x^{4}-2 \cdot x^{2}+x}{2 \cdot x^{2}+x+4}}{\frac{\left.+x+4)+\left(x^{\wedge} 2 / 2-x / 4\right)-15 / 8\right)}{(1)}}
\end{aligned}
$$

> If you do this example on your handheld, the propFrac function
> scrolls off the top of the screen.

In this example:

- $\frac{31 x+60}{8}$ is the remainder of $\mathbf{x}^{4}-\mathbf{2} \mathbf{x}^{2}+\mathbf{x}$ divided by $\mathbf{2} \mathbf{x}^{2}+\mathbf{x}+\mathbf{4}$.
- $\frac{x^{2}}{2}-\frac{x}{4}-15 / 8$ is the quotient.


## Overview of the Calc Menu

You can use the F3 Calc toolbar menu to select commonly used calculus functions.

## The Calc Menu

From the Home screen, press F3 to display:


Note: For a complete description of each function and its syntax, refer to the Technical Reference module.

| Menu Item | Description |
| :--- | :--- |
| $d$ differentiate | Differentiates an expression with respect to a specified <br> variable. |


| fintegrate | Integrates an expression with respect to a specified <br> variable. |
| :--- | :--- |
| limit | Calculates the limit of an expression with respect to a <br> specified variable. |
| $\Sigma$ sum | Evaluates an expression at discrete variable values <br> within a range and then calculates the sum. |
| П product | Evaluates an expression at discrete variable values <br> within a range and then calculates the product. |
| fMin | Finds candidate values of a specified variable that <br> minimize an expression. |
| fMax | Finds candidate values of a specified variable that <br> maximize an expression. |
| arcLen | Returns the arc length of an expression with respect to <br> a specified variable. |
| taylor | Calculates a Taylor polynomial approximation to an <br> expression with respect to a specified variable. |
| nDeriv | Calculates the numerical derivative of an expression <br> with respect to a specified variable. |
| nInt | Calculates an integral as a floating-point number using <br> quadrature (an approximation using weighted sums of <br> integrand values). |
| deSolve | Symbolically solves many 1st and 2nd order differential <br> equations, with or without initial conditions. |

Note: The $d$ symbol for differentiate is a special symbol. It is not the same as typing the letter D on the keyboard. Use F3 1 or 2nd [d].

## Common Calculus Operations

This section gives examples for some of the functions available from the F3 Calc toolbar menu. For complete information about any calculus function, refer to the Technical Reference module.

## Integrating and Differentiating

Use the $\int$ integrate (F3 2) and $d$ differentiate ( $\mathbb{F 3} 1$ 1) functions.
$\int$ (expression, var $\left.[, l o w][, u p]\right)$
L_ lets you specify limits or a constant of integration
$d$ (expression, var [,order])
Integrate $\mathbf{x}^{2} * \sin (x)$ with respect to $x$.
Differentiate the answer with respect to x .

| - $\int x^{2} \cdot \sin (x) d x$ <br> $\left(2-x^{2}\right) \cdot \cos (x)+2 \times \sin (\infty)$ <br> - $\frac{d}{d x}\left[\left(\left[2-x^{2}\right) \cdot \cos (x)+2 \cdot x \cdot \equiv p\right.\right.$ |  |
| :---: | :---: |
| $\frac{x^{2} \cdot \sin (x)}{(11), x)}$ |  |
|  |  |
| To get $d$, use F3 1 or 2nd [d]. Do not |  |
|  |  |
| simply type the letter |  |
| the ke | ar |

Note: You can integrate an expression only; you can differentiate an expression, list, or matrix.

## Finding a Limit

Use the limit (ㅌ3 3) function.
limit(expression, var, point [,direction])
_negative number = from left positive number= from right omitted number or $0=$ both

Find the limit of $\boldsymbol{\operatorname { s i n }}(\mathbf{3 x}) / \mathbf{x}$ as x approaches 0 .


Note: You can find a limit for an expression, list, or matrix.

## Finding a Taylor Polynomial

Use the taylor (ㅍ3 9) function.
taylor(expression, var, order [point]) if omitted, expansion point is 0

Find a 6th order Taylor polynomial for $\boldsymbol{\operatorname { s i n }}(\mathbf{x})$ with respect to $x$.

Store the answer as a user-defined function named $\mathbf{y 1}(\mathbf{x})$.

Then graph $\boldsymbol{\operatorname { s i n }}(\mathbf{x})$ and the Taylor polynomial.


$$
\begin{aligned}
& \text { Graph } \sin (x): G r a p h \\
& \mathrm{y} 1(\mathrm{x})
\end{aligned}
$$



Important: Degree-mode scaling by $\pi / 180$ may cause calculus application results to appear in a different form.

## User-Defined Functions and Symbolic Manipulation

You can use a user-defined function as an argument for the TI-89 Titanium 's built-in algebra and calculus functions.

## For Information about Creating a User-Defined Function

## Refer to:

- "Creating and Evaluating User-Defined Functions" in the Calculator Home Screen module.
- "Graphing a Function Defined on the Home Screen" and "Graphing a Piecewise Defined Function" in the Calculator Home Screen module.
- "Overview of Entering a Function" in the Programming module.


## Undefined Functions

You can use functions such as $\mathbf{f}(\mathbf{x}), \mathbf{g}(\mathbf{t}), \mathbf{r}(\theta)$, etc., that have not been assigned a definition. These "undefined" functions yield symbolic results. For example:

Use DelVar to ensure that $\mathbf{f}(\mathbf{x})$ and $\mathbf{g}(\mathbf{x})$ are not defined.

Then find the derivative of $\mathbf{f}(\mathbf{x}) \mathbf{g}(\mathbf{x})$ with
 respect to x .

Note: To select $\mathbf{d}$ from the Calc toolbar menu, press F3 1 (or press 2nd [ $d$ ] on the keyboard).

## Single-Statement Functions

You can use user-defined functions consisting of a single expression. For example:

- Use STO® to create a user-defined secant function, where:

$$
\sec x=\frac{1}{\cos x}
$$

Then find the limit of $\boldsymbol{\operatorname { s e c } ( \mathbf { x } )}$ as $x$ approaches $\pi / 4$.

| $\frac{1}{\cos (x)}+\sec (x)$ | Done <br> $\sqrt{2}$ |
| :---: | :---: |
| $\begin{aligned} & \quad \lim \sec (x) \\ & x \rightarrow \frac{\pi}{4} \end{aligned}$ |  |
| 1imit(seec (x), $x, \pi / 4$ ) |  |
|  | 2 C 30 |

Note: To select limit from the Calc toolbar menu, press F3 3.

- Use Define to create a user-defined function $\mathbf{h}(\mathbf{x})$, where:

$$
h(x)=\int_{0}^{x} \frac{\sin t}{t}
$$

Then find a 5th order Taylor polynomial for $\mathbf{h}(\mathbf{x})$ with respect to $x$.
Note: To select $\int$ from the Calc toolbar
 menu, press F3 2 (or press [2nd [ J ] on the keyboard). To select taylor, press F3 9.

## Multi-Statement vs. Single-Statement Functions

Multi-statement user-defined functions should be used as an argument for numeric functions (such as nDeriv and nint) only.

In some cases, you may be able to create an equivalent single-statement function. For example, consider a piecewise function with two pieces.

| When: | Use expression: |
| :--- | :--- |
| $\mathrm{x}<0$ <br> $\mathrm{x} \geq 0$ | -x |

- If you were to create a multi-statement user-defined function with the form:

```
Func
    If x<0 Then
    Return ëx
Else
    Return 5cos(x)
    EndIf
    EndFunc
```

Define
$y 1(x)=$ Func: If $x<0$
Then: ... :EndFunc


Then numerically integrate $\mathbf{y 1}(\mathbf{x})$ with respect to $x$.

Note: To select nInt from the Calc toolbar menu, press F3 B:nInt.

- Create an equivalent single-statement user-defined function.

Use the TI-89 Titanium's built-in when function.

Then integrate $\mathbf{y} 1(\mathrm{x})$ with respect to x .
Note: To select $\int$ from the Calc toolbar menu, press [F3 2 (or press [2nd [ s ] on the keyboard).


## If You Get an Out-of-Memory Error

The TI-89 Titanium stores intermediate results in memory and then deletes them when the calculation is complete. Depending on the complexity of the calculation, the handheld may run out of memory before a result can be calculated.

## Freeing Up Memory

- Delete unneeded variables and/or Flash applications, particularly large-sized ones.
- Use 2nd [VAR-LINK] as described in Memory and Variable Management to view and delete variables and/or Flash applications.
- On the Home screen:
- Clear the history area (F1 8) or delete unneeded history pairs.
- You can also use F1 9 to reduce the number of history pairs that will be saved.
- Use MODE to set Exact/Approx = APPROXIMATE. (For results that have a large number of digits, this uses less memory than AUTO or EXACT. For results that have only a few digits, this uses more memory.)


## Simplifying Problems

- Split the problem into parts.
- Split solve( $a * b=0, v a r$ ) into solve( $a=0, v a r$ ) and solve( $b=0, v a r)$. Solve each part and combine the results.
- If several undefined variables occur only in a certain combination, replace that combination with a single variable.
- If $m$ and $c$ occur only as $\mathbf{m} * \mathbf{c}^{2}$, substitute e for $\mathbf{m} * \mathbf{c}^{2}$.
- In the expression $\frac{(a+b)^{2}+\sqrt{(a+b)^{2}}}{1-(a+b)^{2}}$, substitute $\mathbf{c}$ for (a+b) and use $\frac{c^{2}+\sqrt{c^{2}}}{1-c^{2}}$. In the solution, replace $c$ with $(\mathbf{a}+\mathbf{b})$.
- For expressions combined over a common denominator, replace sums in denominators with unique new undefined variables.
- In the expression $\frac{x}{\sqrt{a^{2}+b^{2}}+c}+\frac{y}{\sqrt{a^{2}+b^{2}}+c}$, substitute d for $\sqrt{a^{2}+b^{2}}+c$ and are $\frac{x}{d}+\frac{y}{d}$. In the solution, replace d with $\sqrt{a^{2}+b^{2}}+c$.
- Substitute known numeric values for undefined variables at an earlier stage, particularly if they are simple integers or fractions.
- Reformulate a problem to avoid fractional powers.
- Omit relatively small terms to find an approximation.


## Special Constants Used in Symbolic Manipulation

The result of a calculation may include one of the special constants described in this section. In some cases, you may also need to enter a constant as part of your entry.

true, false

These indicate the result of an identity or a Boolean expression.


This notation indicates an "arbitrary integer" that represents any integer.

When an arbitrary integer occurs multiple times in the same session, each occurrence is numbered consecutively. After it reaches 255, arbitrary integer consecutive numbering restarts at @n0. Use Clean Up 2:NewProb to reset to @n1.

Note: For @, press:

- STOD
$\infty, \boldsymbol{e}$
$\infty$ represents infinity, and e represents the constant 2.71828... (base of the natural logarithms).

These constants are often used in entries as well as results.


## Notes:

For $\infty$, press:

- $\quad[\infty]$

For $e$, press:
瞱 $\quad[\mathrm{e} x]$

## undef

This indicates that the result is undefined.

Mathematically undefined
$\pm \infty$ (undetermined sign)


Non-unique limit

## Constants and Measurement Units

## Entering Constants or Units

You can use a menu to select from a list of available constants and units, or you can type them directly from the keyboard.

## From a Menu

The following shows how to select a unit, but you can use the same general procedure to select a constant.

From the Home screen:

1. Type the value or expression.

2. Use $\odot$ and $\Theta$ to move the cursor to the applicable category.

Note: Use $2 n d \Theta$ and $2 n d$ © to scroll one page at a time through the categories.

4. To select the highlighted (default) unit, press ENTER.

- or -

To select a different unit from the category, press (1). Then highlight the applicable unit, and press ENTER.
Note: If you created a user-defined unit for an existing category, it is listed in the menu.
 the cursor by typing the first letter of a unit.

The selected unit is placed in the entry line. Constant and unit names always begin with
 an underscore ( _ ).

## From the Keyboard

If you know the abbreviation that the TI-89 Titanium uses for a particular constant or unit, you can type it directly from the keyboard. For example:

## 256_m

- The first character must be an underscore ( _ ). For _, press:
- [-]
- A space or a multiplication symbol (*) before the underscore is optional. For example, 256_m, 256 _m, and 256*_m are equivalent.
- However, if you are adding units to a variable, you must put a space or $*$ before the underscore. For example, $x_{-} m$ is treated as a variable, not as $x$ with a unit.

Note: You can type units in either uppercase or lowercase characters.

## Combining Multiple Units

You may need to combine two or more units from different categories.
For example, suppose you want to enter a velocity in meters per second. In the UNITS dialog box, however, the Velocity category does not contain this unit.

You can enter meters per second by combining _m and _s from the Length and Time categories, respectively.


3*9.8_m/_s
Combine the units _m and _s. There is no pre-defined $\mathrm{m} / \mathrm{s}$ unit.

Note: Create a user-defined unit for frequently used combinations.

## Using Parentheses with Units in a Calculation

In a calculation, you may need to use parentheses ( ) to group a value and its units so that they are evaluated properly. This is particularly true for division problems. For example:

| To calculate: | Enter: |
| :---: | :---: |
| $\frac{100 \_\mathrm{m}}{2 \_\mathrm{s}}$ | for (2_s). This is important for division. <br> If you omit the parentheses, you will get unexpected units. For example: 100_m/2_s 50. *_m *_s |

Note: If you have any doubt about how a value and its units will be evaluated, group them within parentheses ( ).

Here's why you get unexpected units if you do not use parentheses. In a calculation, a unit is treated similar to a variable. For example: $\mathbf{1 0 0}^{\mathbf{m}}$ is treated as $\mathbf{1 0 0}$ __m and 2_s is treated as 2*_s. Without parentheses, the entry is calculated as:


## Converting from One Unit to Another

You can convert from one unit to another in the same category, including any userdefined units.

## For All Units Except Temperature

If you use a unit in a calculation, it is converted and displayed automatically in the current default unit for that category, unless you use the conversion operator as described later. The following examples assume that your default units are set to the SI system of metric units.

## Notes:

- Refer to the list of pre-defined units.
- From the UNITS dialog box, you can select available units from a menu.

To multiply 20 times 6 kilometers.
$20 * 6 \_k m$

| - $20 \cdot 6 \cdot \mathrm{~km}$ |  | 1200 | - |
| :---: | :---: | :---: | :---: |
| 20\%6_km |  |  |  |
| Mill\| | Fifl ituta | FUNE | $1 / 30$ |

Shown in the default unit for Length, (_m in SI system).

If you want to convert to a unit other than the default, use the conversion operator.

| L- For $\downarrow$, press 2nd [ $¢ \square$. |
| :---: |
|  |  |
|  |  |

To convert 4 light years to kilometers:
4_Ityr
To convert 186000 miles/second to
 kilometers/hour:

186000_mi/_s _ _km/_hr

If an expression uses a combination of units, you can specify a conversion for some of the units only. Any units for which you do not specify a conversion will be displayed according to your defaults.

To convert 186000 miles/second from miles to kilometers:

```
186000_mi/_s _km
```

To convert 186000 miles/second from seconds to hours:
186000_mi/_s > 1/_hr

To enter meters per second squared:
27_m/_s^2

To convert meters per second squared from seconds to hours:

$$
27 \_m / \_s^{\wedge} 2 \mid 1 / \_h r^{\wedge} 2
$$

Because a Time conversion is not specified, it is shown in its default unit ( $\_s$ in this example).


Because a Length conversion is not specified, it is shown in its default unit (_m in this example).


## For Temperature Values

To convert a temperature value, you must use $\operatorname{tmpCnv}()$ instead of the operator.

$$
\begin{array}{r}
\operatorname{tmpCnv}\left(\text { expression_ }{ }_{-}^{\circ} \text { tempUnitl, },{ }^{\circ} \text { tempUnit } 2\right. \text { ) } \\
\text { For } \left.{ }^{\circ} \text {, press 2nd [ }{ }^{\circ}\right]
\end{array}
$$

For example, to convert $100{ }^{\circ} \mathrm{C}$ to _ ${ }^{\circ} \mathrm{F}$ : $\operatorname{tmpCnv}\left(100{ }^{\circ}{ }^{\circ},{ }_{-}{ }^{\circ} f\right)$


## For Temperature Ranges

To convert a temperature range (the difference between two temperature values), use $\Delta t m p C n v()$.
$\Delta \mathbf{t m p C n v}\left(\right.$ expression ${ }^{\circ}{ }^{\text {tempUnit1, }}{ }^{\circ}{ }^{\text {tempUnit } 2)}$

For example, to convert a $100{ }^{\circ} \mathrm{C}$ range to its equivalent range in _${ }^{\circ} \mathrm{F}$ :

$\Delta \operatorname{tmpCnv}\left(100{ }_{-}^{\circ} c,{ }_{-}{ }^{\prime} f\right)$
Note: For $\Delta$, press:
$\rightarrow \square$ [D]


## Setting the Default Units for Displayed Results

All results involving units are displayed in the default unit for that category. For example, if the default unit for Length is _m, any length result is displayed in meters (even if you entered _km or _ft in the calculation).

## If You're Using the SI or ENG/US System

The SI and ENG/US systems of measurement (set from Page 3 of the MODE screen) use built-in default units, which you cannot change.

The default units for these systems are available.

|  |
| :---: |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

## Setting Custom Defaults

To set custom defaults:

1. Press MODE F3 (1) 3 to set Unit System = CUSTOM.
2. Press $\Theta$ to highlight SET DEFAULTS.

3. Press (1) to display the CUSTOM UNIT DEFAULTS dialog box.
4. For each category, you can highlight its default, press $(1)$, and select a unit from the list.
5. Press ENTER twice to save your changes and exit the MODE screen.


## Notes:

- You can also use setUnits( ) or getUnits( ) to set or return information about default units. Refer to the Technical Reference module.
- When the CUSTOM UNIT DEFAULTS dialog box first appears, it shows the current default units.


## What is the NONE Default?

Many categories let you select NONE as the default unit.
This means that results in that category are displayed in the default units of its components.

For example, Area $=$ Length $^{2}$, so Length is the component of Area.


- If the defaults are Area = _acre and Length = _m (meters), area results are shown with _acre units.
- If you set Area $=$ NONE, area results are shown with _ $m^{2}$ units.

Note: NONE is not available for base categories such as Length and Mass that have no components.

## Creating Your Own User-Defined Units

In any category, you can expand the list of available units by defining a new unit in terms of one or more pre-defined units. You can also use "standalone" units.

## Why Use Your Own Units?

Some example reasons to create a unit are:

- You want to enter length values in dekameters. Define 10_m as a new unit named _dm.
- Instead of entering _ $m / s^{2}$ as an acceleration unit, you define that combination of units as a single unit named _ms2.
- You want to calculate how many times someone blinks. You can use _blinks as a valid unit without defining it. This "standalone" unit is treated similar to a variable that is not defined. For instance, 3_blinks is treated the same as 3 a.

Note: If you create a user-defined unit for an existing category, you can select it from the UNITS dialog box menu. But you cannot use MODE to select the unit as a default for displayed results.

## Rules for User-Defined Unit Names

The naming rules for units are similar to variables.

- Can have up to 8 characters.
- First character must be an underscore. For _, press:
- [-]
- Second character can be any valid variable name character except _ or a digit. For example, _ $9 f$ is not valid.
- Remaining characters (up to 6) can be any valid variable name character except an underscore.


## Defining a Unit

Define a unit the same way you store to a variable.

```
definition \(\rightarrow\) _newUnit
    \(\square\) For \(\rightarrow\), press STO
```

For example, to define a dekameter unit:

$$
\text { 10_m } \rightarrow \text { _dm }
$$

To define an acceleration unit:

$$
\text { _m/_s^2 } \rightarrow \text { _ms2 }
$$

To calculate 195 blinks in 5 minutes as _blinks/_min:

```
195_blinks/(5_min)
```



## Notes:

- User-defined units are displayed in lowercase characters, regardless of the case you use to define them.
- User-defined units such as _dm are stored as variables. You can delete them the same as you would any variable.


## List of Pre-Defined Constants and Units

This section lists the pre-defined constants and units by category. You can select any of these from the UNITS dialog box. If you use MODE to set default units, note that categories with only one defined unit are not listed.

## Defaults for SI and ENG/US

The SI and ENG/US systems of measurement use built-in default units. In this section, the built-in defaults are indicated by (SI) and (ENG/US). In some categories, both systems use the same default.

Some categories do not have default units.

## Constants

|  | Description | Value |
| :---: | :---: | :---: |
| _c | speed of light | 2.99792458E8_m/_s |
| _Cc | coulomb constant | 8.9875517873682 E9_N - $\mathrm{m}^{2}$ /_coul ${ }^{2}$ |
| _g | acceleration of gravity | 9.80665 _m/_s ${ }^{2}$ |
| _Gc | gravitational constant | $6.67259 E-11 \_\mathrm{m}^{3} /$ _kg/_s ${ }^{2}$ |
| _h | Planck's constant | 6.6260755E-34_J•_s |
| _k | Boltzmann's constant | 1.380658E-23_J/_ ${ }^{\circ} \mathrm{K}$ |
| _Me | electron rest mass | 9.1093897E-31_kg |


|  | Description | Value |
| :---: | :---: | :---: |
| _Mn | neutron rest mass | 1.6749286E-27_kg |
| _Mp | proton rest mass | 1.6726231E-27_kg |
| _Na | Avogadro's number | 6.0221367 E 23 /_mol |
| _q | electron charge | 1.60217733E-19_coul |
| _Rb | Bohr radius | 5.29177249E-11_m |
| _Rc | molar gas constant | 8.31451 _ $/$ / mol/_ ${ }^{\circ} \mathrm{K}$ |
| _Rdb | Rydberg constant | 10973731.53413/_m |
| _Vm | molar volume | 2.241409E-2_m ${ }^{3} /$ mol |
| _ $\varepsilon 0$ | permittivity of a vacuum | 8.8541878176204E-12_F/_m |
| - ${ }^{\circ}$ | Stefan-Boltzmann constant | $5.6705119 \mathrm{E}-8 . \mathrm{W} / \mathrm{m}^{2}$ _ $^{\circ} \mathrm{K}^{4}$ |
| _¢0 | magnetic flux quantum | $2.0678346161 \mathrm{E}-15$ WWb |
| _ $\mu 0$ | permeability of a vacuum | 1.2566370614359E-6_N/_A ${ }^{2}$ |
| ${ }_{\mu}{ }^{\text {b }}$ | Bohr magneton | 9.2740154E-24_J •_m²/Wb |

## Notes:

- The calculator simplifies unit expressions and displays results according to your default units. Therefore, constant values displayed on your screen may appear different from the values in this table.
- For Greek characters, refer to Quick Reference Key Table.

Length

| _Ang | angstrom | _mi | mile |
| :---: | :---: | :---: | :---: |
| _au | astronomical unit | _mil | 1/1000 inch |
| _cm | centimeter | _mm | millimeter |
| _fath | fathom | _Nmi | nautical mile |
| _fm | fermi | _pc | parsec |
| _ft | foot (ENG/US) | _rod | rod |
| _in | inch | _yd | yard |
| _km | kilometer | _ $\mu$ | micron |
| _Ityr | light year | _A | angstrom |
| _m | meter (SI) |  |  |

## Area

| _acre | acre | NONE (SI) (ENG/US) |  |
| :--- | :--- | :--- | :--- |
| _ha | hectare |  |  |

## Volume

| _cup | cup | _ml | milliliter |
| :--- | :--- | :---: | :--- |
| _floz | fluid ounce | _pt | pint |
| flozUK | British fluid ounce | _qt | quart |


| _gal | gallon | _tbsp | tablespoon |
| :--- | :--- | :--- | :--- |
| _galUK | British gallon | _tsp | teaspoon |
| - I | liter | NONE $($ SI $)$ (ENG/US) |  |

## Time

| _day | day | _s | second (SI) (ENG/US) |
| :---: | :--- | :--- | :--- |
| _hr | hour | _week | week |
| min | minute | _yr | year |
| ms | millisecond | $\_\mu \mathrm{s}$ | microsecond |
| ns | nanosecond |  |  |

## Velocity

| - knot | knot | _mph | miles per hour |
| :---: | :--- | :--- | :--- |
| - kph | kilometers per hour | NONE (SI) (ENG/US) |  |

## Acceleration

no pre-defined units $\square$

## Temperature

| $-{ }^{\circ} \mathrm{C}$ | ${ }^{\circ}$ Celsius (For ${ }^{\circ}$, press <br> 2nd $\left.\left[{ }^{\circ}\right].\right)$ | $-{ }^{\circ} \mathrm{K}$ | ${ }^{\circ}$ Kelvin |
| :---: | :--- | :--- | :--- |
| $-{ }^{\circ} \mathrm{F}$ | ${ }^{\circ}$ Fahrenheit | $-{ }^{\circ} \mathrm{R}$ | ${ }^{\circ}$ Rankine (no default) |

## Luminous Intensity

| _cd | candela (no default) |  |  |
| :--- | :--- | :--- | :--- |

## Amount of Substance

| -mol | mole (no default) |  |  |
| :--- | :--- | :--- | :--- |

## Mass

| _amu | atomic mass unit | _oz | ounce |
| :---: | :---: | :---: | :---: |
| _gm | gram | _slug | slug |
| _kg | kilogram (SI) | _ton | ton |
| _lb | pound (ENG/US) | _tonne | metric ton |
| _mg | milligram | _tonUK | long ton |
| _mton | metric ton |  |  |

## Force

| _dyne | dyne | _N | newton (SI) |
| :---: | :--- | :--- | :--- |
| _kgf | kilogram force | _tonf | ton force |
| _lbf | pound force (ENG/US) |  |  |

## Energy

| _Btu | British thermal unit (ENG/US) | _J | joule (SI) |
| :---: | :---: | :---: | :---: |
| _cal | calorie | _kcal | kilocalorie |
| _erg | erg | _kWh | kilowatt-hour |
| _eV | electron volt | _latm | liter-atmosphere |
| _ftb | foot-pound |  |  |

## Power

| - hp | horsepower (ENG/US) | _W | watt (SI) |
| :--- | :--- | :--- | :--- |
| - kW | kilowatt |  |  |

## Pressure

| - atm | atmosphere | _mmHg | millimeters of mercury |
| :--- | :--- | :--- | :--- |
| - bar | bar | $\_^{\mathrm{Pa}}$ | pascal (SI) |


| -inH 2 O | inches of water | _psi | pounds per square <br> inch (ENG/US) |
| :---: | :--- | :--- | :--- |
| -inHg | inches of mercury | - torr | millimeters of mercury |
| -mmH 2 O | millimeters of water |  |  |

## Viscosity, Kinematic

| St | stokes |  |  |
| :--- | :--- | :--- | :--- |

## Viscosity, Dynamic

| -P | poise |  |  |
| :--- | :--- | :--- | :--- |

## Frequency

| -GHz | gigahertz | -kHz | kilohertz |
| :--- | :--- | :--- | :--- |
| -Hz | hertz (SI) (ENG/US | -MHz | megahertz |

## Electric Current

| $\_$A | ampere (SI) (ENG/US) | -mA | milliampere |
| :--- | :--- | :--- | :--- |
| $-k A$ | kiloampere | $\_\mu \mathrm{A}$ | microampere |

## Charge

| _coul | coulomb (SI) (ENG/US) |  |  |
| :--- | :--- | :--- | :--- |

## Potential

| - kV | kilovolt | _V | volt (SI) (ENG/US) |
| :--- | :--- | :--- | :--- |
| -mV | millivolt | _volt | volt |

## Resistance

| $-\mathrm{k} \Omega$ | kilo ohm | _ohm | ohm |
| :--- | :--- | :--- | :--- |
| $-\mathrm{M} \Omega$ | megaohm | $\_^{\Omega}$ | ohm (SI) (ENG/US) |

## Conductance

| _mho | mho (ENG/US) | _siemens | siemens (SI) |
| :--- | :--- | :--- | :--- |
| - mmho | millimho | $\_\mu$ mho | micromho |

## Capacitance

| -F | farad (SI) (ENG/US) | $\__{\mathrm{pF}}$ | picofarad |
| :--- | :--- | :--- | :--- |
| $\mathrm{n}^{\mathrm{nF}}$ | nanofarad | $\__{\mu \mathrm{F}}$ | microfarad |

## Mag Field Strength

| -Oe | oersted | NONE (SI) (ENG/US) |
| :--- | :--- | :--- |

## Mag Flux Density

| _Gs | gauss | $-^{\top}$ | tesla (SI) (ENG/US) |
| :--- | :--- | :--- | :--- |

## Magnetic Flux

| -Wb | weber (SI) (ENG/US) |  |  |
| :--- | :--- | :--- | :--- |

## Inductance

| _henry | henry (SI) (ENG/US) | $\__{-} \mathrm{nH}$ | nanohenry |
| :--- | :--- | :---: | :--- |
| $\_\mathrm{mH}$ | millihenry | $\__{-} \mathrm{H}$ | microhenry |

## Basic Function Graphing

## Overview of Steps in Graphing Functions

To graph one or more $\mathbf{y}(\mathbf{x})$ functions, use the general steps shown below. For a detailed description of each step, refer to the following pages. You may not need to do all the steps each time you graph a function.

## Graphing Functions

1. Set Graph mode (MODE) to FUNCTION. Also set Angle mode, if necessary.

2. Define x and y components on $\mathrm{Y}=$ Editor ( $\bullet[Y=]$ ).
3. Select (ㅌ4) which defined functions to graph.
Note: To turn off any stat data plots, press [55 5 or use F4 to deselect them.
4. Set the display style for a function. 2nd [F6]

This is optional. For multiple equations,
 this helps visually distinguish one from another.
5. Define the viewing window $\square$ [WINDOW]).

F2 Zoom also changes the viewing window.
6. Change the graph format if necessary.

F1 9

- or -
$\rightarrow \square$

KMir=-1日。
$\times \Gamma \mathrm{B}=1 \mathrm{~B}$
$\times \mathrm{x} \cdot \mathrm{C}=1=$

$\cdots \times x=10$.
보튼ㅇ․
자르르․․․․

7. Graph the selected functions $(\square$ [GRAPH $]$ ).


## Exploring the Graph

From the Graph screen, you can:

- Display the coordinates of any pixel by using the free-moving cursor, or of a plotted point by tracing a function.
- Use the F2 Zoom toolbar menu to zoom in or out on a portion of the graph.
- Use the F5 Math toolbar menu to find a zero, minimum, maximum, etc.


## Setting the Graph Mode

Before graphing $\mathbf{y}(\mathbf{x})$ functions, you must select FUNCTION graphing. You may also need to set the Angle mode, which affects how the TI-89 Titanium graphs trigonometric functions.

## Graph Mode

1. Press MODE to display the MODE dialog box, which shows the current mode settings.

2. Set the Graph mode to FUNCTION. Refer to "Setting Modes" in Operating the Calculator.
For graphs that do not use complex numbers, set Complex Format = REAL. Otherwise, it may affect graphs that use powers, such as $\mathbf{x}^{1 / 3}$.

While this module specifically describes $\mathbf{y}(\mathbf{x})$ function graphs, the calculator lets you select from six Graph mode settings.

| Graph Mode Setting | Description |
| :--- | :--- |
| FUNCTION | $\mathbf{y}(\mathbf{x})$ functions |
| PARAMETRIC | $\mathbf{x}(\mathbf{t})$ and $\mathbf{y}(\mathbf{t})$ parametric equations |


| Graph Mode Setting | Description |
| :--- | :--- |
| POLAR | $\mathbf{r}(\theta)$ polar equations |
| SEQUENCE | $\mathbf{u}(\mathbf{n})$ sequences |
| 3D | $\mathbf{z}(\mathbf{x}, \mathbf{y})$ 3D equations |
| DIFFERENTIAL EQUATION | $\mathbf{y}^{\prime}(\mathbf{t})$ differential equations |

## Angle Mode

When using trigonometric functions, set the Angle mode for the units (RADIAN or DEGREE) in which you want to enter and display angle values.

## Checking the Status Line

To see the current Graph mode and Angle mode, check the status line at the bottom of the screen.

| r-til\| | FiȦ[1 HUTD | FUKIE |
| :---: | :---: | :---: |
|  | Angle <br> Mode | Graph <br> Mode |

## Defining Functions for Graphing

In FUNCTION graphing mode, you can graph functions named $\mathbf{y 1}(\mathbf{x})$ through $\mathbf{y 9 9}(\mathbf{x})$. To define and edit these functions, use the $Y=$ Editor. (The $Y=$ Editor lists function names for
the current graphing mode. For example, in POLAR graphing mode, function names are r1( $\theta)$, r2( $\theta)$, etc.)

## Defining a New Function

1. Press $\square[Y=]$ to display the $Y=$ Editor.


Plots - You can scroll above y1= to see a list of stat plots.
Function List - You can scroll through the list of functions and definitions.

Entry Line - Where you define or edit the function highlighted in the list.

Note: The function list shows abbreviated function names such as $\mathbf{y 1}$, but the entry line shows the full name $\mathbf{y 1}(\mathbf{x})$.
2. Press $\Theta$ and $\Theta$ to move the cursor to any undefined function. (Use 2nd $\Theta$ and 2nd $\odot$ to scroll one page at a time.)
3. Press ENTER or [F3 to move the cursor to the entry line.
4. Type the expression to define the function.

- The independent variable in function graphing is $x$.
- The expression can refer to other variables, including matrices, lists, and other functions. Only floats and lists of floats will produce a plot.

Note: For an undefined function, you do not need to press ENTER or F33. When you begin typing, the cursor moves to the entry line.
5. When you complete the expression, press ENTER.

The function list now shows the new function, which is automatically selected for graphing.

Note: If you accidentally move the cursor to the entry line, press ESC to move it back to the function list.

## Editing a Function

From the $\mathrm{Y}=$ Editor:

1. Press $\Theta$ and $\Theta$ to highlight the function.
2. Press ENTER or F3 to move the cursor to the entry line.
3. Do any of the following:

- Use (1) and (1) to move the cursor within the expression and edit it. Refer to "Editing an Expression in the Entry Line" in Operating the Calculator.
- or -
- Press CLEAR once or twice to clear the old expression, and then type the new one.

4. Press ENTER.

The function list now shows the edited function, which is automatically selected for graphing.

Note: To cancel any editing changes, press ESC instead of ENTER.

## Clearing a Function

From the $Y=$ Editor:

| To erase: | Do this: |
| :--- | :--- |
| A function from the function <br> list | Highlight the function and press $\square$ or <br> CLEAR. |
| A function from the entry line | Press CLEAR once or twice (depending on <br> the cursor's location) and then press |
|  | ENTER. | | Press F1 and then select 8:Clear |
| :--- | :--- |
| Functions. When prompted for |
| confirmation, press ENTER. |

Note: F1 8 does not erase any stat plots.
You don't have to clear a function to prevent it from being graphed. You can select the functions you want to graph.

## Shortcuts to Move the Cursor

From the $\mathrm{Y}=$ Editor:

| Press: | To: |
| :--- | :--- |
| $\Theta$ or | Go to function 1 or to the last defined function, <br> respectively. If the cursor is on or past the last <br> defined function, $\Theta \Theta$ goes to function 99. |

## From the Home Screen or a Program

You can also define and evaluate a function from the Home screen or a program.

- Use the Define and Graph commands. Refer to:
- "Graphing a Function Defined on the Home Screen" and "Graphing a Piecewise Defined Function" in Additional Graphing Topics.
- "Overview of Entering a Function" in Programming.
- Store an expression directly to a function variable. Refer to:
- "Storing and Recalling Variable Values" in Operating the Calculator.
- "Creating and Evaluating User-Defined Functions" in Calculator Home Screen.

Note: User-defined functions can have almost any name. However, if you want them to appear in the $Y=$ Editor, use function names $\mathbf{y 1}(\mathbf{x}), \mathbf{y 2}(\mathbf{x})$, etc.

## Selecting Functions to Graph

Regardless of how many functions are defined in the $\mathrm{Y}=$ Editor, you can select the ones you want to graph.

## Selecting or Deselecting Functions

Press $\bullet[Y=]$ to display the $Y=$ Editor.
A " $\checkmark$ " indicates which functions will be graphed the next time you display the Graph screen.


If PLOT numbers are displayed, those stat plots are selected.
In this example, Plots 1 and 2 are selected. To view them, scroll above y1=.

| To select or deselect: | Do this: |
| :--- | :--- |
| A specified function | - Move the cursor to highlight the function. |
|  | - $\quad$ Press F4. |
|  | This procedure selects a deselected <br> function or deselects a selected function. |
| All functions | - $\quad$ Press F5 to display the All toolbar menu. |
|  | - $\quad$ Select the applicable item. |
|  |  |
|  |  |
|  |  |

You don't have to select a function when you enter or edit it; it is selected automatically. To turn off any stat plots, press F5 5 or use F4 to deselect them.

## From the Home Screen or a Program

You can also select or deselect functions from the Home screen or a program.

- Use the FnOn and FnOff commands (available from the Home screen's F4 Other toolbar menu) for functions. Refer to the Technical Reference module.
- Use the PlotsOn and PlotsOff commands for stat plots. Refer to the Technical Reference module.


## Setting the Display Style for a Function

For each defined function, you can set a style that specifies how that function will be graphed. This is useful when graphing multiple functions. For example, set one as a solid line, another as a dotted line, etc.

## Displaying or Changing a Function's Style

From the $Y=$ Editor:

1. Move the cursor to highlight the applicable function.
2. Select the Style menu and press: 2nd [F6]


- Although the Line item is initially highlighted, the function's current style is indicated by a $\checkmark$ mark.
- To exit the menu without making a change, press ESC.

3. To make a change, select the applicable style.

| Style | Description |
| :--- | :--- |
| Line | Connects plotted points with a line. This is the default. |
| Dot | Displays a dot at each plotted point. |
| Square | Displays a solid box at each plotted point. |
| Thick | Connects plotted points with a thick line. |
| Animate | A round cursor moves along the leading edge of the graph <br> but does not leave a path. |
| Path | A round cursor moves along the leading edge of the graph <br> and does leave a path. |
| Above | Shades the area above the graph. |
| Below | Shades the area below the graph. |

To set Line as the style for all functions, press F5 and select 4:Reset Styles.

## If You Use Above or Below Shading

The TI-89 Titanium has four shading patterns, used on a rotating basis. If you set one function as shaded, it uses the first pattern. The next shaded function uses the second pattern, etc. The fifth shaded function reuses the first pattern.

When shaded areas intersect, their patterns overlap.


## From the Home Screen or a Program

You can also set a function's style from the Home screen or a program. Refer to the Style command in the Technical Reference module.

## Defining the Viewing Window

The viewing window represents the portion of the coordinate plane displayed on the Graph screen. By setting Window variables, you can define the viewing window's boundaries and other attributes. Function graphs, parametric graphs, etc., have their own independent set of Window variables.

## Displaying Window Variables in the Window Editor

Press $\quad$ [WINDOW] to display the Window Editor.

```
F[H7]
$mis,zom
<M.\Xi<=10.
<EC1=1
<EC-1%=10
Mm.Gx=10.
Ma|=0
<ysec=1=
```

Window Variables
(shown in Window Editor)


Corresponding Viewing Window (shown on Graph screen)
Variable Description
xmin, xmax, Boundaries of the viewing window.
ymin, ymax

| Variable | Description |
| :---: | :---: |
| xscl, yscl | Distance between tick marks on the x and y axes. |
| xres | Sets pixel resolution (1 through 10) for function graphs. The default is 2 . <br> - At 1, functions are evaluated and graphed at each pixel along the x axis. <br> - At 10, functions are evaluated and graphed at every 10th pixel along the x axis. |

To turn off tick marks, set $\mathbf{x s c l}=\mathbf{0}$ and/or $\mathbf{y s c l}=\mathbf{0}$. Small values of $\mathbf{x r e s}$ improve the graph's resolution but may reduce the graphing speed.

## Changing the Values

From the Window Editor:

1. Move the cursor to highlight the value you want to change.
2. Do any of the following:

- Type a value or an expression. The old value is erased when you begin typing.
- or -
- Press CLEAR to clear the old value; then type the new one.
- or -
- Press (1) or © $(1)$ to remove the highlighting; then edit the value.

Values are stored as you type them; you do not need to press ENTER. ENTER simply moves the cursor to the next Window variable. If you type an expression, it is evaluated when you move the cursor to a different Window variable or leave the Window Editor.

## From the Home Screen or a Program

You can also store values directly to the Window variables from the Home screen or a program. Refer to "Storing and Recalling Variable Values" in Operating the Calculator.

## Changing the Graph Format

You can set the graph format to show or hide reference elements such as the axes, a grid, and the cursor's coordinates. Function graphs, parametric graphs, etc., have their own independent set of graph formats.

## Displaying Graph Format Settings

From the $Y=$ Editor, Window Editor, or Graph screen, press F1 and select 9:Format.


- The GRAPH FORMATS dialog box shows the current settings.
- To exit without making a change, press

Labils.............. DFF $\ddagger$
Entar=SilyE ESCOCACEL ESC.

You also can display the GRAPH FORMATS dialog box from the $\mathrm{Y}=$ Editor, Window Editor, or Graph screen. Press:

- 1

| Format | Description |
| :--- | :--- |
| Coordinates | Shows cursor coordinates in rectangular (RECT) or polar <br> (POLAR) form, or hides (OFF) the coordinates. |


| Format | Description |
| :--- | :--- |
| Graph Order | Graphs functions one at a time (SEQ) or all at the same <br> time (SIMUL). |
| Grid | Shows (ON) or hides (OFF) grid points that correspond to <br> the tick marks on the axes. |
| Axes | Shows (ON) or hides (OFF) the x and y axes. |
| Leading Cursor | Shows (ON) or hides (OFF) a reference cursor that tracks <br> the functions as they are graphed. |
| Labels | Shows (ON) or hides (OFF) labels for the x and y axes. |

To turn off tick marks, define the viewing window so that $\mathbf{x s c l}$ and/or $\mathbf{y s c l}=\mathbf{0}$.

## Changing Settings

From the GRAPH FORMATS dialog box:

1. Move the cursor to highlight the format setting.
2. Press (©) to display a menu of valid settings for that format.
3. Select a setting. Either:

- Move the cursor to highlight the setting, and then press ENTER.
- Or -
- Press the number for that setting.

4. After changing all applicable format settings, press ENTER to save your changes and close the GRAPH FORMATS dialog box.
Note: To cancel a menu or exit the dialog box without saving any changes, use ESC instead of ENTER.

## Graphing the Selected Functions

When you are ready to graph the selected functions, display the Graph screen. This screen uses the display style and viewing window that you previously defined.

## Displaying the Graph Screen

Press $\square$ [GRAPH]. The TI-89 Titanium automatically graphs the selected functions.


BUSY indicator shows while graphing is in progress.

If you select an F22 Zoom operation from the $\mathrm{Y}=$ Editor or Window Editor, the TI-89 Titanium automatically displays the Graph screen.

## Interrupting Graphing

While graphing is in progress:

- To pause graphing temporarily, press ENTER. (The PAUSE indicator replaces BUSY.) To resume, press ENTER again.
- To cancel graphing, press $0 \mathbb{N}$. To start graphing again from the beginning, press F4 (ReGraph).


## If You Need to Change the Viewing Window

Depending on various settings, a function may be graphed such that it is too small, too large, or offset too far to one side of the screen. To correct this:

- Redefine the viewing window with different boundaries.
- Use a Zoom operation.


## Smart Graph

When you display the Graph screen, the Smart Graph feature displays the previous window contents immediately, provided nothing has changed that requires regraphing.

Smart Graph updates the window and regraphs only if you have:

- Changed a mode setting that affects graphing, a function's graphing attribute, a Window variable, or a graph format.
- Selected or deselected a function or stat plot. (If you only select a new function, Smart Graph adds that function to the Graph screen.)
- Changed the definition of a selected function or the value of a variable in a selected function.
- Cleared a drawn object.
- Changed a stat plot definition.


## Displaying Coordinates with the Free-Moving Cursor

To display the coordinates of any location on the Graph screen, use the free-moving cursor. You can move the cursor to any pixel on the screen; the cursor is not confined to a graphed function.

## Free-Moving Cursor

When you first display the Graph screen, no cursor is visible. To display the cursor, press a cursor pad arrow. The cursor moves from the center of the screen, and its coordinates are displayed.

$y 1(x)=x^{2}$
The " $c$ " indicates these are cursor coordinates. The values are stored in the $\mathbf{x c}$ and $\mathbf{y c}$ system variables. Rectangular coordinates use xc and yc. Polar coordinates use rc and $\theta$ c.

If your screen does not show coordinates, set the graph format so that Coordinates $=$ RECT or POLAR. Press:

- 1

To move the free-moving cursor: Press:
To an adjoining pixel
A cursor pad arrow for any direction.

To move the free-moving cursor: Press:
In increments of 10 pixels
2nd and then a cursor pad arrow.

Note: To hide the cursor and its coordinates temporarily, press CLEAR, ESC, or ENTER. The next time you move the cursor, it moves from its last position.

When you move the cursor to a pixel that appears to be "on" the function, it may be near the function but not on it.


To increase the accuracy:

- Use the Trace tool described on the next page to display coordinates that are on the function.
- Use a Zoom operation to zoom in on a portion of the graph.


## Tracing a Function

To display the exact coordinates of any plotted point on a graphed function, use the F3 Trace tool. Unlike the free-moving cursor, the trace cursor moves only along a function's plotted points.

## Beginning a Trace

From the Graph screen, press F3.
The trace cursor appears on the function, at the middle $x$ value on the screen. The cursor's coordinates are displayed at the bottom of the screen.

If multiple functions are graphed, the trace cursor appears on the lowest-numbered function selected in the $Y=$ Editor. The function number is shown in the upper right part of the screen.

If any stat plots are graphed, the trace cursor appears on the lowest-numbered stat plot.

Moving along a Function

| To move the trace cursor: | Do this: |
| :---: | :---: |
| To the previous or next plotted point | Press (1) or (1). |
| Approximately 5 plotted points (it may be more or less than 5 , depending on the xres Window variable) | Press 2nd (1) or 2nd (1). |

```
To move the trace cursor: Do this:
To a specified x value on the function Type the x value and
press ENTER.
```

Note: If you enter an $x$ value, it must be between $\mathbf{x m i n}$ and $\mathbf{x m a x}$.
The trace cursor moves only from plotted point to plotted point along the function, not from pixel to pixel.


If your screen does not show coordinates, set the graph format so that Coordinates = RECT or POLAR. Press:

- 1

Each displayed $y$ value is calculated from the $x$ value; that is, $\mathbf{y}=\mathbf{y n}(\mathbf{x})$. If the function is undefined at an $x$ value, the $y$ value is blank.

You can continue to trace a function that goes above or below the viewing window. You cannot see the cursor as it moves in that "off the screen" area, but the displayed coordinate values show its correct coordinates.

Note: Use QuickCenter to trace a function that goes above or below the window.

## Moving from Function to Function

Press $\Theta$ or $\odot$ to move to the previous or next selected function at the same x value. The new function number is shown on the screen.

The "previous or next" function is based on the order of the selected functions in the $Y=$ Editor, not the appearance of the functions as graphed on the screen.

## Automatic Panning

If you trace a function off the left or right edge of the screen, the viewing window automatically pans to the left or right. There is a slight pause while the new portion of the graph is drawn.


Before automatic pan


After automatic pan

After an automatic pan, the cursor continues tracing.
Note: Automatic panning does not work if stat plots are displayed or if a function uses a shaded display style.

## Using QuickCenter

If you trace a function off the top or bottom of the viewing window, you can press ENTER to center the viewing window on the cursor location.


Before using QuickCenter


After using QuickCenter

After QuickCenter, the cursor stops tracing. If you want to continue tracing, press F3.
You can use QuickCenter at any time during a trace, even when the cursor is still on the screen.

## Canceling Trace

To cancel a trace at any time, press ESC.
A trace is also canceled when you display another application screen such as the $Y=$ Editor. When you return to the Graph screen and press F3 to begin tracing:

- If Smart Graph regraphed the screen, the cursor appears at the middle $x$ value.
- If Smart Graph does not regraph the screen, the cursor appears at its previous location (before you displayed the other application).


## Using Zooms to Explore a Graph

The F2 Zoom toolbar menu has several tools that let you adjust the viewing window. You can also save a viewing window for later use.

## Overview of the Zoom Menu

Press F2 from the $\mathrm{Y}=$ Editor, Window Editor, or Graph screen.


> Procedures for using ZoomBox, ZoomIn, ZoomOut, ZoomStd, Memory, and SetFactors are given later in this section.

For more information about the other items, refer to the Technical Reference module.

Note: If you select a Zoom tool from the $\mathrm{Y}=$ Editor or Window Editor, the TI-89 Titanium automatically displays the Graph screen.

| Zoom Tool | Description |
| :--- | :--- |
| ZoomBox | Lets you draw a box and zoom in on that box. |
| ZoomIn, | Lets you select a point and zoom in or out by an amount <br> defined by SetFactors. |
| ZoomOut | Sets $\Delta x$ and $\Delta \mathrm{y}$ to .1, and centers the origin. |
| ZoomDec | Adjusts Window variables so that a square or circle is <br> shown in correct proportion (instead of a rectangle or <br> ellipse). |


| Zoom Tool | Description |
| :---: | :---: |
| ZoomStd | Sets Window variables to their default values. <br> $x \min =-10 \quad y \min =-10 \quad x$ res $=2$ <br> $x \max =10 \quad y \max =10$ <br> $\mathrm{xscl}=1 \quad \mathrm{yscl}=1$ |
| ZoomTrig | Sets Window variables to preset values that are often appropriate for graphing trig functions. Centers the origin and sets: $\begin{array}{lcc} \Delta \mathrm{x}=\pi / 24 & (.130899 \ldots \text { radians } & \mathrm{ymin}=-4 \\ & \text { or } 7.5 \text { degrees }) & \mathrm{ymax}=4 \\ \mathrm{xscl}=\pi / 2 & (1.570796 \ldots \text { radians } & \mathrm{yscl}=0.5 \\ & \text { or } 90 \text { degrees }) & \end{array}$ |
| ZoomInt | Lets you select a new center point, and then sets $\Delta x$ and $\Delta \mathrm{y}$ to 1 and sets $\mathbf{x s c l}$ and $\mathbf{y s c l}$ to 10. |
| ZoomData | Adjusts Window variables so that all selected stat plots are in view. |
| ZoomFit | Adjusts the viewing window to display the full range of dependent variable values for the selected functions. In function graphing, this maintains the current xmin and xmax and adjusts ymin and ymax. |
| Memory | Lets you store and recall Window variable settings so that you can recreate a custom viewing window. |
| SetFactors | Lets you set Zoom factors for Zoomln and ZoomOut. |

$\Delta x$ and $\Delta y$ are the distances from the center of one pixel to the center of an adjoining pixel.

## Zooming In with a Zoom Box

1. From the F2 Zoom menu, select 1:ZoomBox.

The screen prompts for 1 st Corner?
2. Move the cursor to any corner of the box you want to define, and then press ENTER.
The cursor changes to a small square, and the screen prompts for 2 nd Corner?

Note: To move the cursor in larger increments, use 2nd (1), 2nd $\odot$, etc.
3. Move the cursor to the opposite corner of the zoom box.

As you move the cursor, the box stretches.
4. When you have outlined the area you want to zoom in on, press ENTER.
The Graph screen shows the zoomed area. You can cancel ZoomBox by pressing ESC before you press ENTER.


## Zooming In and Out on a Point

1. From the F2 Zoom menu, select 2:ZoomIn or 3:ZoomOut.

A cursor appears, and the screen prompts for New Center?

2. Move the cursor to the point where you want to zoom in or out, and then press ENTER.

The TI-89 Titanium adjusts the Window
 variables by the Zoom factors defined in SetFactors.

- For a ZoomIn, the $x$ variables are divided by $\mathbf{x F a c t}$, and the y variables are divided by $y$ Fact.
new $\mathbf{x m i n}=\mathbf{x m i n} / \mathbf{x F a c t}$, etc.
- For a ZoomOut, the $x$ variables are multiplied by $\mathbf{x F a c t}$, and the y variables are multiplied by $\mathbf{y F a c t}$.
new $\mathbf{x m i n}=\mathbf{x m i n} * \mathbf{x F a c t}$, etc.


## Changing Zoom Factors

The Zoom factors define the magnification and reduction used by ZoomIn and ZoomOut.

1．From the F2 Zoom menu，select C：SetFactors to display the ZOOM FACTORS dialog box．


Zoom factors must be $\geq 1$ ，but they do not have to be integers．The default setting is 4 ．

Note：To exit without saving any changes，press ESC．
2．Use $\Theta$ and $\Theta$ to highlight the value you want to change．Then：
－Type the new value．The old value is cleared automatically when you begin typing．
－or－
－Press $(1)$ or $(\mathbb{1}$ to remove the highlighting，and then edit the old value．
3．Press ENTER（after typing in an input box，you must press ENTER twice）to save any changes and exit the dialog box．

## Saving or Recalling a Viewing Window

After using various Zoom tools，you may want to return to a previous viewing window or save the current one．

1．From the F2 Zoom menu，select B：Memory to display its submenu．

> 1: Zanmpren 2: 200 m3: Zロロm下icl

2．Select the applicable item．

| Select: | To: |
| :--- | :--- |
| 1:ZoomPrev | Return to the viewing window displayed before the <br> previous zoom. |
| 2:ZoomSto | Save the current viewing window. (The current Window <br> variable values are stored to the system variables <br> zxmin, zxmax, etc.) |
| $\mathbf{3 : Z o o m R c l}$ | Recall the viewing window last stored with ZoomSto. |

Note: You can store only one set of Window variable values at a time. Storing a new set overwrites the old set.

## Restoring the Standard Viewing Window

You can restore the Window variables to their default values at any time. From the F2 Zoom menu, select 6:ZoomStd.

## Using Math Tools to Analyze Functions

On the Graph screen, the F5 Math toolbar menu has several tools that help you analyze graphed functions.

## Overview of the Math Menu

Press F5 from the Graph screen.


| Math Tool | Description |
| :--- | :--- |
| Value | Evaluates a selected $\mathbf{y}(\mathbf{x})$ function at a specified x value. |
| Zero, <br> Minimum, <br> Maximum | Finds a zero ( x -intercept), minimum, or maximum point <br> within an interval. |
| Intersection | Finds the intersection of two functions. |
| Derivatives | Finds the derivative (slope) at a point. |
| ff(x)dx | Finds the approximate numerical integral over an interval. |
| Inflection | Finds the inflection point of a curve, where its second <br> derivative changes sign (where the curve changes <br> concavity). |
| Distance | Draws and measures a line between two points on the <br> same function or on two different functions. |
| Tangent | Draws a tangent line at a point and displays its equation. |
| Arc | Finds the arc length between two points along a curve. |

Math Tool Description

Shade Depends on the number of functions graphed.

- If only one function is graphed, this shades the function's area above or below the $x$ axis.
- If two or more functions are graphed, this shades the area between any two functions within an interval.

Note: For Math results, cursor coordinates are stored in system variables xc and yc (rc and $\theta$ c if you use polar coordinates). Derivatives, integrals, distances, etc., are stored in the system variable sysMath.

## Finding $y(x)$ at a Specified Point

1. From the Graph screen, press F5 and select 1:Value.
2. Type the $x$ value, which must be a real value between $x m i n$ and $x m a x$. The value can be an expression.
3. Press ENTER.

The cursor moves to that $x$ value on the first function selected in the $\mathrm{Y}=$ Editor, and its coordinates are displayed.

4. Press $\Theta$ or $\Theta$ to move the cursor between functions at the entered $x$ value. The corresponding y value is displayed.

If you press $(1)$ or $(1)$, the free-moving cursor appears. You may not be able to move it back to the entered x value.

You can also display function coordinates by tracing the function (F3), typing an $x$ value, and pressing ENTER.

## Finding a Zero, Minimum, or Maximum within an Interval

1. From the Graph screen, press F5 and select 2:Zero, 3:Minimum, or 4:Maximum.
2. As necessary, use $\odot$ and $\Theta$ to select the applicable function. Note: Typing $x$ values is a quick way to set bounds.
3. Set the lower bound for $x$. Either use (1) and (1) to move the cursor to the lower bound or type its $x$ value.
4. Press ENTER. $A>$ at the top of the screen marks the lower bound.
5. Set the upper bound, and press ENTER. The cursor moves to the solution, and its coordinates are displayed.


## Finding the Intersection of Two Functions within an Interval

1. From the Graph screen, press $F 5$ and select 5 :Intersection.
2. Select the first function, using $\Theta$ or $\Theta$ as necessary, and press ENTER. The cursor moves to the next graphed function.
3. Select the second function, and press ENTER.
4. Set the lower bound for $x$. Either use (1) and (1) to move the cursor to the lower bound or type its $x$ value.
5. Press ENTER. $A>$ at the top of the screen marks the lower bound.
6. Set the upper bound, and press ENTER.

The cursor moves to the intersection, and its coordinates are displayed.


## Finding the Derivative (Slope) at a Point

1. From the Graph screen, press F5 and select 6:Derivatives. Then select 1:dy/dx from the submenu.
2. As necessary, use $\Theta$ and $\Theta$ to select the applicable function.
3. Set the derivative point. Either move the cursor to the point or type its x value.
4. Press ENTER.

The derivative at that point is displayed.

## Finding the Numerical Integral over an Interval

1. From the Graph screen, press F5 and select $7: /[\mathbf{f}(\mathbf{x}) \mathbf{d x}$.
2. As necessary, use $\Theta$ and $\Theta$ to select the applicable function.

Note: Typing $x$ values is a quick way to set the limits.
3. Set the lower limit for $x$. Either use (1) and (1) to move the cursor to the lower limit or type its $x$ value.
4. Press ENTER. $\mathrm{A}>$ at the top of the screen marks the lower limit.

Note: To erase the shaded area, press F4 (ReGraph).
5. Set the upper limit, and press ENTER. The interval is shaded, and its approximate numerical integral is displayed.


## Finding an Inflection Point within an Interval

1. From the Graph screen, press $F 5$ and select $8:$ Inflection.
2. As necessary, use $\Theta$ and $\Theta$ to select the applicable function.
3. Set the lower bound for $x$. Either use (1) and (1) to move the cursor to the lower bound or type its $x$ value.
4. Press ENTER. A at the top of the screen marks the lower bound.
5. Set the upper bound, and press ENTER.

The cursor moves to the inflection point (if any) within the interval, and its coordinates are displayed.


## Finding the Distance between Two Points

1. From the Graph screen, press F5 and select 9:Distance.
2. As necessary, use $\Theta$ and $\Theta$ to select the function for the first point.
3. Set the first point. Either use (1) or (1) to move the cursor to the point or type its $x$ value.
4. Press ENTER. A + marks the point.
5. If the second point is on a different function, use $\odot$ and $\Theta$ to select the function.
6. Set the second point. (If you use the cursor to set the point, a line is drawn as you move the cursor.)
7. Press ENTER.

The distance between the two points is displayed, along with the connecting line.


## Drawing a Tangent Line

1. From the Graph screen, press F5 and select A:Tangent.
2. As necessary, use $\Theta$ and $\Theta$ to select the applicable function. Note: To erase a drawn tangent line, press F4 (ReGraph).
3. Set the tangent point. Either move the cursor to the point or type its x value.
4. Press ENTER.


The tangent line is drawn, and its equation is displayed.

## Finding an Arc Length

1. From the Graph screen, press F5 and select B:Arc.
2. As necessary, use $\Theta$ and $\Theta$ to select the applicable function.
3. Set the first point of the arc. Either use (1) or (1) to move the cursor or type the $x$ value.
4. Press ENTER. A + marks the first point.
5. Set the second point, and press ENTER. A + marks the second point, and the arc length is displayed.


## Shading the Area between a Function and the x Axis

You must have only one function graphed. If you graph two or more functions, the Shade tool shades the area between two functions.

1. From the Graph screen, press F5 and select C:Shade. The screen prompts for Above X axis?
2. Select one of the following. To shade the function's area:

- Above the $x$ axis, press ENTER.
- Below the x axis, press:
alpha $\mathbf{N}$

3. Set the lower bound for $x$. Either use (1) and (1) to move the cursor to the lower bound or type its $x$ value.

Note: If you do not press $(1)$ or $(\mathfrak{1}$, or type an $x$ value when setting the lower and upper bound, xmin and $\mathbf{x m a x}$ will be used as the lower and upper bound, respectively.
4. Press ENTER. A at the top of the screen marks the lower bound.
5. Set the upper bound, and press ENTER. The bounded area is shaded. To erase the shaded area, press F4 (ReGraph).


## Shading the Area between Two Functions within an Interval

You must have at least two functions graphed. If you graph only one function, the Shade tool shades the area between the function and the $x$ axis.

1. From the Graph screen, press F5 and select C:Shade. The screen prompts for Above?
2. As necessary, use $\Theta$ or $\Theta$ to select a function. (Shading will be above this function.)
3. Press ENTER. The cursor moves to the next graphed function, and the screen prompts for Below?
4. As necessary, use $\Theta$ or $\Theta$ to select another function. (Shading will be below this function.)
5. Press ENTER.
6. Set the lower bound for $x$. Either use (1) and (1) to move the cursor to the lower bound or type its $x$ value.
Note: If you do not press © or (1), or type an $x$ value when setting the lower and upper bound, xmin and xmax will be used as the lower and upper bound, respectively.
7. Press ENTER. $A>$ at the top of the screen marks the lower bound.
8. Set the upper bound, and press ENTER.

The bounded area is shaded. To erase the shaded area, press F4 (ReGraph).


## Polar Graphing

## Overview of Steps in Graphing Polar Equations

To graph polar equations, use the same general steps used for $\mathbf{y}(\mathbf{x})$ functions as described in Basic Function Graphing. Any differences that apply to polar equations are described on the following pages.

## Graphing Polar Equations

1. Set Graph mode (MODE) to POLAR. Also set Angle mode, if necessary.

2. Define $x$ and $y$ components on $Y=$ Editor ( $\bullet[\mathrm{Y}=]$ ).
3. Select ([F4) which defined equations to graph. Select the x or y component, or both.


Note: To turn off any stat data plots press
F5 5 or use F4 to deselect them.

4．Set the display style for an equation．You can set either the x or y component．

2nd［F6］

This is optional．For multiple equations， this helps visually distinguish one from another．

5．Define the viewing window（ $\square$［winDow］）．
F2 Zoom also changes the viewing window．

日mir＝
$\theta \mathrm{m} \times 12,56670144$
日曰ter＝ 130199698997
人Mir＝－10
$\times m i=10=$
$\times \mathrm{x} \times=10$
－ $\mathrm{m}_{\mathrm{i}} \mathrm{i}=-1 \mathrm{~B}$ 。


6．Change the graph format if necessary．
F1 9
－or－
$\bullet \square$


7．Graph the selected equations （ - ［GRAPH］）．


## Exploring the Graph

From the Graph screen，you can：
－Display the coordinates of any pixel by using the free－moving cursor，or of a plotted point by tracing a polar equation．

- Use the F2 Zoom toolbar menu to zoom in or out on a portion of the graph.
- Use the F5 Math toolbar menu to find derivatives, tangents, etc. Some menu items are not available for polar graphs.


## Differences in Polar and Function Graphing

This module assumes that you already know how to graph $\mathbf{y}(\mathbf{x})$ functions as described in Basic Function Graphing. This section describes the differences that apply to polar equations.

## Setting the Graph Mode

Use MODE to set Graph = POLAR before you define equations or set Window variables. The $Y=$ Editor and the Window Editor let you enter information for the current Graph mode setting only.

You should also set the Angle mode to the units (RADIAN or DEGREE) you want to use for $\theta$.

## Defining Polar Equations on the $\mathrm{Y}=$ Editor



You can use the Define command from the Home screen (see the Technical Reference module) to define functions and equations for any graphing mode, regardless of the current mode.

The $\mathrm{Y}=$ Editor maintains an independent function list for each Graph mode setting. For example, suppose:

- In FUNCTION graphing mode, you define a set of $\mathbf{y}(\mathbf{x})$ functions. You change to POLAR graphing mode and define a set of $\mathbf{r}(\theta)$ equations.
- When you return to FUNCTION graphing mode, your $\mathbf{y}(\mathbf{x})$ functions are still defined in the $\mathrm{Y}=$ Editor. When you return to POLAR graphing mode, your $\mathbf{r}(\theta)$ equations are still defined.


## Selecting the Display Style

The Above and Below styles are not available for polar equations and are dimmed on the $\mathrm{Y}=$ Editor's Style toolbar menu.

## Window Variables

The Window Editor maintains an independent set of Window variables for each Graph mode setting (just as the $Y=$ Editor maintains independent function lists). Polar graphs use the following Window variables.

| Variable | Description |
| :--- | :--- |
| $\theta \min , \theta \max$ | Smallest and largest $\theta$ values to evaluate. |


| Variable | Description |
| :---: | :---: |
| Ostep | Increment for the $\theta$ value. Polar equations are evaluated at: <br> $r(\theta$ min $)$ <br> $r(\theta$ min $+\theta$ step $)$ <br> r( $\theta$ min $+2(\theta$ step $)$ ) <br> ... not to exceed ... <br> r( $\theta$ max) |
| xmin, xmax, ymin, ymax | Boundaries of the viewing window. |
| xscl, yscl | Distance between tick marks on the x and y axes. |

Note: You can use a negative $\theta$ step. If so, $\theta$ min must be greater than $\theta$ max.
Standard values (set when you select 6:ZoomStd from the F2 Zoom toolbar menu) are:

| $\theta \min =0$. |  | $x \min =-10$. | $y \min =-10$. |
| :--- | :--- | :--- | :--- |
| $\theta \max =2 \pi$ | (6.2831853 $\ldots$ radians or <br> 360 degrees $)$ | $x \max =10$. | $y \max =10$. |
| $\theta$ step $=\pi / 24$ | (.1308996... radians or 7.5 <br> degrees) | $\mathrm{xscl}=1$. | $\mathrm{yscl}=1$. |
|  |  |  |  |

You may need to change the standard values for the $\theta$ variables $(\theta$ min, $\theta$ max, $\theta$ step $)$ to ensure that enough points are plotted.

## Setting the Graph Format

To display coordinates as rand $\theta$ values, use:

F1 9

- or -
$\square 1$
to set Coordinates $=$ POLAR. If Coordinates $=$ RECT, the polar equations will be graphed properly, but coordinates will be displayed as x and y .

When you trace a polar equation, the $\theta$ coordinate is shown even if Coordinates = RECT

## Exploring a Graph

As in function graphing, you can explore a graph by using the following tools. Any displayed coordinates are shown in polar or rectangular form as set in the graph format.
Tool For Polar Graphs:

Free-Moving Works just as it does for function graphs.
Cursor
F2 Zoom Works just as it does for function graphs.

- Only $\mathbf{x}$ (xmin, xmax, $\mathbf{x s c l}$ ) and $\mathbf{y}$ (ymin, ymax, yscl) Window variables are affected.
- The $\theta$ Window variables ( $\theta$ min, $\theta$ max, $\theta$ step) are not affected unless you select 6:ZoomStd (which sets $\theta \min =0, \theta \max =2 \pi$, and $\theta$ step $=\pi / 24$ ).
Tool For Polar Graphs:

F3 Trace Lets you move the cursor along a graph one $\theta$ step at a time.

- When you begin a trace, the cursor is on the first selected equation at $\theta$ min.
- QuickCenter applies to all directions. If you move the cursor off the screen (top or bottom, left or right), press ENTER to center the viewing window on the cursor location.
- Automatic panning is not available. If you move the cursor off the left or right side of the screen, the TI-89 Titanium / Voyage ${ }^{\text {TM }} 200$ graphing calculator will not automatically pan the viewing window. However, you can use QuickCenter.


During a trace, you can also evaluate $\mathbf{r}(\theta)$ by typing the $\theta$ value and pressing ENTER.
Note: You can use QuickCenter at any time during a trace, even if the cursor is still on the screen.

## Parametric Graphing

## Overview of Steps in Graphing Parametric Equations

To graph parametric equations, use the same general steps used for $\mathbf{y}(\mathbf{x})$ functions as described in Basic Function Graphing. Any differences that apply to parametric equations are described on the following pages.

## Graping Parametic Equations

1. Set Graph mode (MODE) to PARAMETRIC. Also set Angle mode, if necessary.

2. Define x and y components on $\mathrm{Y}=$ Editor ( $-[Y=]$ ).
3. Select ( (F4) , which defined equations to graph. Select the x or y component, or both.


Note: To turn off any stat data plots, press F5 5 or use F4 to deselect them.
4. Set the display style for an equation. You can set either the x or y component. 2nd [F6]

This is optional. For multiple equations, this helps visually distinguish one from another.
5. Define the viewing window ( $\square$ [wiNDOW]).
(F2) Zoom also changes the viewing window.
tmin= tstep= 02 xMin=-2 $\times m \mathrm{x}=25$. $x=\mathrm{Cl}=5$ ymir=$\cdots m=10$.

6. Change the graph format if necessary.
(F1 9

- or -
- 1


7. Graph the selected equations ( $\downarrow$ [GRAPH]).


## Exploring the Graph

From the Graph screen, you can:

- Display the coordinates of any pixel by using the free-moving cursor, or of a plotted point by tracing a parametric equation.
- Use the F2 Zoom toolbar menu to zoom in or out on a portion of the graph.
- Use the F5 Math toolbar menu to find derivatives, tangents, etc. Some menu items are not available for parametric graphs.


## Differences in Parametric and Function Graphing

This module assumes that you already know how to graph $\mathbf{y}(\mathbf{x})$ functions as described in Basic Function Graphing. This section describes the differences that apply to parametric equations.

## Setting the Graph Mode

Use MODE to set Graph = PARAMETRIC before you define equations or set Window variables. The $\mathrm{Y}=$ Editor and the Window Editor let you enter information for the current Graph mode setting only.

## Defining Parametric Equations on the $\mathrm{Y}=$ Editor

To graph a parametric equation, you must define both its $x$ and $y$ components. If you define only one component, the equation cannot be graphed. (However, you can use single components to generate an automatic table as described in Tables.)


Be careful when using implied multiplication with $\mathbf{t}$. For example:

| Enter: | Instead of: | Because: |
| :--- | :--- | :--- |
| $\mathrm{t} * \cos (60)$ | $\mathrm{t} \cos (60)$ | tcos is interpreted as a user-defined function <br> called tcos, not as implied multiplication. |
|  |  | In most cases, this refers to a nonexistent <br> function. So the TI-89 Titanium simply returns <br> the function name, not a number. |
|  |  |  |

Note: When using $\mathbf{t}$, be sure implied multiplication is valid for your situation. You can use the Define command from the Home screen (see the Technical Reference module) to define functions and equations for any graphing mode, regardless of the current mode.

The $\mathrm{Y}=$ Editor maintains an independent function list for each Graph mode setting. For example, suppose:

- In FUNCTION graphing mode, you define a set of $\mathbf{y}(\mathbf{x})$ functions. You change to PARAMETRIC graphing mode and define a set of $x$ and $y$ components.
- When you return to FUNCTION graphing mode, your $\mathbf{y}(\mathbf{x})$ functions are still defined in the $Y=$ Editor. When you return to PARAMETRIC graphing mode, your $x$ and $y$ components are still defined.


## Selecting Parametric Equations

To graph a parametric equation, select either its x or y component or both. When you enter or edit a component, it is selected automatically.

Selecting x and y components separately can be useful for tables as described in Tables. With multiple parametric equations, you can select and compare all the $x$ components or all the $y$ components.

## Selecting the Display Style

You can set the style for either the x or y component. For example, if you set the x component to Dot, the TI-89 Titanium automatically sets the y component to Dot.

Note: Use the Animate and Path styles for interesting projectile-motion effects.
The Above and Below styles are not available for parametric equations and are dimmed on the $\mathrm{Y}=$ Editor's Style toolbar menu.

## Window Variables

The Window Editor maintains an independent set of Window variables for each Graph mode setting (just as the $\mathrm{Y}=$ Editor maintains independent function lists). Parametric graphs use the following Window variables.

Note: You can use a negative tstep. If so, tmin must be greater than tmax.

| Variable | Description |
| :---: | :---: |
| tmin, tmax | Smallest and largest t values to evaluate. |
| tstep | Increment for the $\mathbf{t}$ value. Parametric equations are evaluated at: |
| xmin, xmax, ymin, ymax | Boundaries of the viewing window. |

Variable Description
xscl, yscl Distance between tick marks on the $x$ and $y$ axes.
Standard values (set when you select 6:ZoomStd from the F2 Zoom toolbar menu) are:

| $\operatorname{tmin}=0$ |  | $x \min =-10$. | $y m i n=-10$. |
| :--- | :--- | :--- | :--- |
| $\operatorname{tmax}=2 \pi$ | $(6.2831853 \ldots$ radians <br> or 360 degrees $)$ | $x \max =10$. | $y m a x=10$. |
| tstep $=\pi / 24$ | (.1308996 $\ldots$ radians <br> or 7.5 degrees $)$ | $x \operatorname{xsl}=1$. | $y s c l=1$. |

You may need to change the standard values for the $\mathbf{t}$ variables (tmin, tmax, tstep) to ensure that enough points are plotted.

## Exploring a Graph

As in function graphing, you can explore a graph by using the following tools.
Note: During a trace, you can also evaluate $\mathbf{x}(\mathrm{t})$ and $\mathbf{y}(\mathrm{t})$ by typing the t value and pressing ENTER. You can use QuickCenter at any time during a trace, even if the cursor is still on the screen.

## Tool

For Parametric Graphs:
Free-Moving Works just as it does for function graphs.
Cursor

| Tool | For Parametric Graphs: |
| :---: | :---: |
| F2 Zoom | Works just as it does for function graphs, with the following exceptions: <br> - Only $\mathbf{x}$ (xmin, $\mathbf{x m a x}, \mathbf{x s c l})$ and $\mathbf{y}$ (ymin, ymax, yscl) Window variables are affected. <br> - The $\mathbf{t}$ Window variables (tmin, tmax, tstep) are not affected unless you select 6:ZoomStd (which sets $\boldsymbol{t m i n}=0, \operatorname{tmax}=2 \pi$, and tstep $=\pi / 24$ ). |
| F3 Trace | Lets you move the cursor along a graph one tstep at a time. <br> - When you begin a trace, the cursor is on the first selected parametric equation at tmin. <br> - QuickCenter applies to all directions. If you move the cursor off the screen (top or bottom, left or right), press ENTER to center the viewing window on the cursor location. <br> - Automatic panning is not available. If you move the cursor off the left or right side of the screen, the TI-89 Titanium will not automatically pan the viewing window. However, you can use QuickCenter. |
| F5) Math | Only 1:Value, 6:Derivatives, 9:Distance, A:Tangent, and B:Arc are available for parametric graphs. These tools are based on $\mathbf{t}$ values. For example: <br> - 1:Value displays $x$ and $y$ values for a specified $t$ value. <br> - 6:Derivatives finds $\mathbf{d y} / \mathbf{d x}, \mathbf{d y} / \mathbf{d t}$, or $\mathbf{d x} / \mathbf{d t}$ at a point defined for a specified $\mathbf{t}$ value. |

## Sequence Graphing

## Overview of Steps in Graphing Sequences

To graph sequences, use the same general steps used for $\mathbf{y}(\mathbf{x})$ functions as described in Basic Function Graphing. Any differences are described on the following pages.

## Graphing Sequences

1. Set Graph mode (MODE) to SEQUENCE. Also set Angle mode, if necessary.

2. Define sequences and, if needed, initial values on $Y=$ Editor ( $\bullet[Y=]$ ).
3. Select ([F4) which defined sequences to graph. Do not select initial values.

Note: To turn off any stat data plots, press F5 5 or use F4 to deselect them.
4. Set the display style for a sequence.

2nd [F6]
For sequences, the default style is Square.


5．Define the viewing window（ $\square$［WINDOW］）．
F2 Zoom also changes the viewing window．

6．Change the graph format if necessary．
F1 9
－or－
－ 1


7．Graph the selected equations （ $\dagger$［GRAPH］）．


## Exploring the Graph

From the Graph screen，you can：
－Display the coordinates of any pixel by using the free－moving cursor，or of a plotted point by tracing a sequence．
－Use the F2 Zoom toolbar menu to zoom in or out on a portion of the graph．
－Use the F5 Math toolbar menu to evaluate a sequence．Only 1：Value is available for sequences．
－Plot sequences on Time（the default），Web，or Custom axes．
Note：You can also evaluate a sequence while tracing．Simply enter the n value directly from the keyboard．

## Differences in Sequence and Function Graphing

This module assumes that you already know how to graph $\mathbf{y}(\mathbf{x})$ functions as described in Basic Function Graphing. This section describes the differences that apply to sequences.

## Setting the Graph Mode

Use MODE to set Graph = SEQUENCE before you define sequences or set Window variables. The $Y=$ Editor and the Window Editor let you enter information for the current Graph mode setting only.

## Defining Sequences on the $\mathbf{Y}=$ Editor



If a sequence requires more than one initial value, enter them as a list enclosed in braces \{ \} and separated by commas. You must use a list to enter two or more initial values.

| $\begin{aligned} & 43=43(n-1)+43(n-2) \\ & 43=61 \\ & 4 i=1 \end{aligned}$ | ter $\{1,0\}$ even though $\{10\}$ is |
| :---: | :---: |
|  | , |

If a sequence requires an initial value but you do not enter one, you will get an error when graphing.

On the $\mathrm{Y}=$ Editor, Axes lets you select the axes that are used to graph the sequences. Optionally, for sequences only, you can select different axes for the graph. TIME is the default.

| Axes | Description |
| :--- | :--- |
| TIME | Plots $n$ on the $x$ axis and $\mathbf{u}(\mathrm{n})$ on the y axis. |
| WEB | Plots $\mathbf{u}(\mathrm{n}-1)$ on the x axis and $\mathbf{u ( n )}$ on the y axis. |
| CUSTOM | Lets you select the x and y axes. |

The $\mathrm{Y}=$ Editor maintains an independent function list for each Graph mode setting. For example, suppose:

- In FUNCTION graphing mode, you define a set of $\mathbf{y}(\mathbf{x})$ functions. You change to SEQUENCE graphing mode and define a set of $\mathbf{u}(\mathbf{n})$ sequences.
- When you return to FUNCTION graphing mode, your $\mathbf{y}(\mathbf{x})$ functions are still defined in the $\mathrm{Y}=$ Editor. When you return to SEQUENCE graphing mode, your $\mathbf{u}(\mathbf{n})$ sequences are still defined.

Note: You can use the Define command from the Home screen (see Technical Reference) to define functions and equations for any graphing mode, regardless of the current mode.

## Selecting Sequences

With TIME and WEB axes, the TI-89 Titanium graphs only the selected sequences. If you entered any sequences that require an initial value, you must enter the corresponding ui value.

Note: With TIME and CUSTOM axes, all defined sequences are evaluated even if they are not plotted.

You can select a sequence

You cannot select its initial value.


With CUSTOM axes, when you specify a sequence in the custom settings, it is graphed regardless of whether it is selected.

## Selecting the Display Style

Only the Line, Dot, Square, and Thick styles are available for sequence graphs. Dot and Square mark only those discrete integer values (in plotstep increments) at which a sequence is plotted.

## Window Variables

The Window Editor maintains an independent set of Window variables for each Graph mode setting (just as the $Y=$ Editor maintains independent function lists). Sequence graphs use the following Window variables.

| Variable | Description |
| :--- | :--- |
| nmin, nmax | Smallest and largest n values to evaluate. Sequences are <br> evaluated at: <br> $\mathrm{u}(\mathrm{nmin})$ <br> $\mathrm{u}(\mathrm{nmin}+1)$ <br> $\mathrm{u}(\mathrm{nmin}+2)$ <br> $\ldots$. not to exceed ... <br> $\mathrm{u}(\mathrm{nmax})$ |
| plotStrt | The term number that will be the first one plotted <br> (depending on plotstep). For example, to begin plotting <br> with the 2nd term in the sequence, set plotstrt = 2. The <br> first term will be evaluated at nmin but not plotted. |
| plotStep | Incremental n value for graphing only. This does not affect <br> how the sequence is evaluated, only which points are <br> plotted. For example, suppose plotstep $=2$. The <br> sequence is evaluated at each consecutive integer but is <br> plotted at only every other integer. |
| $\mathbf{x m i n}, \mathbf{x m a x}$, | Boundaries of the viewing window. <br> $\mathbf{y m i n}, \mathbf{y m a x}$ |
| $\mathbf{x s c l , ~ y s c l}$ | Distance between tick marks on the x and y axes. |

Note: Both nmin and nmax must be positive integers, although nmin can be zero; nmin, nmax, plotstrt and plotstep must be integers $\geq 1$. If you do not enter integers, they will be rounded to integers.

Standard values (set when you select 6:ZoomStd from the F2 Zoom toolbar menu) are:

| nmin $=1$. | $x \min =-10$. | $y \min =-10$. |
| :--- | :--- | :--- |
| nmax $=10$. | $x \max =10$. | $y \max =10$. |
| plotstrt $=1$. | $x s c l=1$. | $y s c l=1$. |
| plotstep $=1$. |  |  |

You may need to change the standard values for the n and plot variables to ensure that sufficient points are plotted.

To see how plotstrt affects graph, look at the following examples of a recursive sequence.

This graph is plotted beginning with the 1 st term.


This graph is plotted beginning with the 9th term.


Note: Both of these graphs use the same Window variables, except for plotstrt.

With TIME axes (from Axes on the $Y=$ Editor), you can set plotstrt = 1 and still graph only a selected part of the sequence. Simply define a viewing window that shows only the area of the coordinate plane you want to view.

You could set:

- $\quad \mathbf{x m i n}=$ first n value to graph
- $\quad \mathbf{x m a x}=\mathbf{n m a x}$ (although you can use other values)

- $\quad$ ymin and $y m a x=$ expected values for the sequence


## Changing the Graph Format

The Graph Order format is not available.

- With TIME or CUSTOM axes, multiple sequences are always plotted simultaneously.
- With WEB axes, multiple sequences are always plotted sequentially.


## Exploring a Graph

As in function graphing, you can explore a graph by using the following tools. Any displayed coordinates are shown in rectangular or polar form as set in the graph format.

| Tool | For Sequence Graphs: |
| :--- | :--- |
| Free-Moving | Works just as it does for function graphs. |
| Cursor |  |

Cursor

| Tool | For Sequence Graphs: |
| :---: | :---: |
| (F2) Zoom | Works just as it does for function graphs. <br> - Only $\mathbf{x}$ ( $\mathbf{x m i n}, \mathbf{x m a x}, \mathbf{x s c l})$ and $\mathbf{y}$ ( $\mathbf{y m i n}, \mathbf{y m a x}, \mathbf{y s c l}$ ) Window variables are affected. <br> - The n and plot Window variables (nmin, nmax, plotStrt, plotStep) are not affected unless you select 6:ZoomStd (which sets all Window variables to their standard values). |
| F3] Trace | Depending on whether you use TIME, CUSTOM, or WEB axes, Trace operates very differently. <br> - With TIME or CUSTOM axes, you move the cursor along the sequence one plotstep at a time. To move approximately ten plotted points at a time, press 2nd (1) or 2nd (1). <br> - When you begin a trace, the cursor is on the first selected sequence at the term number specified by plotstrt, even if it is outside the viewing window. <br> - QuickCenter applies to all directions. If you move the cursor off the screen (top or bottom, left or right), press ENTER to center the viewing window on the cursor location. <br> - With WEB axes, the trace cursor follows the web, not the sequence. |
| F5 Math | Only 1:Value is available for sequence graphs. <br> - With TIME and WEB axes, the $\mathbf{u}(\mathbf{n})$ value (represented by $\mathbf{y c}$ ) is displayed for a specified $n$ value. <br> - With CUSTOM axes, the values that correspond to x and $y$ depend on the axes you choose. |

During a trace, you can evaluate a sequence by typing a value for $n$ and pressing ENTER. You can use QuickCenter at any time during a trace, even if the cursor is still on the screen.

## Setting Axes for Time, Web, or Custom Plots

For sequences only, you can select different types of axes for the graph. Examples of the different types are given later in this module.

## Displaying the AXES Dialog Box

From the $\mathrm{Y}=$ Editor, Axes:

- Depending on the current Axes setting, some items may be dimmed.
- To exit without making any changes, press ESC.


| Item | Description |
| :--- | :--- |
| Axes | TIME - Plots $\mathbf{u}(\mathbf{n})$ on the $y$ axis and $n$ on the $x$ axis. <br>  <br>  <br>  <br> WEB - Plots $\mathbf{u ( n )}$ on the $y$ axis and $\mathbf{u ( n - 1 )}$ on the $x$ axis. <br> CUSTOM - Lets you select the $x$ and $y$ axes. |
| Build Web | Active only when Axes = WEB, this specifies whether a web <br> is drawn manually (TRACE) or automatically (AUTO). |


| Item | Description |
| :--- | :--- |
| X Axis | Active only when Axes = CUSTOM, these let you select the |
| and | value or sequence to plot on the $x$ and $y$ axes. |

Y Axis

To change any of these settings, use the same procedure that you use to change other types of dialog boxes, such as the MODE dialog box.

## Using Web Plots

A web plot graphs $\mathbf{u}(\mathbf{n})$ vs. $\mathbf{u ( n - 1})$, which lets you study the long-term behavior of a recursive sequence. The examples in this section also show how the initial value can affect a sequence's behavior.

## Valid Functions for Web Plots

A sequence must meet the following criteria; otherwise, it will not be graphed properly on WEB axes. The sequence:

- Must be recursive with only one recursion level; u(n-1) but not $\mathbf{u}(\mathbf{n - 2})$.
- Cannot reference $n$ directly.
- Cannot reference any other defined sequence except itself.


## When You Display the Graph Screen

After you select WEB axes and display the Graph screen, the TI-89 Titanium:

- Draws a $\mathbf{y}=\mathbf{x}$ reference line.
- Plots the selected sequence definitions as functions, with $\mathbf{u}(\mathbf{n}-\mathbf{1})$ as the independent variable. This effectively converts a recursive sequence into a nonrecursive form for graphing.

For example, consider the sequence $u 1(\mathrm{n})=\sqrt{5-u 1(n-1)}$ and an initial value of ui1=1.
The TI-89 Titanium draws the $\mathbf{y}=\mathbf{x}$ reference line and then plots $\mathrm{y}=\mathrm{y}=\sqrt{5-x}$.

## Drawing the Web

After the sequence is plotted, the web may be displayed manually or automatically, depending on how you set Build Web on the AXES dialog box.

| If Build Web $=$ | The web is: |
| :--- | :--- |
| TRACE | Not drawn until you press F3. The web is then drawn <br> step-by-step as you move the trace cursor (you must <br> have an initial value before using Trace). |
|  | Note: With WEB axes, you cannot trace along the <br> sequence itself as you do in other graphing modes. |
| AUTO | Drawn automatically. You can then press F3] to trace the <br> web and display its coordinates. |

The web:

1. Starts on the $x$ axis at the initial value ui (when plotstrt $=1$ ).
2. Moves vertically (either up or down) to the sequence.
3. Moves horizontally to the $\mathbf{y}=\mathbf{x}$ reference line.
4. Repeats this vertical and horizontal movement until $\mathbf{n}=\mathbf{n m a x}$.

Note: The web starts at plotstrt. The value of $n$ is incremented by 1 each time the web moves to the sequence (plotStep is ignored).

## Example: Convergence

1. On the $Y=$ Editor $(\square[Y=])$, define $u 1(n)=-.8 u 1(n-1)+3.6$. Set initial value ui1 $=-4$.
2. Set Axes $=$ TIME.
3. On the Window Editor ( $\square$ [wiNDOW]), set the Window variables.

| nmin=1 | xmin=0 | $y \min =-10$ |
| :--- | :--- | :--- |
| nmax=25 | $x \max =25$ | $y \max =10$ |
| plotstrt=1 | xscl=1 | $y s c l=1$ |
| plotstep=1 |  |  |

4. Graph the sequence ( $\square$ [GRAPH]). By default, a sequence uses the Square display style.

5. On the $\mathrm{Y}=$ Editor, set Axes $=\mathbf{W E B}$ and Build Web $=$ AUTO.
6. On the Window Editor, change the Window variables.

| $n \min =1$ | $x \min =-10$ | $y m i n=-10$ |
| :--- | :--- | :--- |
| $n m a x=25$ | $x m a x=10$ | $y m a x=10$ |
| plotstrt=1 | $x s c l=1$ | $y s c l=1$ |
| plotstep=1 |  |  |

7. Regraph the sequence.

Web plots are always shown as lines, regardless of the selected display style.
Note: During a trace, you can
 move the cursor to a specified $n$ value by typing the value and pressing ENTER.
8. Press $\mathbb{F 3}$. As you press ( 1 , the trace cursor follows the web. The screen displays the cursor coordinates nc, xc, and yc (where xc and yc represent $\mathbf{u}(\mathbf{n - 1})$ and $\mathbf{u ( n )}$, respectively).

As you trace to larger values of nc, you can see xc and yc approach the convergence point.

Note: When the nc value changes, the cursor is on the sequence. The next time you press (1), nc stays the same but the cursor is now on the $\mathbf{y}=\mathrm{x}$ reference line.

## Example: Divergence

1. On the $Y=\operatorname{Editor}(\square[Y=])$, define $u 1(n)=3.2 u 1(n-1)-.8(u 1(n-1))^{2}$. Set initial value ui1 $=4.45$.
2. Set Axes $=$ TIME.
3. On the Window Editor ( $\square$ [WINDOW]), set the Window variables.

| nmin=0 | xmin=0 | $y \min =-75$ |
| :--- | :--- | :--- |
| nmax $=10$ | $x \max =10$ | $y \max =10$ |
| plotstrt=1 | $x s c l=1$ | $y s c l=1$ |
| plotstep=1 |  |  |

4. Graph the sequence ( $\bullet$ [GRAPH]).

Because the sequence quickly diverges to large negative values, only a few points are plotted.

5. On the $\mathrm{Y}=$ Editor, set $\mathbf{A x e s}=\mathbf{W E B}$ and Build $\mathbf{W e b}=\mathbf{A U T O}$.
6. On the Window Editor ( $\square$ [window]), set the Window variables.

| $n \min =0$ | $x \min =-10$ | $y m i n=-10$ |
| :--- | :--- | :--- |
| $n m a x=10$ | $x m a x=10$ | $y m a x=10$ |
| plotstrt=1 | $x s c l=1$ | $y s c l=1$ |
| plotstep $=1$ |  |  |

7. Regraph the sequence.

The web plot shows how quickly the sequence diverges to large negative values.


## Example: Oscillation

This example shows how the initial value can affect a sequence.

1. On the $Y=$ Editor $(\square[Y=])$, use the same sequence defined in the divergence example: $u 1(n)=3.2 u 1(n-1)-.8(u 1(n-1))^{2}$. Set initial value ui1 $=0.5$.
2. Set Axes $=$ TIME.
3. On the Window Editor ( $\square[$ wiNDOW $]$ ), set the Window variables.

| nmin=1 | $x \min =0$ | $y \min =0$ |
| :--- | :--- | :--- |
| nmax $=100$ | $x \max =100$ | $y \max =5$ |
| plotstrt=1 | $x$ scl=10 | $y s c l=1$ |
| plotstep=1 |  |  |

4. Graph the sequence ( $\square$ [GRAPH $]$ ).

Note: Compare this graph with the divergence example. This is the same sequence with a different initial value.

5. On the $Y=$ Editor, set $A x e s=W E B$ and Build $\mathbf{W e b}=$ AUTO.
6. On the Window Editor ( $\square$ [wiNDOW]), set the Window variables.

| nmin=1 | xmin=2.68 | $y \min =4.7$ |
| :--- | :--- | :--- |
| nmax=100 | $x \max =6.47$ | $y \max =47$ |
| plotstrt=1 | $x s c l=1$ | $y s c l=1$ |

plotstep=1
7. Regraph the sequence.

Note: The web moves to an orbit oscillating between two stable points.


$$
y=3.2 x-.8 x^{2}
$$

8. Press [F3. Then use (1) to trace the web.

As you trace to larger values of nc, notice that xc and yc oscillate between 2.05218 and 3.19782.
9. On the Window Editor, set plotstrt=50. Then regraph the sequence.

Note: By starting the web plot at a later term, the stable oscillation orbit is
 shown more clearly.

## Using Custom Plots

CUSTOM axes give you great flexibility in graphing sequences. As shown in the following example, CUSTOM axes are particularly effective for showing relationships between one sequence and another.

## Example: Predator-Prey Model

Using the predator-prey model in biology, determine the numbers of rabbits and foxes that maintain population equilibrium in a certain region.
$\mathbf{R}=$ Number of rabbits
$\mathbf{M} \quad=\quad$ Growth rate of rabbits if there are no foxes (use .05)
$\mathbf{K}=$ Rate at which foxes can kill rabbits (use .001)
$\mathbf{W}=$ Number of foxes
G $\quad=\quad$ Growth rate of foxes if there are rabbits (use .0002)
D = Death rate of foxes if there are no rabbits (use .03)
$\mathbf{R n}=R_{\mathrm{n}-1}\left(1+\mathrm{M}-\mathrm{KW}_{\mathrm{n}-1}\right)$
$W n=W_{n-1}\left(1+G R_{n-1}-D\right)$

1. On the $Y=$ Editor $(\square[Y=])$, define the sequences and initial values for $R_{n}$ and $W_{n}$.

$$
\begin{aligned}
& \mathrm{u} 1(\mathrm{n})=\mathrm{u} 1(\mathrm{n}-1) *(1+.05-.001 * \mathrm{u} 2(\mathrm{n}-1)) \\
& \mathrm{ui} 1=200 \\
& \mathrm{u} 2(\mathrm{n})=\mathrm{u} 2(\mathrm{n}-1) *(1+.0002 * \mathrm{u} 1(\mathrm{n}-1)-.03) \\
& \mathrm{u} 2=50
\end{aligned}
$$

Note: Assume there are initially 200 rabbits and 50 foxes.
2. Set Axes $=$ TIME.
3. On the Window Editor ( $\square$ [wiNDOW]), set the Window variables.

| nmin=0 | xmin=0 | $y \min =0$ |
| :--- | :--- | :--- |
| nmax=400 | $x \max =400$ | $y m a x=300$ |
| plotstrt=1 |  |  |
| plotstep=1 | $x s c l=100$ | $y s c l=100$ |

4. Graph the sequence
( $\square$ [GRAPH]).
Note: Use F3 to individually trace the number of rabbits u1(n) and foxes u2(n) over time

(n).
5. On the $Y=$ Editor, set $\mathbf{A x e s}=\mathbf{C U S T O M}, \mathbf{X}$ Axis $=\mathbf{u 1}$, and $\mathbf{Y}$ Axis $=\mathbf{u 2}$.
6. On the Window Editor ( $\square$ [wiNDOW]), set the Window variables.

| nmin=0 | xmin=84 | $y \min =25$ |
| :--- | :--- | :--- |
| nmax=400 | $x \max =237$ | $y \max =75$ |
| plotstrt=1 | $x s c l=50$ | $y s c l=10$ |
| plotstep=1 |  |  |

7. Regraph the sequence.

Note: Use F3 to trace both the number of rabbits (xc) and foxes (yc) over the cycle of 400 generations.


## Using a Sequence to Generate a Table

Previous sections described how to graph a sequence. You can also use a sequence to generate a table. Refer to Tables for detailed information.

## Example: Fibonacci Sequence

In a Fibonacci sequence, the first two terms are 1 and 1. Each succeeding term is the sum of the two immediately preceding terms.

1. On the $Y=$ Editor $(\square[Y=])$, define the sequence and set the initial values as shown.


2. Set Window variables ( $\square$ [WINDOW]) so $n \times 1 n=1$,
$n m a x=10$
that nmin has the same value as tbIStart. ㅆ․․․ $=10$. Flotstrot ×Min= -
$\times M B \times=10$ -
$\times S E 1=1$

$1 \mathrm{~m}=\mathrm{x}=10$.

- 

4. Display the table ( $\square$ [TABLE]).


Fibonacci sequence is in column 2.
5. Scroll down the table $(\Theta$ or 2nd $\Theta)$ to see more of the sequence.

## 3D Graphing

## Overview of Steps in Graphing 3D Equations

To graph 3D equations, use the same general steps used for $\mathbf{y}(\mathbf{x})$ functions as described in Basic Function Graphing. Any differences that apply to 3D equations are described on the following pages.

## Graphing 3D Equations

1. Set Graph mode (MODE) to 3D. Also set Angle mode, if necessary.
2. Define 3D equations on $Y=$ Editor ( $\bullet[Y=]$ ).
3. Select (F4) which equation to graph. You can select only one 3D equation.
To turn off any stat data plots, press F5 5 or use F4 to deselect them.
4. Define the viewing cube ( $\square$ [window]). For 3D graphs, the viewing window is called the viewing cube. F2 Zoom also changes the viewing cube.


ечеө=20. Eyy $=70$. Ey
品in=-i
$\times \mathrm{M} \exists \mathrm{x}=1{ }^{\circ}$
$\times \mathrm{xmax}=1 \mathrm{id}=$

信 $\times \times 10$.



$\underset{n}{2 m o x t a i n}=5$
5. Change the graph format if necessary.

F1 9

- or -
- 1

Note: To help you see the orientation of 3D graphs, turn on Axes and Labels.
6. Graph the selected equations ( $\triangle$ [GRAPH]).

Note: Before displaying the graph, the
 screen shows the "percent evaluated."

## Exploring the Graph

From the Graph screen, you can:

- Trace the equation.
- Use the F2 Zoom toolbar menu to zoom in or out on a portion of the graph. Some of the menu items are dimmed because they are not available for 3D graphs.
- Use the F5 Math toolbar menu to evaluate the equation at a specified point. Only 1:Value is available for 3D graphs.

You can also evaluate $\mathbf{z}(\mathbf{x}, \mathbf{y})$ while tracing. Type the x value and press ENTER; then type the $y$ value and press ENTER.

## Differences in 3D and Function Graphing

This module assumes that you already know how to graph $\mathbf{y}(\mathbf{x})$ functions as described in Basic Function Graphing. This section describes the differences that apply to 3D equations.

## Setting the Graph Mode

Use MODE to set Graph = 3D before you define equations or set Window variables. The $\mathrm{Y}=$ Editor and the Window Editor let you enter information for the current Graph mode setting only.

## Defining 3D Equations on the $\mathrm{Y}=$ Editor



The $\mathrm{Y}=$ Editor maintains an independent function list for each Graph mode setting. For example, suppose:

- In FUNCTION graphing mode, you define a set of $\mathbf{y}(\mathbf{x})$ functions. You change to 3D graphing mode and define a set of $\mathbf{z}(\mathbf{x}, \mathbf{y})$ equations.
- When you return to FUNCTION graphing mode, your $\mathbf{y}(\mathbf{x})$ functions are still defined in the $Y=$ Editor. When you return to 3D graphing mode, your $\mathbf{z}(\mathbf{x}, \mathbf{y})$ equations are still defined.

Note: You can use the Define command from the Home screen (see the Technical Reference module) to define functions and equations for any graphing mode, regardless of the current mode.

## Selecting the Display Style

Because you can graph only one 3D equation at a time, display styles are not available. On the $Y=$ Editor, the Style toolbar menu is dimmed.

For 3D equations, however, you can use:
F1 9

- or -
- 1
to set the Style format to WIRE FRAME or HIDDEN SURFACE.


## Window Variables

The Window Editor maintains an independent set of Window variables for each Graph mode setting (just as the $Y=$ Editor maintains independent function lists). 3D graphs use the following Window variables.

| Variable | Description |
| :--- | :--- |
| eye $\theta$, eye $\phi$, <br> eye $\psi$ | Angles (always in degrees) used to view the graph. |

## Variable Description

xmin, xmax, Boundaries of the viewing cube.
ymin, ymax,
zmin, zmax
xgrid, ygrid The distance between xmin and xmax and between ymin and ymax is divided into the specified number of grids. The $\mathbf{z}(\mathbf{x}, \mathbf{y})$ equation is evaluated at each grid point where the grid lines (or grid wires) intersect.
The incremental value along $x$ and $y$ is calculated as:
$x$ increment $=\frac{x \max -x \min }{x \text { grid }}$
$y$ increment $=\frac{y m a x-y \min }{y g r i d}$

The number of grid wires is xgrid +1 and $y g r i d+1$. For example, when xgrid =14 and ygrid =14, the $x y$ grid consists of $225(15 \times 15)$ grid points.


## ncontour

The number of contours evenly distributed along the displayed range of $z$ values.

Note: If you enter a fractional number for xgrid or ygrid, it is rounded to the nearest whole number $\geq 1$. The 3D mode does not have scl Window variables, so you cannot set tick marks on the axes.

Standard values (set when you select 6:ZoomStd from the F2 Zoom toolbar menu) are:

| eye $=20$. |  | $x \min =-10$. |  |
| :--- | :--- | :--- | :--- |
| eymin $=-10$. |  | zmin $=-10$. |  |
| eye $=70$. | $x \max =10$. | ymax $=10$. | $z m a x=10$. |
| eye. | xgrid $=14$. | ygrid $=14$. | ncontour $=5$. |

You may need to increase the standard values for the grid variables (xgrid, ygrid) to ensure that enough points are plotted.

Note: Increasing the grid variables decreases the graphing speed.

## Setting the Graph Format

The Axes and Style formats are specific to the 3D graphing mode.

## Exploring a Graph

As in function graphing, you can explore a graph by using the following tools. Any displayed coordinates are shown in rectangular or cylindrical form as set in the graph format. In 3D graphing, cylindrical coordinates are shown when you use use:
(F1) 9

- or -
to set Coordinates $=$ POLAR.


## Tool For 3D Graphs:

Free-Moving The free-moving cursor is not available.
Cursor
[20 Zoom Works essentially the same as it does for function graphs, but remember that you are now using three dimensions instead of two.

- Only the following zooms are available:

2:ZoomIn, 3:ZoomOut, 5:ZoomSqr, 6:ZoomStd, A:ZoomFit, B:Memory, C:SetFactors

- Only $\mathbf{x}$ (xmin, xmax), $\mathbf{y}$ (ymin, ymax), and $\mathbf{z}$ (zmin, $z m a x$ ) Window variables are affected.
- The grid (xgrid, ygrid) and eye (eye $\theta$, eye $\phi$, eye $\psi$ ) Window variables are not affected unless you select 6:ZoomStd (which resets these variables to their standard values).

F3 Trace Lets you move the cursor along a grid wire from one grid point to the next on the 3D surface.

- When you begin a trace, the cursor appears at the midpoint of the xy grid.
- QuickCenter is available. At any time during a trace, regardless of the cursor's location, you can press ENTER to center the viewing cube on the cursor.
- Cursor movement is restricted in the x and y directions. You cannot move the cursor beyond the viewing cube boundaries set by $\mathbf{x m i n}, \mathbf{x m a x}, \mathbf{y m i n}$, and ymax.


## Tool

For 3D Graphs:
F5 Math Only 1:Value is available for 3D graphs. This tool displays
the $z$ value for a specified $x$ and $y$ value.
After selecting 1:Value, type the $x$ value and press ENTER.
Then type the $y$ value and press ENTER.
Note: During a trace, you can also evaluate $\mathbf{z}(\mathbf{x}, \mathbf{y})$. Type the x value and press ENTER; then type the y value and press ENTER.

## Moving the Cursor in 3D

When you move the cursor along a 3D surface, it may not be obvious why the cursor moves as it does. 3D graphs have two independent variables ( $\mathbf{x}, \mathbf{y}$ ) instead of one, and the $x$ and $y$ axes have a different orientation than other graphing modes.

## How to Move the Cursor

On a 3D surface, the cursor always follows along a grid wire.

| Cursor Key | Moves the cursor to the next grid point in the: |
| :---: | :--- |
| $(1)$ | Positive x direction |
| $(1)$ | Negative x direction |
| $\odot$ | Positive y direction |
| $\odot$ | Negative y direction |

Note: You can move the cursor only within the x and y boundaries set by Window variables xmin, xmax, ymin, and ymax.

Although the rules are straightforward, the actual cursor movement can be confusing unless you know the orientation of the axes.

In 2D graphing, the $x$ and $y$ axes always have the same orientation relative to the Graph screen.

In 3D graphing, $x$ and $y$ have a different orientation relative to the Graph screen. Also, you can rotate and/or elevate the viewing angle.



$$
\text { eye } \theta=20 \text { eye } \phi=70 \text { eye } \psi=0
$$

Note: To show the axes and their labels from the $\mathrm{Y}=$ Editor, Window Editor, or Graph screen, use:

- 1


## Simple Example of Moving the Cursor

The following graph shows a sloped plane that has the equation $z 1(x, y)=-(x+y) / 2$. Suppose you want to trace around the displayed boundary.

When you press F3, the trace cursor appears at the midpoint of the xy grid. Use the cursor pad to move the cursor to any edge.


By displaying and labeling the axes, you can more easily see the pattern in the cursor movement. To move grid points closer together, you can increase Window variables xgrid and ygrid.

When the trace cursor is on an interior point in the displayed plane, the cursor moves from one grid point to the next along one of the grid wires. You cannot move diagonally across the grid. Notice that the grid wires may not appear parallel to the axes.

## Example of the Cursor on a Hidden Surface

On more complex shapes, the cursor may appear as if it is not on a grid point. This is an optical illusion caused when the cursor is on a hidden surface.

For example, consider a saddle shape $z 1(x, y)=\left(x^{2}-y^{2}\right) / 3$. The following graph shows the view looking down the $y$ axis.

립 $=90$.
Eリヒ $\ddagger=70$.
튼낭․


$\times \times 9 \times 10=14$

10
$m a x=10$,
max $=16$


noontour= 5

Now look at the same shape at $10^{\circ}$ from the $x$ axis (eye $\left.\theta=10\right)$.


You can move the cursor so that it does not appear to be on a grid point.


If you cut away the front side, you can see the cursor is actually on a grid point on the hidden back side.

Note: To cut away the front of the saddle in this example, set $\mathbf{x m a x}=0$ to show only negative x values.

## Example of an "Off the Curve" Cursor

Although the cursor can move only along a grid wire, you will see many cases where the cursor does not appear to be on the 3D surface at all. This occurs when the $z$ axis is too short to show $\mathbf{z}(\mathbf{x}, \mathbf{y})$ for the corresponding x and y values.

For example, suppose you trace the paraboloid $z(x, y)=x^{2}+.5 y^{2}$ graphed with the indicated Window variables. You can easily move the cursor to a position such as:


Although the cursor is actually tracing the paraboloid, it appears off the curve because the trace coordinates:

- $\quad \mathbf{x c}$ and $\mathbf{y c}$ are within the viewing cube.
- but -
- $\quad \mathbf{z c}$ is outside the viewing cube.

Note: QuickCenter lets you center the viewing cube on the cursor's location. Simply press ENTER.

When $\mathbf{z c}$ is outside the $z$ boundary of the viewing cube, the cursor is physically displayed at $\mathbf{z m i n}$ or $\mathbf{z m a x}$ (although the screen shows the correct trace coordinates).

## Rotating and/or Elevating the Viewing Angle

In 3D graphing mode, the eye $\theta$ and eye $\phi$ Window variables let you set viewing angles that determine your line of sight. The eye $\psi$ Window variable lets you rotate the graph around that line of sight.

## How the Viewing Angle Is Measured

The viewing angle has three components:

- eye $\theta$ - angle in degrees from the positive $x$ axis.
- eye $\phi$ - angle in degrees from the positive $z$ axis.
- eye $\psi$ - angle in degrees by which the graph is rotated counter-clockwise around the line of sight set by eye $\theta$ and eye $\phi$.



Min=-1日。
$x \max =10$
xarid
분=-10,


$\operatorname{zmir}=-10$.
$\operatorname{cma}=10$
zmax=10;
noontomes.
Do not enter a ${ }^{\circ}$ symbol. For example, type 20,70 , and 0 , not $20^{\circ}, 70^{\circ}$ and $0^{\circ}$.

Note: When eye $\psi=0$, the $z$ axis is vertical on the screen. When eye $\psi=90$, the $z$ axis is rotated $90^{\circ}$ counterclockwise and is horizontal.

In the Window Editor ( $\bullet$ [Window]), always enter eye $\theta$, eye $\phi$, and eye $\psi$ in degrees, regardless of the current angle mode.

## Effect of Changing eye日 theta

The view on the Graph screen is always oriented along the viewing angle. From this point of view, you can change eye $\theta$ to rotate the viewing angle around the $z$ axis.

| $z 1(x, y)=\left(x^{3} y-y^{3} x\right) / 390$ | In this example eye $\phi=$ |
| :--- | :--- |
|  | 70 |



$$
\text { eye } \theta=20
$$



eye $\theta=50$

eye $\theta=80$


Note: This example increments eye $\theta$ by 30.

## Effect of Changing eye $\phi$ phi

By changing eye $\phi$, you can elevate your viewing angle above the xy plane. If $90<$ eye $\phi<$ 270, the viewing angle is below the xy plane.

| $z 1(x, y)=\left(x^{3} y-y^{3} x\right) / 390$ |  | In this example eye $\theta=$ 20 |
| :---: | :---: | :---: |
|  | $\text { eye } \phi=90$ |  |



$$
\text { eye } \phi=70
$$


eye $\phi=50$


Note: This example starts on the xy plane (eye $\phi=90$ ) and decrements eye $\phi$ by 20 to elevate the viewing angle.

## Effect of Changing eyew psi

The view on the Graph screen is always oriented along the viewing angles set by eye $\theta$ and eye $\phi$. You can change eye $\psi$ to rotate the graph around that line of sight.

Note: During rotation, the axes expand or contract to fit the screen's width and height. This causes some distortion as shown in the example.

| $z 1(x, y)=\left(x^{3} y-y^{3} x\right) / 390$ | eye $\psi=0$ |
| :---: | :---: |
| In this example, |  |
| eye $\theta=20$ and eye $\phi=70$ |  |



$$
\text { eуе } \psi=45
$$



еуе $\psi=90$


When eye $\psi=0$, the $z$ axis runs the height of the screen.


When eye $\psi=90$, the $z$ axis runs the width of the screen.


As the $z$ axis rotates $90^{\circ}$, its range ( -10 to 10 in this example) expands to almost twice its original length. Likewise, the x and y axes expand or contract.

## From the Home Screen or a Program

The eye values are stored in the system variables eye $\theta$, eye $\phi$, and eye $\psi$. You can access or store to these variables as necessary.

To type $\phi$ or $\psi$, press $\square \square$ alpha [F] or $\rightarrow \square$, respectively. You can also press [2nd [CHAR] and use the Greek menu.

## Animating a 3D Graph Interactively

After plotting any 3D graph, you can change the viewing angle interactively by using the cursor.

## The Viewing Orbit

When using $(\uparrow$ and ( © to animate a graph, think of it as moving the viewing angle along its "viewing orbit" around the graph.

Moving along this orbit can cause the $z$ axis to wobble slightly during the animation.


Note: The viewing orbit affects the eye Window variables in differing amounts.

## Animating the Graph

To: Do this:

Animate the graph incrementally. Press and release the cursor quickly.
Move along the viewing orbit. (1) or ( ${ }^{(1)}$

Change the viewing orbit's elevation. $\Theta$ or $\odot$
(primarily increases or decreases eyeф)
Animate the graph continuously. Press and hold the cursor for about 1 second, and then release it. To stop, press ESC], ENTER, ON, or $\square$ [ $\quad$ ] (space).

Change between 4 animation Press $\square$ or $\square$. speeds (increase or decrease the incremental changes in the eye Window variables).

Change the viewing angle of a non- Press $\mathrm{X}, \mathrm{Y}$ or Z , respectively. animated graph to look along the x , $y$, or $z$ axis.
Return to the initial eye angle values. Press 0 (zero).

Notes: If the graph is shown in expanded view, it returns to normal view automatically when you press a cursor key.

- After animating the graph, you can stop and then re-start the animation in the same direction by pressing: ENTER or alpha [-]
- During an animation, you can switch to the next graph format style by pressing: I
- You can view a graphic that shows the eye angles.


## Animating a Series of Graph Pictures

You can also animate a graph by saving a series of graph pictures and then flipping (or cycling) through those pictures. Refer to "Animating a Series of Graph Pictures" Additonal Graphing Topics. This method gives you more control over the Window variable values, particularly eye $\psi$, which rotates the graph.

## Changing the Axes and Style Formats

With its default settings, the TI-89 Titanium displays hidden surfaces on a 3D graph but does not display the axes. However, you can change the graph format at any time.

## Displaying the GRAPH FORMATS Dialog Box

From the $\mathrm{Y}=$ Editor, Window Editor, or Graph screen, press:
F1 9

- The dialog box shows the current graph format settings.
- To exit without making a change, press ESC.


To change any of these settings, use the same procedure that you use to change other types of dialog boxes, such as the MODE dialog box.

## Examples of Axes Settings

To display the valid Axes settings, highlight the current setting and press (1).

### 1.0IFF 2:BXES

$$
z 1(x, y)=x^{2}+.5 y^{2}
$$

- AXES - Shows standard xyz axes.

- BOX - Shows 3-dimensional box axes.

The edges of the box are determined by the Window variables xmin, xmax, etc.


In many cases, the origin $(0,0,0)$ is inside the box, not at a corner. For example, if $\mathbf{x} \min =\mathbf{y m i n}=\mathbf{z m i n}=-10$ and $\mathbf{x m a x}=\mathbf{y m a x}=\mathbf{z m a x}=10$, the origin is at the center of the box.

Note: Setting Labels = ON is helpful when you display either type of 3D axes.

## Examples of Style Settings

Note: WIRE FRAME is faster to graph and may be more convenient when you're experimenting with different shapes.

To display the valid Style settings, highlight the current setting and press (1).
 to differentiate the two sides of the 3D shape.

Later sections in this module describe CONTOUR LEVELS, WIRE AND COUNTOUR, and implicit plots.

## Be Aware of Possible Optical Illusions

The eye angles used to view a graph (eye $\theta$, eye $\phi$, and eye $\psi$ Window variables) can result in optical illusions that cause you to lose perspective on a graph. Typically, most optical illusions occur when the eye angles are in a negative quadrant of the coordinate system.

Optical illusions may be more noticeable with box axes. For example, it may not be immediately obvious which is the "front" of the box.


Note: The first two examples show the graphs as displayed on the screen. The second two examples use artificial shading (which is not displayed on the screen) to show the front of the box.

To minimize the effect of optical illusions, use the GRAPH FORMATS dialog box to set Style $=$ HIDDEN SURFACE .

## Contour Plots

In a contour plot, a line is drawn to connect adjacent points on the 3D graph that have the same $z$ value. This module discusses the CONTOUR LEVELS and WIRE AND CONTOUR graph format styles.

## Selecting the Graph Format Style

In 3D graphing mode, define an equation and graph it as you would any 3D equation, with the following exception. Display the GRAPH FORMATS dialog box by pressing F1 9 from the $\mathrm{Y}=$ Editor, Window editor, or Graph screen. Then set:

```
Style = CONTOUR LEVELS
- Or -
Style = WIRE AND CONTOUR
```

| GFiAFH FDFl-4ATS |  |  |
| :---: | :---: | :---: |
| Codrdindtes FECT 9 |  |  |
| HXe5........... DFF ${ }^{\text {¢ }}$ |  |  |
| Labicls........ DFF ${ }^{\text {\% }}$ |  |  |
| sture $1:$ WIRE FRHPE |  |  |
|  |  |  |
|  |  |  |
| Entst |  |  |

- For CONTOUR LEVELS, only the contours are shown.
- The viewing angle is set initially so that you are viewing the contours by looking down the $z$ axis. You can change the viewing angle as necessary.
- The graph is shown in expanded view. To switch between expanded and normal view, press $\boxtimes$.
- The Labels format is set to OFF automatically.
- For WIRE AND CONTOUR, the contours are drawn on a wire frame view. The viewing angle, view (expanded or normal), and Labels format retain their previous settings.


## Notes:

- From the Graph screen, you can switch from one graph format style to the next (skipping IMPLICIT PLOT) by pressing: T
- Pressing:
$\square$
to select CONTOUR LEVELS does not affect the viewing angle, view, or Labels
format as it does if you use:
$\bullet \square$

| Style | $z 1(x, y)=\left(x^{3} y-y^{3} x\right) / 390 \quad z 1(x, y)=x^{2}+.5 y^{2}-5$ |
| :--- | :--- |
| Looking down $z$ axis |  |
| CONTOUR |  |

Using eye $\theta=20$, еуе $\phi=70$, еуе $\psi=0$


WIRE AND CONTOUR


Note: These examples use the same $x, y$, and $z$ Window variable values as a ZoomStd viewing cube. If you use ZoomStd, press $Z$ to look down the $z$ axis. Do not confuse the contours with the grid lines. The contours are darker.

## How Are Z Values Determined？

You can set the ncontour Window variable（ $\square$［wiNDOW］）to specify the number of contours that will be evenly distributed along the displayed range of $z$ values，where：

$$
\text { increment }=\frac{z m a x-z m i n}{\text { ncontour }+1}
$$

The $z$ values for the contours are：

```
zmin + increment
zmin + 2(increment)
zmin + 3(increment)
zmin + ncontour(increment)
```


気时 $=70$
 $\times \mathrm{m}: \times 10$ ． $\times \mathrm{gri} \mathrm{i}=14$ ． ㅂmin＝－10． ymax $=10$. ygrid＝14．立in＝－1家： $2 m a x=10$ ． noontour＝5．

The default is 5 ．You can set this to 0 through 20.

If ncontour＝5 and you use the standard viewing window（ $\mathbf{z m i n}=-\mathbf{1 0}$ and $\mathbf{z m a x = 1 0}$ ），the increment is 3.333 ．Five contours are drawn for $z=-6.666,-3.333,0,3.333$ ，and 6.666 ．

Note，however，that a contour is not drawn for a $z$ value if the 3D graph is not defined at that $z$ value．

## Drawing a Contour for the Z Value of a Selected Point Interactively

If a contour graph is currently displayed, you can specify a point on the graph and draw a contour for the corresponding $z$ value.

1. To display the Draw menu, press: 2nd [F6]

2. Select 7:Draw Contour.
3. Either:

- Type the point's $x$ value and press ENTER, and then type the $y$ value and press ENTER.
- or -
- Move the cursor to the applicable point. (The cursor moves along the grid lines.) Then press ENTER.

For example, suppose the current graph is $z 1(x, y)=x^{2}+.5 y^{2}-5$. If you specify $x=2$ and $y=3$, a contour is drawn for $z=3.5$.

Note: Any existing contours remain on the graph. To remove the default contours, display the Window editor ( $\bullet$ [WINDOW]) and set ncontour=0.

## Drawing Contours for Specified Z Values

From the Graph screen, display the Draw menu and then select 8:DrwCtour. The Home screen is displayed automatically with DrwCtour in the entry line. You can then specify one or more $z$ values individually or generate a sequence of $z$ values.

Some examples are:

| DrwCtour 5 | Draws a contour for $\mathrm{z}=5$. |
| :---: | :---: |
| DrwCtour \{1,2,3\} | Draws contours for $\mathrm{z}=1,2$, and 3 . |
| DrwCtour <br> $\operatorname{seq}(\mathrm{n}, \mathrm{n},-10,10,2)$ | Draws contours for a sequence of $z$ values from -10 through 10 in steps of $2(-10,-8,-6$, etc.). |

Note: To remove the default contours, use $\square$ [WINDOW] and set ncontour=0.
The specified contours are drawn on the current 3D graph. (A contour is not drawn if the specified $z$ value is outside the viewing cube or if the 3D graph is not defined at that $z$ value.)

## Notes about Contour Plots

## For a contour plot:

- You can use the cursor keys to animate the contour plot.
- You cannot trace ([F3) the contours themselves. However, you can trace the wire frame as seen when Style=WIRE AND CONTOUR.
- It may take awhile to evaluate the equation initially.
－Because of possible long evaluation times，you first may want to experiment with your 3D equation by using Style＝WIRE FRAME．The evaluation time is much shorter．Then，after you＇re sure you have the correct Window variable values， display the Graph Formats dialog box and set Style＝CONTOUR LEVELS or WIRE AND CONTOUR．
－ 1


## Example：Contours of a Complex Modulus Surface

The complex modulus surface given by $\mathbf{z}(\mathbf{a}, \mathbf{b})=\mathbf{a b s}(\mathbf{f}(\mathbf{a} \mathbf{+} \mathbf{b} \mathbf{i}))$ shows all the complex zeros of any polynomial $\mathbf{y = f}(\mathbf{x})$ ．

## Example

In this example，let $f(x)=x^{3}+1$ ．By substituting the general complex form $x+y i$ for $x$ ，you can express the complex surface equation as $z(x, y)=a b s\left((x+y i)^{3}+1\right)$ ．

1．Use MODE to set Graph＝3D．
2．Press $\bullet[Y=]$ ，and define the equation：
$\mathrm{z} 1(\mathrm{x}, \mathrm{y})=\mathrm{abs}\left((\mathrm{x}+\mathrm{y} *)^{\wedge}{ }^{\wedge} 3+1\right)$

|  <br> $\checkmark{ }_{z 1}=(x+y \cdot \mathfrak{i})^{3}+1 \mid$ <br>  <br>  |
| :---: |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

3．Press $\square$［window］，and set the Window variables as shown．

틉ㅂ $=-90$.它ジも二回。

$\times m i r=-1.5$
$\times \infty-\exists x=1.5$

빈ำ二 1 －



피… $\times 2=2$

4. Display the Graph Formats dialog box:
$\square$ Turn on the axes, set
Style = CONTOUR LEVELS, and return to the Window editor.

5. Press [GRAPH] to graph the equation.

It will take awhile to evaluate the graph; so be patient. When the graph is displayed, the complex modulus surface touches the xy plane at exactly the complex zeros of the polynomial:
$-1, \frac{1}{2}+\frac{\sqrt{3}}{2} i$, and $\frac{1}{2}-\frac{\sqrt{3}}{2} i$
6. Press [F3, and move the trace cursor to the zero in the fourth quadrant.

The coordinates let you estimate $.428-.857 i$ as the zero.
7. Press ESC. Then use the cursor keys to animate the graph and view it from different eye angles.


The zero is precise when $\mathrm{z}=0$.


This example shows eye $\theta=70$, eуe $\phi=70$, and eye $\psi=0$.

## Notes:

- For more accurate estimates, increase the xgrid and ygrid Window variables. However, this increases the graph evaluation time.
- When you animate the graph, the screen changes to normal view. Use $\boxtimes$ to toggle between normal and expanded views.


## Implicit Plots

An implicit plot is used primarily as a way to graph 2D implicit forms that cannot be graphed in function graphing mode. Technically, an implicit plot is a 3D contour plot with a single contour drawn for $\mathbf{z = 0}$ only.

## Explicit and Implicit Forms

In 2D function graphing mode, equations have an explicit form $\mathbf{y}=\mathrm{f}(\mathbf{x})$, where y is unique for each value of $x$.

Many equations, however, have an implicit form $f(\mathbf{x}, \mathbf{y})=g(\mathbf{x}, \mathbf{y})$, where you cannot explicitly solve for y in terms of x or for x in terms of y .

$y$ is not unique for each $x$, so you cannot graph this in function graphing mode.

By using implicit plots in 3D graphing mode, you can graph these implicit forms without solving for y or x .

Rearrange the implicit form as an equation set to zero.

In the $Y=$ Editor, enter the non-zero side of the equation. This is valid because an implicit plot automatically sets the equation equal to zero.

For example, given the ellipse equation shown to the right, enter the implicit form in the $\mathrm{Y}=$ Editor.

$$
f(x, y)-g(x, y)=0
$$

$$
z 1(x, y)=f(x, y)-g(x, y)
$$

$$
\begin{gathered}
\text { If } x^{2}+.5 y^{2}=30 \\
\text { then } z 1(x, y)=x^{2}+.5 y^{2}-30 .
\end{gathered}
$$

Notes: You can also graph many implicit forms if you either:

- Express them as parametric equations.
- Break them into separate, explicit functions.


## Selecting the Graph Format Style

In 3D graphing mode, define an appropriate equation and graph it as you would any 3D equation, with the following exception. Display the GRAPH FORMATS dialog box from the $Y=$ Editor, Window editor, or Graph screen:

- 1

Note: From the Graph screen, you can switch to the other graph format styles by and then set Style $=$ IMPLICIT PLOT.
GFiAFH FDFMATS
pressing:
$\square$

However, to return to IMPLICIT PLOT press:
$\rightarrow \square$

- The viewing angle is set initially so that you are viewing the plot by looking down the $z$ axis. You can change the viewing angle as necessary.
- The plot is shown in expanded view. To switch between expanded and normal view, press ${ }^{\text {® }}$.
- The Labels format is set to OFF automatically.

| Style | $\begin{gathered} x^{2}-y^{2}=4 \\ z 1(x, y)=x^{2}-y^{2}-4 \end{gathered}$ | $\begin{gathered} \sin (x)+\cos (y)=e(x * y) \\ z 1(x, y)=\sin (x)+\cos (y)-e(x * y) \end{gathered}$ |
| :---: | :---: | :---: |
| IMPLICIT PLOT |  |  |

Note: These examples use the same $x, y$, and $z$ Window variable values as a ZoomStd viewing cube. If you use ZoomStd, press $Z$ to look down the $z$ axis.

## Notes About Implicit Plots

For an implicit plot:

- The ncontour Window variable has no affect. Only the $\mathbf{z = 0}$ contour is drawn, regardless of the value of ncontour. The displayed plot shows where the implicit form intersects the xy plane.
- You can use the cursor keys to animate the plot.
- You cannot trace ([F3) the implicit plot itself. However, you can trace the unseen wire frame graph of the 3D equation.
- It may take awhile to evaluate the equation initially.
- Because of possible long evaluation times, you first may want to experiment with your 3D equation by using Style=WIRE FRAME. The evaluation time is much shorter. Then, after you're sure you have the correct Window variable values, set Style=IMPLICIT PLOT.
$\bullet$
$\square$


## Example: Implicit Plot of a More Complicated Equation

You can use the IMPLICIT PLOT graph format style to plot and animate a complicated equation that cannot be graphed otherwise. Although it may take a long time to evaluate such a graph, the visual results can justify the time required.

## Example

Graph the equation $\sin \left(x^{4}+y-x^{3} y\right)=.1$ ．

1．Use MODE to set Graph＝3D．
2．Press $\square[Y=]$ ，and define the equation：
$z 1(x, y)=\sin \left(x^{\wedge} 4+y-x^{\wedge} 3 y\right)-.1$


3．Press $\downarrow$［WINDOW］，and set the Window variables as shown．

> 르라二気
> xirio 10
> $\times m \equiv x=10$
> x 9 r id=14.
> - miro
> $-3 \times 10$

> zMir=-1日。
> $\geq \times 3 x=10$,

4．Press：
$\square$ DTurn on the axes，set
Style $=$ IMPLICIT PLOT，and return to the Window editor．


5．Press $\square$［GRAPH to graph the equation． It will take awhile to evaluate the graph； so be patient．


The graph shows where $\sin \left(x^{4}+y-x^{3} y\right)=.1$
6. Use the cursor keys to animate the graph and view it from different eye angles.


Note: For more detail, increase the xgrid and ygrid Window variables. However, this increases the graph evaluation time.

In expanded view, this example shows
eye $\theta=-127.85$,
eye $\phi=52.86$, and
еуе $\psi=-18.26$.

Note: When you animate the graph, the screen changes to normal view. Press $\boxtimes$ to switch between normal and expanded views.

## Differential Equation Graphing

## Overview of Steps in Graphing Differential Equations

To graph differential equations, use the same general steps used for $\mathbf{y}(\mathbf{x})$ functions as described in Basic Function Graphing. Any differences are described on the following pages.

## Graphing Differential Equations

1. Set Graph mode (MODE) to DIFF EQUATIONS. Also set Angle mode, if necessary.

2. Define equations and, optionally, initial conditions on $Y=$ Editor ( $\Delta[\mathrm{Y}=]$ ).
3. Select (F4) which defined functions to graph.
Note: To turn off any stat data plots, press F5 5 or use F4 to deselect them.

4. Set the display style for a function. [2nd [F6]

## 路



- or -

Note: The Fields format is critical, depending on the order of the equation.
6. Set the axes as applicable, depending on the Fields format.
[2nd [F7]
Note: Valid Axes settings depend on the Fields format.
7. Define the viewing window ( $\square$ [WINDOW]).

Note: Depending on the Solution Method and Fields formats, different Window variables are displayed. F2 Zoom also changes the viewing window.

to 0 =0,
tmax=10. $t=t e p=1$ xmin=-10, $x \max =110$.人Ec1=10,
 $\mathrm{y}=\mathrm{cl}=10$ ncurves=0 fiftol=201
8. Graph the selected functions ( $\bullet$ [GRAPH]).


## Differences in Diff Equations and Function Graphing

This module assumes that you already know how to graph $\mathbf{y}(\mathbf{x})$ functions as described in Basic Function Graphing. This section describes the differences.

## Setting the Graph Mode

Use MODE to set Graph = DIFF EQUATIONS before you define differential equations or set Window variables. The $\mathrm{Y}=$ Editor and the Window Editor let you enter information for the current Graph mode setting only.

## Defining Differential Equations on the $\mathrm{Y}=$ Editor



Note: You can use the Define command from the Home screen to define functions and equations.

When entering equations in the $Y=$ Editor, do not use $\mathbf{y}(\mathrm{t})$ formats to refer to results. For example:

Do not use implied multiplication between a variable and parenthetical expression. If you do, it is treated as a function call.

```
Enter: y1' = .001y1*(100-y1)
Not: y1' = .001y1(t)*(100-y1(t))
```

Only 1st-order equations can be entered in the $Y=$ Editor. To graph 2nd- or higher-order equations, you must enter them as a system of 1 st-order equations.

Detailed information is available on setting initial conditions.

## Selecting Differential Equations



Important: Selecting y1' will graph the y1 solution curve, not the derivative y1', depending on the axis setting.

## Selecting the Display Style

With the Style menu, only the Line, Dot, Square, Thick, Animate, and Path styles are available. Dot and Square mark only those discrete values (in tstep increments) at which a differential equation is plotted.
[0] [2nd [F6]

## Setting Graph Formats

From the $\mathrm{Y}=$ Editor, Window Editor, or Graph screen, press:
F1 9
—or - $\square$


The formats affected by differential equations are:

| Graph format | Description |
| :--- | :--- |
| Graph Order | Not available. |
| Solution Method | Specifies the method used to solve the differential <br> equations. |
|  | - $\quad$ RK - Runge-Kutta method. For information about the |
|  | $\quad$algorithm used for this method, refer to the Technical |
|  | Reference module. |
|  | EULER - Euler method. |
|  | The method lets you choose either greater accuracy or <br> speed. Typically, RK is more accurate than EULER but <br> takes longer to find the solution. |
|  |  |

Specifies whether to draw a field for the differential equation.

- SLPFLD - Draws a slope field for only one 1st-order equation, with $t$ on the $x$ axis and the solution on the $y$ axis.
- DIRFLD - Draws a direction field for only one 2ndorder equation (or system of two 1 st-order equations), with axes determined by the custom axes settings.
- FLDOFF - Does not display a field. This is valid for equations of any order, but you must use it for 3rd- or higher-order. You must enter the same number of initial conditions for all equations in the $\mathrm{Y}=$ Editor.

Important: The Fields graph format is critical in successfully graphing differential equations.

Note: If you press ENTER while a slope or direction field is being drawn, the graph pauses after the field is drawn but before the solutions are plotted. Press ENTER gain to continue. To cancel graphing, press 0 N .

## Setting Axes

In the $Y=$ Editor, Axes may or may not be available, depending on the current graph format.

If it is available, you can select the axes that are used to graph the differential equations. [2nd [F7]


| Axes | Description |
| :--- | :--- |
| TIME | Plots $t$ on the $x$ axis and $y$ (the solutions to the selected <br> differential equations) on the $y$ axis. |
| CUSTOM | Lets you select the $x$ and $y$ axes. |

## Window Variables

Differential equation graphs use the following Window variables. Depending on the Solution Method and Fields graph formats, not all of these variables are listed in the Window Editor ( $\quad$ [WINDOW]) at the same time.

| Variable | Description |
| :---: | :---: |
| t0 | Time at which the initial conditions entered in the $Y=$ Editor occur. You can set t0 in the Window Editor and $Y=$ Editor. (If you set t0 in the $Y=$ Editor, tplot is set to the same value automatically.) |
| tmax, tstep | Used to determine the $t$ values where the equations are plotted: $\begin{aligned} & \mathrm{y}^{\prime}(\mathrm{t} 0) \\ & \mathrm{y}^{\prime}(\mathrm{t} 0+\text { tstep }) \\ & \mathrm{y}^{\prime}(\mathrm{t} 0+2 * \text { tstep }) \\ & \ldots . \text { not to exceed ... } \\ & \mathrm{y}^{\prime}(\text { tmax }) \end{aligned}$ <br> If Fields = SLPFLD, tmax is ignored. Equations are plotted from to to both edges of the screen in tstep increments. |

## Variable Description

tplot $\quad$ First $t$ value plotted. If this is not a tstep increment, plotting begins at the next tstep increment. In some situations, the first points evaluated and plotted starting at t0 may not be interesting visually. By setting tplot greater than t0, you can start the plot at the interesting area, which speeds up the graphing time and avoids unnecessary clutter on the Graph screen.

Note: If $\mathbf{t m a x}<\mathbf{t 0}$, tstep must be negative. If Fields=SLPFLD, tplot is ignored and is assumed to be the same as to.

| Variable | Description |
| :--- | :--- |
| xmin, $\mathbf{x m a x}$, <br> ymin, $\mathbf{y m a x}$ | Boundaries of the viewing window. |
| $\mathbf{x s c l}, \mathbf{y s c l}$ | Distance between tick marks on the x and y axes. |
| ncurves | Number of solution curves (0 through 10) that will be <br> drawn automatically if you do not specify an initial <br> condition. By default, ncurves $=\mathbf{0}$. |


| Variable | Description |
| :---: | :---: |
|  | When ncurves is used, $\mathbf{t 0}$ is set temporarily at the middle of the screen and initial conditions are distributed evenly along the $y$ axis, where: |
|  | $\text { increment }=\frac{y \max -y \min }{\text { ncurves }+1}$ |
|  | ```The \(y\) values for the initial conditions are: ymin + increment ymin \(+2 *\) (increment) ymin + ncurves*(increment)``` |
| diftol | (Solution Method = RK only) Tolerance used by the RK method to help select a step size for solving the equation; must be $\geq 1 \mathrm{E}-14$. |
| fldres | (Fields $=$ SLPFLD or DIRFLD only) Number of columns ( 1 through 80) used to draw a slope or direction field across the full width of the screen. |
| Estep | (Solution Method = EULER only) Euler iterations between tstep values; must be an integer >0. For more accuracy, you can increase Estep without plotting additional points. |
| dtime | (Fields = DIRFLD only) Point in time at which a direction field is drawn. |

Standard values (set when you select 6:ZoomStd from the F2 Zoom toolbar menu) are:

| 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 $=14$. |

You may need to change the standard values for the $t$ variables to ensure that sufficient points are plotted.

## The fidpic System Variable

When a slope or direction field is drawn, a picture of the field is stored automatically to a system variable named fldpic. If you perform an operation that regraphs the plotted equations but does not affect the field, the TI-89 Titanium reuses the picture in fldpic instead of having to redraw the field. This can speed up the regraphing time significantly.
fldpic is deleted automatically when you exit the differential equation graphing mode or when you display a graph with Fields = FLDOFF.

## Exploring a Graph

As in function graphing, you can explore a graph by using the following tools. Any displayed coordinates are shown in rectangular or polar form as set in the graph format.

| Tool | For Differential Equation Graphs: |
| :--- | :--- |
| Free-Moving | Works just as it does for function graphs. |

Cursor

| Tool | For Differential Equation Graphs: |
| :---: | :---: |
| (F2) Zoom | Works just as it does for function graphs. <br> - Only $\mathbf{x}$ (xmin, $\mathbf{x m a x}, \mathbf{x s c l})$ and $\mathbf{y}$ (ymin, ymax, $\mathbf{y s c l}$ ) Window variables are affected. <br> - The $\mathbf{t}$ Window variables (t0, tmax, tstep, tplot) are not affected unless you select 6:ZoomStd (which sets all Window variables to their standard values). |
| F3 Trace | Lets you move the cursor along the curve one tstep at a time. To move approximately ten plotted points at a time, press 2nd (1) or 2nd (1). <br> If you enter initial conditions in the $Y=$ Editor or let the ncurves Window variable plot curves automatically, you can trace the curves. If you use: <br> 2nd [F8] <br> IC from the Graph screen to select initial conditions interactively, you cannot trace the curves. <br> QuickCenter applies to all directions. If you move the cursor off the screen (top or bottom, left or right), press ENTER to center the viewing window on the cursor location. Use $\Theta$ or $\Theta$ to view results on all plotted curves. |
| F5] Math | Only 1:Value is available. <br> - With TIME axes, the $\mathbf{y}(\mathrm{t})$ solution value (represented by $y c$ ) is displayed for a specified $t$ value. <br> - With CUSTOM axes, the values that correspond to x and y depend on the axes you choose. |

Note: During a trace, you can move the cursor to a particular point by typing a value for $t$ and pressing ENTER. You can use QuickCenter at any time during a trace, even if the cursor is still on the screen.

## Setting the Initial Conditions

You can enter initial conditions in the $\mathrm{Y}=$ Editor, let the TI-89 Titanium calculate initial conditions automatically, or select them interactively from the Graph screen.

## Entering Initial Conditions in the $\mathbf{Y}=$ Editor

You can specify one or more initial conditions in the $Y=$ Editor. To specify more than one, enter them as a list enclosed in braces \{ \} and separated by commas.

To enter initial conditions for the $\mathbf{y 1}$ ' equation, use the yi1 line, etc.

To specify when the initial conditions occur, use t0. This is also the first $t$ evaluated for the graph.
To graph a family of solutions, enter a list of initial conditions.


For a 2 nd- or higher-order differential equation, you must define a system of 1st-order equations in the $\mathrm{Y}=$ Editor.

If you enter initial conditions, you must enter the same number of initial conditions for each equation in the system. Otherwise, a Dimension error occurs.

| Fit ${ }^{\text {Fots }}$ |  |  |
| :---: | :---: | :---: |
| $\begin{aligned} & \text { FLDTS } \\ & t=0, \end{aligned}$ |  |  |
|  |  |  |
|  |  |  |
| yil=\% 0.5 |  |  |
| -42'=- 1 |  |  |
|  |  |  |

## If You Do Not Enter an Initial Condition in the Y= Editor

If you do not enter initial conditions, the ncurves Window variable ( $\square$ [winDOW]) specifies the number of solution curves graphed automatically. By default, ncurves $=\mathbf{0}$. You can enter a value from 0 through 10. However, the Fields graph format and the Axes setting determine whether ncurves is used.

| f Fields $=$ | Then: |
| :--- | :--- |
| SLPFLD | Uses ncurves, if not set to 0, to graph curves. |
| DIRFLD | Ignores ncurves. Does not graph any curves. |
| FLDOFF | Uses ncurves if Axes $=$ TIME (or if Axes = Custom and the <br> x axis is t ). Otherwise, a Diff Eq setup error occurs. |

When ncurves is used, $\mathbf{t 0}$ is set temporarily at the middle of the Graph screen. However, the value of t 0 as set in the $\mathrm{Y}=$ Editor or Window Editor is not changed.

## Notes:

- Without entering initial conditions, use SLPFLD (with ncurves=0) or DIRFLD to display a slope or direction field only.
- SLPFLD is for a single 1st-order equation only. DIRFLD is for a 2 nd-order equation (or system of two 1st-order equations) only.


## Selecting an Initial Condition Interactively from the Graph Screen

When a differential equation is graphed (regardless of whether a solution curve is displayed), you can select a point on the Graph screen and use it as an initial condition.
If Fields = Do this:

SLPFLD Press:

- or -

DIRFLD
[围 2nd [F8]
Specify an initial condition. Either:

- Move the cursor to the applicable point and press ENTER.
- or -
- For each of the two coordinates, type a value and press ENTER.
- For SLPFLD (1st-order only), enter values for $\mathbf{t 0}$ and y (t0).
- For DIRFLD (2nd-order or system of two 1st-order equations only), enter values for both $\mathbf{y}(\mathbf{t 0})$ initial conditions, where $\mathbf{t 0}$ is the value set in the $\mathrm{Y}=$ Editor or Window Editor.
A circle marks the initial condition and the solution curve is drawn.

| If Fields $=$ | Do this: |
| :--- | :--- |
| FLDOFF | • Press: |
|  |  |
|  | 囲 |

You are prompted to select the axes for which you want to enter initial conditions.


Your selections will be used as the axes for the graph.

- You can accept the defaults or change them. Then press ENTER.
- Specify an initial condition as described for SLPFLD or DIRFLD.

Note: With SLPFLD or DIRFLD, you can select initial conditions interactively regardless of whether you enter initial conditions in the $\mathrm{Y}=$ Editor. With FLDOFF, you can select initial conditions interactively. However, if three or more equations are entered, you must enter a single value (not a list) as the initial condition for each equation in the $Y=$ Editor. Otherwise, a Dimension error occurs when graphing.

## Note about Tracing a Solution Curve

When you enter initial conditions in the $\mathrm{Y}=$ Editor or let ncurves graph solution curves automatically, you can use [F3] to trace the curves. However, you cannot trace a curve drawn by selecting an initial condition interactively. These curves are drawn, not plotted.

## Defining a System for Higher-Order Equations

In the $\mathrm{Y}=$ Editor, you must enter all differential equations as 1st-order equations. If you have an nth-order equation, you must transform it into a system of $n 1$ st-order equations.

## Transforming an Equation into a 1st-Order System

A system of equations can be defined in various ways, but the following is a general method.

1. Rewrite the original differential equation as necessary.
a) Solve for the highest-ordered derivative.

$$
\begin{aligned}
& \hline y^{\prime \prime}+y^{\prime}+\mathrm{y}=\mathrm{e}^{\mathrm{x}} \\
& \mathrm{y}^{\prime \prime}=e^{\mathrm{x}}-\mathrm{y}^{\prime}-\mathrm{y} \\
& \mathrm{y}^{\prime \prime}=e^{\mathrm{t}}-\mathrm{y}^{\prime}-\mathrm{y} \\
& \hline
\end{aligned}
$$

b) Express it in terms of $y$ and $t$.
c) On the right side of the equation only, substitute to eliminate any references to derivative values.

Note: To produce a 1st-order equation, the right side must contain non-derivative variables only.

| In place of: | Substitute: |  |
| :---: | :---: | :---: |
| y | y1 | $\mathrm{y}^{\prime \prime}=e^{\mathrm{t}}-\mathrm{y} 2-\mathrm{y} 1$ |
| $y^{\prime}$ | y2 |  |
| y" | y3 | - Do not substitute |
| y'" | y4 | on the left side at |
| y (4) | y5 | this time. |
| ! | ! |  |

d) On the left side of the equation, substitute for the derivative value as shown below.

| In place of: | Substitute: |  |
| :---: | :---: | :---: |
|  | $\mathrm{y}^{\prime}$ |  |
| $\mathrm{y}^{\prime}$ | $\mathrm{y}^{\prime}$ | $\mathrm{y} 2^{\prime}=e^{\mathrm{t}}-\mathrm{y} 2-\mathrm{y} 1$ |
| $\mathrm{y}^{\prime \prime}$ | $\mathrm{y}^{\prime}$ |  |
| $\mathrm{y}^{\prime \prime \prime}$ | $\mathrm{y}^{\prime}$ |  |
| $\mathrm{y}(4)$ | $\vdots$ |  |
| $\vdots$ | $\vdots$ |  |

2. On the applicable lines in the $\mathrm{Y}=$ Editor, define the system of equations as:
y1' = y2
y2' = y3
y3' = y4

- up to -

|  |
| :---: |
| FLITES |
| $\pm{ }^{t} 0=0$, |
| 11 $1=$ |
| - $42=e^{t}-42-91$ |

yn ' = your nth-order equation
Note: Based on the above substitutions, the $y^{\prime}$ lines in the $Y=$ Editor represent:

$$
\begin{aligned}
& y 1^{\prime}=y ' \\
& \text { y2' = y" } \\
& \text { etc. }
\end{aligned}
$$

Therefore, this example's 2 nd-order equation is entered on the y2' line.
In a system such as this, the solution to the y 1 ' equation is the solution to the nth-order equation. You may want to deselect any other equations in the system.

## Example of a 2nd-Order Equation

The 2nd-order differential equation $y "+y=0$ represents a simple harmonic oscillator. Transform this into a system of equations for the $Y=$ Editor. Then, graph the solution for initial conditions $y(0)=0$ and $y^{\prime}(0)=1$.

## Example

1. Press MODE and set Graph=DIFF EQUATIONS.
2. Define a system of equations for the 2 ndorder equation.

Rewrite the equation and make the necessary substitutions.

$$
\begin{aligned}
& y^{\prime \prime \prime}+\mathrm{y}=0 \\
& \mathrm{y}^{\prime \prime}=-\mathrm{y} \\
& \mathrm{y}^{\prime \prime}=-\mathrm{y} 1 \\
& \mathrm{y} 2^{\prime}=-\mathrm{y} 1
\end{aligned}
$$

yi1 is the initial condition for $\mathrm{y}(0)$.
4. Enter the initial conditions:
yi1=0 and yi2=1
Note: $\mathbf{t 0}$ is the time at which the initial conditions occur. It is also the first t evaluated for the graph. By default, $\mathbf{t 0 = 0}$.

yi2 is the initial condition for $\mathrm{y}^{\prime}(0)$.
5. Press:

F1 9

- or
© $\square \square$ and set Axes = ON, Labels = OFF, Solution Method = RK, and Fields = DIRFLD.

Important: For 2nd-order equations, you must set Fields=DIRFLD or FLDOFF.
6. In the $\mathrm{Y}=$ Editor, press:
[-2nd [F7]and make sure
Axes $=$ CUSTOM with $\mathbf{y 1}$ and y 2 as the axes.


Important: Fields=DIRFLD cannot plot a time axis. An Invalid Axes error occurs if Axes=TIME or if $t$ is set as a CUSTOM axis.
7. In the Window Editor ( $\square$ [wiNDOW]), set the Window variables.

| $\mathrm{t} 0=0$ | $\mathrm{xmin}=-2$ | ncurves=0 |
| :--- | :--- | :--- |
| tmax=10 | $\mathrm{xmax}=2$ | diftol=.001 |
| tstep=.1 | $\mathrm{xscl}=1$ | fldres=14 |
| tplot=0 | ymin=-2 | dtime=0 |
|  | ymax=2 |  |
|  | yscl=1 |  |

8. Display the Graph screen ( $\square$ [GRAPH $]$ ).
$x$ axis $=y 1=y$


If you select ZoomSqr ([25 5), you can see that the phase-plane orbit is actually a circle. However, ZoomSqr will change your Window variables.

To examine this harmonic oscillator in more detail, use a split screen to graph the manner in which $y$ and $y^{\prime}$ change with respect to time ( $t$ ).
9. Press $M O D E$ and change the mode settings on Page 2 as shown. Then close the MODE dialog box, which redraws the graph.
Note: To display different graphs in both parts of a split screen, you must use the 2-graph mode.
10. Press [2nd [ $\boxplus$ ] to switch to the right side of the split screen.
11. Use F4 to select $\mathbf{y} 1$ ' and $\mathbf{y} 2$ '.

The right side uses the same equations as the left side. However, no equations
 are selected initially in the right side.
12. Press:

F1 9

- or - - 1 Set Fields = FLDOFF.

Important: Because FieIds=DIRFLD cannot

plot a time axis, you must change the Fields setting. FLDOFF turns off all fields.

13. In the $Y=$ Editor, press:
[ 2nd [F7]and make sure Axes = TIME.

14. In the Window Editor, change ymin and ymax as shown to the right.
ymin=-2.
ymax=2.

Note: When you enter 2-graph mode, Window variables for the right side are set to their defaults.
15. Press [GRAPH] to display the Graph screen for graph \#2.

The left side shows the phase-plane orbit. The right side shows the solution curve and its derivative.
16. To return to a full screen of the original graph, press 2nd [ $\ddagger$ ] to switch to the left side. Then press MODE and change the Split Screen setting.

Split Screen = FULL


## Example of a 3rd-Order Equation

For the 3rd-order differential equation $y^{\prime "+}+2 y^{\prime \prime}+2 y^{\prime}+\mathbf{y}=\boldsymbol{\operatorname { s i n }}(\mathbf{x})$, write a system of equations to enter in the $Y=$ Editor. Then graph the solution as a function of time. Use initial conditions $y(0)=0, y^{\prime}(0)=1$, and $y^{\prime \prime}(0)=1$.

## Example

1. Press MODE and set Graph=DIFF EQUATIONS.
2. Define a system of equations for the 3rdorder equation.
Rewrite the equation and make the necessary substitutions.
3. In the $Y=$ Editor $(\square[Y=])$, enter the system of equations.
4. Enter the initial conditions: $y i 1=0, y i 2=1$, and $y i 3=1$

Note: $\mathbf{t 0}$ is the time at which the initial conditions occur. By default, $\mathbf{t 0}=\mathbf{0}$.
5. Be sure that only $\mathbf{y} 1$ ' is selected. Use F5 to deselect any other equations.
$y^{\prime \prime \prime}+2 y^{\prime \prime}+2 y^{\prime}+y=$ $\sin (x)$
$y^{\prime \prime \prime}=\sin (x)-2 y^{\prime \prime}-2 y^{\prime}$
-y
$y^{\prime \prime \prime}=\sin (\mathrm{t})-2 \mathrm{y}^{\prime \prime}-2 \mathrm{y}^{\prime}$
-y
$y^{\prime \prime \prime}=\sin (\mathrm{t})-2 \mathrm{y} 3-2 \mathrm{y} 2$

- y1
y 3 ' $=\sin (\mathrm{t})-2 \mathrm{y} 3-2 \mathrm{y} 2$
- y1


6. 



Solution Method = RK, and


Fields $=$ FLDOFF .
Important: For 3rd- or higher-order equations, you must set Fields=FLDOFF. Otherwise, an Undefined variable error occurs when graphing.
7. In the $\mathrm{Y}=$ Editor, press:
[2nd [F7] Set Axes = TIME.
Note: With Axes=TIME, the solution to the selected equation is plotted against time
 ( t ).
8. In the Window Editor $(\square$ [WiNDOW]), set the Window variables.

| t0 =0 | xmin=-1 | ncurves=0 |
| :--- | :--- | :--- |
| tmax=10 | $x \max =10$ | diftol $=.001$ |
| tstep $=.1$ | $x s c l=1$. |  |
| tplot=0 | $y m i n=-3$ |  |
|  | $y m a x=3$ |  |
|  | $y s c l=1$ |  |

9. Display the Graph screen ( $\left.\square^{[G R A P H}\right]$ ).

Note: To find the solution at a particular time, use F3 to trace the graph.


## Setting Axes for Time or Custom Plots

Setting the axes can give you great flexibility in graphing differential equations. Custom axes are particularly effective for showing different kinds of relationships.

## Displaying the AXES Dialog Box

From the $\mathrm{Y}=$ Editor, press:
[20] [F7]


If Fields = SLPFLD, Axes is unavailable.
[0] 2nd [F7]

| Item | Description |
| :--- | :--- |
| Axes | TIME — Plots $t$ on the $x$ axis and $y$ (solutions to all selected <br> differential equations) on the $y$ axis. <br> CUSTOM - Lets you select the $x$ and $y$ axes. |

X Axis, Y Axis Active only when Axes = CUSTOM, these let you select what you want to plot on the $x$ and $y$ axes.

t- time
$y$ — solutions ( $\mathrm{y} 1, \mathrm{y} 2$, etc.) of all selected differential equations
$y^{\prime}$ — values of all selected differential equations ( y 1 ', $\mathrm{y} 2^{\prime}$, etc.)
$\mathrm{y} 1, \mathrm{y} 2$, etc. - the solution to the corresponding differential equation, regardless of whether that equation is selected
$y 1$ ', y 2 ', etc. - the value of the right-hand side of the corresponding differential equation, regardless of whether that equation is selected

Note: t is not valid for either Axis when Fields=DIRFLD. If you select t , an Invalid axes error occurs when graphing.

## Example of Time and Custom Axes

Using the predator-prey model from biology, determine the numbers of rabbits and foxes that maintain population equilibrium in a certain region. Graph the solution using both time and custom axes.

## Predator－Prey Model

Use the two coupled 1st－order differential equations：
$y 1^{\prime}=-y 1+0.1 y 1 * y 2$ and $y 22^{\prime}=3 y 2-y 1 * y 2$
where：
y1＝Population of foxes
yi1＝Initial population of foxes（2）
y2＝Population of rabbits
yi2 $=$ Initial population of rabbits（5）

1．Use MODE to set Graph＝DIFF EQUATIONS．

2．In the $Y=$ Editor（ $\bullet[Y=]$ ），define the differential equations and enter the initial conditions．

Note：To speed up graphing times，clear any other equations in the $\mathrm{Y}=$ Editor．With FLDOFF，all equations are evaluated even if they are not selected．

3．Press：
F1 9
－or－
国 Set Axes＝ON，Labels＝ON， Solution Method＝RK，and


Fields $=$ FLDOFF ．

4. In the $\mathrm{Y}=$ Editor, press:

## [-2nd [F7] Set Axes $=$ TIME.


5. In the Window Editor ( $\square$ [WINDOW]), set the Window variables.

| t0 $=0$ | $x \min =-1$ | ncurves=0 |
| :--- | :--- | :--- |
| tmax $=10$ | $x \max =10$ | diftol=.001 |
| tstep $=\pi / 24$ | $x s c l=5$ |  |
| tplot $=0$ | $y m i n=-10$ |  |
|  | $y m a x=40$ |  |
|  | $y s c l=5$ |  |

6. Graph the differential equations ( $\square$ [GRAPH]).
7. Press F6 to trace. Then press 3 ENTER to see the number of foxes (yc for y 1 ) and rabbits (yc for $\mathbf{y 2}$ ) at $\mathbf{t}=3$.
Note: Use $\Theta$ and $\Theta$ to move the trace cursor between the curves for $\mathbf{y 1}$ and $\mathbf{y 2}$.

8. Return to the $\mathrm{Y}=$ Editor. Press:

F1 9

- or -
-     - $\quad$ -

Set Fields = DIRFLD.


Note: In this example, DIRFLD is used for two related differential equations that do not represent a 2 nd-order equation.
9. Press:
[2nd [F7]
Confirm that the axes are set as shown.

10. In the $Y=$ Editor, clear the initial conditions for yi1 and yi2.

11. Return to the Graph screen, which displays only the direction field.

12. To graph a family of solutions, return to the $Y=$ Editor and enter the initial conditions shown below.
yi1 $=\{2,6,7\}$ and $\mathbf{y i} 2=\{5,12,18\}$
Note: Use a list to specify more than one initial condition.
13. Return to the Graph screen, which displays a curve for each pair of initial conditions.
14. Press [F3 to trace. Then press 3 ENTER to see the number of foxes (xc) and rabbits ( yc ) at $\mathrm{t}=3$.
Because $\mathbf{t 0}=\mathbf{0}$ and tmax $=10$, you can trace
 in the range $0 \leq t \leq 10$.

Note: Use $\Theta$ and $\Theta$ to move the trace cursor from one initial condition curve to another.

## Example Comparison of RK and Euler

Consider a logistic growth model $\mathrm{dP} / \mathrm{dt}=.001 * \mathbf{P} *(\mathbf{1 0 0}-\mathrm{P})$, with the initial condition $\mathrm{P}(\mathbf{0})=$ 10. Use the BIdData instruction to compare the graphing points calculated by the RK and Euler solution methods. Then plot those points along with a graph of the equation's exact solution.

## Example

1. Press MODE and set Graph=DIFF EQUATIONS.
2. Express the 1 st-order equation in terms of y 1 ' and y 1 .
```
y1'=.001y1*(100-y1)
```

Do not use implied multiplication between the variable and parentheses. If you do, it is treated as a function call.
3. Enter the equation in the $\mathrm{Y}=$ Editor ( $\bullet[Y=]$ ).
4. Enter the initial condition:
yi1=10
5. Press:
(F1) 9

- or -

囲 - $\quad 1$
Set Solution Method = RK and

|  |
| :---: |
|  |
| t0 is the time at which the initial condition occurs. By default, $\mathrm{t} 0=0$. |

Fields $=$ FLDOFF .
Note: To speed up graphing times, clear any other equations in the $\mathrm{Y}=$ Editor. With FLDOFF, all equations are evaluated even if they are not selected.
6. In the Window Editor ( $\square$ [wiNDOW]), set the Window variables.

| t0=0. | xmin=-1. | ncurves=0. |
| :--- | :--- | :--- |
| tmax=100. | xmax=100. $\quad$ diftol $=.001$ |  |

(1) Important: Change tstep from . 1 (its default) to 1. Otherwise, BldData calculates too many rows for the data variable and a Dimension error occurs.
7. In the Home screen
[ HOME [CALC HOME]
use BldData to create a data variable containing the RK graphing points.
8. Return to the $\mathrm{Y}=$ Editor, press:

F1 9
—or -

## Set Solution Method = EULER.



Note: You do not need to graph the equation before using BIdData. For more information about BIdData, refer to the Technical Reference module.
9. Return to the Home screen, and use BIdData to create a data variable

## BIdData eulerlog

 containing the Euler graphing points.10. Use the Data/Matrix Editor ([APPS) to create a new data variable named errorlog.
Note: errorlog lets you combine the data in rklog and eulerlog so that you can view the two sets of data side by side.
11. In this new data variable, define the c1, c2, and c3 column headers to refer to data in rklog and eulerlog. Also, enter column titles as shown.

To define a column header, move the cursor to that column, press F4, type the reference expression (such as rklog[1] for c1), and press ENTER.

Note: rklog[1] and rklog[2] refer to column 1 and 2 in rklog, respectively. Likewise with eulerlog[2].
12. In the Data/Matrix Editor, press F2. Then press F1 and define Plot 1 for the RK data, as shown to the right.
13. Define Plot 2 for the Euler data. Use the values shown to the right.

(1) c1=rklog[1] or
c1=eulerlog[1]
(2) c2=rklog[2]
(3) c3= eulerlog[2]


Plot Type=xyline
Mark=Cross
$\mathrm{x}=\mathrm{c} 1$
$y=c 3$
14. Return to the $Y=$ Editor, press MODE, and set Graph $=$ FUNCTION.
15. The exact solution to the differential equation is given below. Enter it as y1.
$y 1=\left(100 * e^{\wedge}(x / 10)\right) /\left(e^{\wedge}(x / 10)+9\right)$
Note: You can use deSolve( ) to find this exact, general solution. ,
16. In the Window Editor, set the Window variables.

```
xmin=-10 ymin=-10. xres=2.
xmax=100 ymax=120.
xscl=10 yscl=10.
```

17. Display the Graph screen ( $\square$ [GRAPH]).

Note: The fuzzy line on the graph indicates differences between the RK and Euler values.

18. In the Window Editor, set the Window variables to zoom in so that you can examine the differences in more detail.

| $x \min =39.7$ | $y \min =85.5$ | xres=2 |
| :--- | :--- | :--- |
| $x \max =40.3$ | $y \max =86$ |  |
| $x$ xcl=.1 | $y s c l=.1$ |  |

19. Return to the Graph screen.
20. Press $F 3$ to trace, and then press $\Theta$ or $\odot$ until y1 is selected. (1 shows in upper right corner.) Then enter 40.
RK (Plot 1)
Exact solution (y1)
y1 is selected when 2)
1 shows here

By moving the trace cursor to trace each solution to $\mathbf{x c}=\mathbf{4 0}$, you can find that:

- The exact solution (y1) is 85.8486 , rounded to six digits.
- The RK solution (Plot $\mathbf{1}$ ) is 85.8952 .
- The Euler solution (Plot $\mathbf{2}$ ) is 85.6527 .

You can also use the Data/Matrix Editor to open the errorlog data variable and scroll to time $=40$.

## Example of the deSolve( ) Function

The deSolve() function lets you solve many 1st- and 2nd-order ordinary differential equations exactly.

## Example

For a general solution, use the following syntax. For a particular solution, refer to the Technical Reference module.

## deSolve(1stOr2ndOrderODE, independentVar, dependentVar)

Using the logistic 1st-order differential equation, find the general solution for $y$ with respect to $t$.

## Notes:

- For maximum accuracy, use $1 / 1000$ instead of .001 . A floating-point number can introduce round-off errors.
- This example does not involve graphing, so you can use any Graph mode.

Before using deSolve( ), clear any existing t and y variables. Otherwise, an error occurs.

1. In the Home screen
[- HOME [CALC HOME]use deSolve( ) to find the general solution.

2. Use the solution to define a function.
a) Press $\Theta$ to highlight the solution in the history area. Then press ENTER to autopaste it into the entry line.
b) Insert the Define instruction at the beginning of the line. Then press EENTER.

Note: Press 2nd (1) to move to the beginning of the entry line.
3. For an initial condition $\mathbf{y}=\mathbf{1 0}$ with $\mathbf{t}=\mathbf{0}$, use solve( ) to find the @1 constant.

Note: If you got a different constant (@2, etc.), solve for that constant.


For @, type

4. Evaluate the general solution (y) with the constant @1=9/100 to obtain the particular solution shown.

| $\text { - }- \text { \|e } 1=\frac{9}{100}$ | $\frac{100 \cdot e^{\frac{t}{10}}}{e^{\frac{t}{10}}+9}$ |  |
| :---: | :---: | :---: |
|  |  |  |
|  |  |  |
|  |  |  |
| - |  |  |
| F-MAl\|l| | FUNAC | $4 r^{\prime} \geqslant 0$ |

You can also use deSolve() to solve this problem directly. Enter:

## deSolve(y' $=1 / 1000 \mathrm{y} *(100-\mathrm{y})$ and $\mathrm{y}(0)=10, \mathrm{t}, \mathrm{y})$ <br> Troubleshooting with the Fields Graph Format

If you have difficulties graphing a differential equation, this section can help you correct the problem. Many problems may be related to your Fields graph format setting.

## Setting the Fields Graph Format

From the $\mathrm{Y}=$ Editor, Window Editor, or Graph screen, press:
F1 9
—or -


What Order Equation Are You Graphing?

| If the equation is: | Valid Fields settings are: |
| :--- | :--- |
| 1st-order | SLPFLD or FLDOFF |


| If the equation is: Valid | Valid Fields settings are: |
| :---: | :---: |
| 2nd-order <br> (system of two 1st-order equations) | DIRFLD or FLDOFF |
| 3rd- or higher-order (system of three or more 1st-order equations) | FLDOFF |
| Because Fields = SLPFLD is the default setting, a common error message is shown the right. | to |

When you see this or any other error message:

- For your order of equation, use the previous table to find the valid Fields settings. Change to the applicable setting.
- For a particular Fields setting, check the following for information that applies to that setting.


## Fields=SLPFLD

| In the $Y=$ Editor | Use F4 to select one and only one 1st-order equation. You can enter mulNotele equations, but only one at a time can be selected. <br> The selected equation must not refer to any other equation in the $Y=$ Editor. For example: <br> If $\mathbf{y 1} \mathbf{1 ' y}^{\mathbf{y}} \mathbf{2}$, an Undefined variable error occurs when you graph. |
| :---: | :---: |
| In the Graph screen | If the slope field is drawn but no solution curve is plotted, specify an initial condition. |

## Fields=DIRFLD

| In the $Y=$ Editor | Enter a valid system of two 1st-order equations. For information about defining a valid system for a 2 nd-order equation, refer to Example of a 2nd-Order Equation. <br> Set Axes = CUSTOM: <br> 2nd [F7] If Axes $=$ TIME, an Invalid axes error occurs when you graph. |
| :---: | :---: |
|  |  |
|  | If you enter initial conditions in the $Y=$ Editor, the equations referenced by the custom axes must have the same number of initial conditions. <br> Otherwise, a Dimension error occurs when you graph. |
|  |  |
| With custom axes | Set axes that are valid for your system of equations. <br> Do not select $t$ for either axis. Otherwise, an Invalid axes error occurs when you graph. <br> The two axes must refer to different equations in your system of equations. For example, y 1 vs . y 2 is valid, but y 1 vs . y 1 ' gives an Invalid axes error. |


| In the Graph | If the direction field is drawn but no curve is plotted, enter <br> initial conditions in the $Y=$ Editor or select one interactively <br> screen <br> from the Graph screen. If you did enter initial conditions, |
| :--- | :--- |
| select ZoomFit: |  |
|  | F2 alpha $A$ |

## Fields=FLDOFF

In the If you enter a 2nd- or higher-order equation, enter it as a valid
system of equations. All equations (selected or not) must have the same number of initial conditions. Otherwise, a Dimension error occurs when you graph.


To set Axes = TIME or CUSTOM, press:
(: 2nd [F7]

| With custom |  |
| :--- | :--- |
| axes | If $X A x i s ~ i s ~ n o t ~$ <br> for |
| for each equation in the $Y=$ Editor (whether the equation is |  |
| selected or not). |  |
| Otherwise, a Diff Eq setup error occurs when you graph. |  |

## If You Use the Table Screen to View Differential Equations

You can use the Table screen to view the points for a differential equation graph. However, the table may show different equations than those graphed. The table shows only the selected equations, regardless of whether those equations will be plotted with your current Fields and Axes settings

## Tables

## Overview of Steps in Generating a Table

To generate a table of values for one or more functions, use the general steps shown below. For specific information about setting table parameters and displaying the table, refer to the following pages.

## Generating a Table

1. Set Graph mode and, if necessary, Angle mode (MODE).
Note: Tables are not available in 3D Graph mode.
2. Define functions on $Y=$ Editor $(\square[Y=])$.
3. Select (ㅌ4) which defined functions to display in the table.
Note: For information on defining and selecting functions with the $\mathrm{Y}=$ Editor,
 refer to Basic Function Graphing.
4. Set up the initial table parameters ( $\bullet$ [TBLSET]).
Note: You can specify an automatic table that is based on initial values or that
 matches a graph, or a manual (ask) table.

5. Display the table ( $\square$ [TABLE]).

| 盛 |  |  |  |
| :---: | :---: | :---: | :---: |
|  | ${ }^{41}$ |  |  |
|  | -711. |  |  |
| -8. | ${ }^{-396}$-32. |  |  |
| - | -204. |  |  |
| $\frac{\mathrm{x}=-10 .}{\text { Ela }}$ |  |  |  |
|  |  |  |  |

## Exploring the Table

From the Table screen, you can:

- Scroll through the table to see values on other pages.
- Highlight a cell to see its full value.
- Change the table's setup parameters. By changing the starting or incremental value used for the independent variable, you can zoom in or out on the table to see different levels of detail.
- Change the cell width.
- Edit selected functions.
- Build or edit a manual table to show only specified values of the independent variable.


## Setting Up the Table Parameters

To set up the initial parameters for a table, use the TABLE SETUP dialog box. After the table is displayed, you can also use this dialog box to change the parameters.

## Displaying the TABLE SETUP Dialog Box

To display the TABLE SETUP dialog box, press $\square$ [TBLSET]. From the Table screen, you can also press F2.

Setup Parameter Description

| tbIStart | If Independent $=$ AUTO and Graph $<->$ Table $=$ OFF, <br> this specifies the starting value for the independent <br> variable. |
| :--- | :--- |


| $\Delta$ tbl | If Independent $=$ AUTO and Graph $<->$ Table $=$ OFF, <br> this specifies the incremental value for the independent |
| :--- | :--- |
| variable. $\Delta$ tbl can be positive or negative, but not zero. |  |

## Graph <-> Table If Independent = AUTO:

OFF - The table is based on the values you enter for tblStart and $\Delta$ tbl.
ON - The table is based on the same independent variable values that are used to graph the functions on the Graph screen. These values depend on the Window variables set in the Window Editor and the split screen size.

Setup Parameter Description
Independent AUTO - The TI-89 Titanium automatically generates a series of values for the independent variable based on tblStart, $\Delta$ tbl, and Graph <-> Table.
ASK - Lets you build a table manually by entering specific values for the independent variable.

Note: The table initially starts at tblStart, but you can use $\Theta$ to scroll to prior values.

## Which Setup Parameters to Use

| To generate: | tblStart | $\Delta$ tbl | Graph $\langle->$ Table | Independent |
| :--- | :--- | :--- | :---: | :---: |
| An automatic table |  |  |  |  |
| -Based on initial <br> values | value | value | OFF | AUTO |
| -That matches <br> Graph screen | - | - | ON | AUTO |
| A manual table | - | - | - | ASK |

Note: "-" means that any value entered for this parameter is ignored for the indicated type of table.

In SEQUENCE graphing mode, use integers for tblStart and $\Delta$ tbl.

## Changing the Setup Parameters

From the TABLE SETUP dialog box:

1. Use $\Theta$ and $\Theta$ to highlight the value or setting to change.
2. Specify the new value or setting.

| To change: | Do this: |
| :---: | :---: |
| tbIStart or <br> $\Delta$ tbl | Type the new value. The existing value is erased when you start to type. - or - <br> Press (1) or (1) to remove the highlighting. Then edit the existing value. |
| Graph < - > Table or Independent | Press (1) or © 1 to display a menu of valid settings. Then either: <br> - Move the cursor to highlight the setting and press ENTER. <br> - or - <br> - Press the number for that setting. |

Note: To cancel a menu or exit the dialog box without saving any changes, press ESC instead of ENTER.
3. After changing all applicable values or settings, press ENTER to save your changes and close the dialog box.

## From the Home Screen or a Program

You can set up a table's parameters from the Home screen or a program. You can:

- Store values directly to the system variables tbIStart and $\Delta$ tbl. Refer to "Storing and Recalling Variable Values" in Operating the Calculator.
- Set Graph <-> Table and Independent by using the setTable function. Refer to the Technical Reference module.


## Displaying an Automatic Table

If Independent = AUTO on the TABLE SETUP dialog box, a table is generated automatically when you display the Table screen. If Graph $<->$ Table $=$ ON, the table matches the trace values from the Graph screen. If Graph $<->$ Table $=$ OFF, the table is based on the values you entered for tbIStart and $\Delta$ tbl.

## Before You Begin

Define and select the applicable functions on the $Y=$ Editor $(\square[Y=])$. This example uses $y 1(x)=x^{3}-x / 3$.

Then enter the initial table parameters ( $\bullet$ [TBLSET]).


## Displaying the Table Screen

To display the Table screen, press $\square$ [TABLE] or APPS 5.
The cursor initially highlights the cell that contains the starting value of the independent variable. You can move the cursor to any cell that contains a value.

First column shows values of $-\quad$ Other columns show corresponding the independent variable. values of the functions selected in the $Y=$ Editor.

Header row shows names of independent variable ( x ) and selected functions (y1).

Entry line shows full value of highlighted cell.

| F1V Fise | V: Held | \% | S: |
| :---: | :---: | :---: | :---: |
| $x$ | 41 |  |  |
| 1. | . 66607 |  |  |
| 1.1 | . 96433 |  |  |
| 1.2 | 1.323 |  |  |
| 1.3 | 1.7637 |  |  |
| 1.4 | 2.2773 |  |  |
|  |  |  |  |
|  |  |  |  |
| Fifl\|l | Eifl illt rall |  |  |

Note: You can scroll back from the starting value by pressing $\Theta$ or 2nd $\Theta$.
To move the cursor: Press:
One cell at a time
$\odot, \odot,(1)$, or $(1)$
One page at a time $\quad 2 n d$ and then $\Theta, \odot,(1)$ or $\odot$

The header row and the first column are fixed so that they cannot scroll off the screen.

- When you scroll down or up, the variable and function names are always visible across the top of the screen.
- When you scroll right or left, the values of the independent variable are always visible along the left side of the screen.


## Changing the Cell Width

Cell width determines the maximum number of digits and symbols (decimal point, minus sign, and "E" for scientific notation) that can be displayed in a cell. All cells in the table have the same width.

Note: By default, the cell width is 6.
To change the cell width from the Table screen:


1. Press F1 9

2. Press (1) or (1) to display a menu of valid widths (3-12).
3. Move the cursor to highlight a number and press ENTER. (For single-digit numbers, you can type the number and press ENTER.)
4. Press ENTER to close the dialog box and update the table.

## How Numbers Are Displayed in a Cell

Whenever possible, a number is shown according to the currently selected display modes (Display Digits, Exponential Format, etc.). The number may be rounded as necessary. However:

- If a number's magnitude is too large for the current cell width, the number is rounded and shown in scientific notation.
- If the cell width is too narrow even for scientific notation, "..." is shown.


## Notes:

- If a function is undefined at a particular value, undef is displayed in the cell.
- Use MODE to set the display modes.

By default, Display Digits = FLOAT 6. With this mode setting, a number is shown with up to six digits, even if the cell is wide enough to show more. Other settings similarly affect a displayed number.

| If cell width is: |  |  |  |  |
| :--- | :--- | :--- | :--- | :--- |
| Full Precision | $\mathbf{3}$ | $\mathbf{6}$ | $\mathbf{9}$ | $\mathbf{1 2}$ |
| 1.2345678901 | 1.2 | 1.2346 | 1.23457 | $1.23457^{*}$ |
| -123456.78 | $\ldots$ | -1.2 E 5 | -123457. | $-123457 .{ }^{*}$ |
| .000005 | $\ldots$ | $5 . \mathrm{E}-6$ | .000005 | .000005 |
| 1.2345678 E 19 | $\ldots$ | 1.2 E 19 | 1.2346 E 19 | $1.23457 \mathrm{E} 19^{*}$ |
| $-1.23456789012 \mathrm{E}-200$ | $\ldots$ | $\ldots$ | $-1.2 \mathrm{E}-200$ | $-1.2346 \mathrm{E}-200^{*}$ |
| *Note: Depending on display mode settings, some values are not shown in <br> full precision even when the cell is wide |  |  |  |  |

Note: To see a number in full precision, highlight the cell and look at the entry line.

## If Results are Complex Numbers

A cell shows as much as possible of a complex number (according to the current display modes) and then shows "..." at the end of the displayed portion.

When you highlight a cell containing a complex number, the entry line shows the real and imaginary parts with a maximum of four digits each (FLOAT 4).

## Editing a Selected Function

From a table, you can change a selected function without having to use the $\mathrm{Y}=$ Editor.

1. Move the cursor to any cell in the column for that function. The table's header row shows the function names ( $\mathbf{y 1}$, etc.).
2. Press F4 to move the cursor to the entry line, where the function is displayed and highlighted.

Note: You can use this feature to view a function without leaving the table.
3. Make any changes, as necessary.

- Type the new function. The old function is erased when you begin typing.
- or -
- Press CLEAR to clear the old function. Then type the new one.
- or -
- Press $(1)$ or $(1)$ to remove the highlighting. Then edit the function.

Note: To cancel any changes and return the cursor to the table, press ESC instead of ENTER.
4. Press ENTER to save the edited function and update the table. The edited function is also saved in the $Y=$ Editor.

## If You Want to Change the Setup Parameters

After generating an automatic table, you can change its setup parameters as necessary.
Press F2 or [TBLSET] to display the TABLE SETUP dialog box. Then make your changes.

## Building a Manual (Ask) Table

If Independent = ASK on the TABLE SETUP dialog box, the TI-89 Titanium lets you build a table manually by entering specific values for the independent variable.

## Displaying the Table Screen

To display the Table screen, press $\quad$ [TABLE].
If you set Independent = ASK (with [TBLSET]) before displaying a table for the first time, a blank table is displayed. The cursor highlights the first cell in the independent variable column.

Header row shows names of independent variable ( x ) and selected functions (y1).

Enter a value here.


If you first display an automatic table and then change it to Independent = ASK, the table continues to show the same values. However, you can no longer see additional values by scrolling up or down off the screen.

## Entering or Editing an Independent Variable Value

You can enter a value in column 1 (independent variable) only.

1. Move the cursor to highlight the cell you want to enter or edit.

- If you start with a blank table, you can enter a value in consecutive cells only (row 1, row 2, etc.). You cannot skip cells (row 1, row 3).
- If a cell in column 1 contains a value, you can edit that value.

2. Press F3 to move the cursor to the entry line.
3. Type a new value or expression, or edit the existing value.
4. Press ENTER to move the value to the table and update the corresponding function values.

Note: To enter a new value in a cell, you do not need to press F3. Simply begin typing.

The cursor returns to the entered cell. You can use $\odot$ to move to the next row.

Enter values in any numerical order.

Enter a new value here.
Shows full value of highlighted cell.


Note: In this example, you can move the cursor to column 2, but you can enter values in column 1 only.

## Entering a List in the Independent Variable Column

1. Move the cursor to highlight any cell in the independent variable column.
2. Press F4 to move the cursor to the entry line.
3. Type a series of values, enclosed in braces $\}$ and separated by commas. For example:
$\mathrm{x}=\{1,1.5,1.75,2\}$
You can also enter a list variable or an expression that evaluates to a list.
Note: If the independent variable column contains existing values, they are shown as a list (which you can edit).
4. Press ENTER to move the values into the independent variable column. The table is updated to show the corresponding function values.

## Adding, Deleting, or Clearing

| To: | Do this: |
| :--- | :--- |
| Insert a new row  <br> above a specified row Highlight a cell in the specified row and <br> press: <br>  2nd [F6] <br>  The new row is undefined (undef) until you <br> enter a value for the independent variable. <br> Delete a row Highlight a cell in the row and press [F5. <br> If you highlight a cell in the independent <br> variable column, you can also press $\square . ~$ <br> Clear the entire table <br> (but not the selected <br> Y= functions) Press F1 8. When prompted for <br> confirmation, press ENTER. |  |

## Cell Width and Display Formats

Several factors affect how numbers are displayed in a table.

## From the Home Screen or a Program

System variable tbIInput contains a list of all independent variable values entered in the table, even those not currently displayed. tbIInput is also used for an automatic table, but it contains only the independent variable values that are currently displayed.

Before displaying a table, you can store a list of values directly to the tblInput system variable.

Tables

## Additional Graphing Topics

## Collecting Data Points from a Graph

From the Graph screen, you can store sets of coordinate values and/or math results for later analysis. You can store the information as a single-row matrix (vector) on the Home screen or as data points in a system data variable that can be opened in the Data/Matrix Editor.

## Collecting the Points

1. Display the graph. (This example shows $\mathbf{y} \mathbf{1}(\mathrm{x})=5 \boldsymbol{\operatorname { c o s }}(\mathrm{x})$.)
2. Display the coordinates or math results you want to collect.
3. Save the information to the Home screen or the sysData variable. $-(-1)$ (Home screen) or $\square \square$ (sysData variable)
4. Repeat the process as necessary.

Note: To display coordinates or math results, trace a function with F3 or perform an F5 Math operation (such as Minimum or Maximum). You can also use the freemoving cursor.


Displayed coordinates are added to the Home screen's history area (but not the entry line) as a single-row matrix or vector.


Displayed coordinates are stored in a data variable named sysData, which you can open in the Data/Matrix Editor.

Note: Use a split screen to show a graph and the Home screen or Data/Matrix Editor at the same time.

## Notes about SysData Variable

- When you press:
$\square$
- If sysData does not exist, it is created in the MAIN folder.
- If sysData already exists, new data is appended to the end of any existing data. Existing titles or column headers (for the affected columns) are cleared; titles are replaced with the applicable titles for the new data.
- The sysData variable can be cleared, deleted, etc., just as any other data variable. However, it cannot be locked.
- If the Graph screen contains a function or stat plot that references the current contents of sysData, this command will not operate.


## Graphing a Function Defined on the Home Screen

In many cases, you may create a function or expression on the Home screen and then decide to graph it. You can copy an expression to the $\mathrm{Y}=$ Editor, or graph it directly from the Home screen without using the $\mathrm{Y}=$ Editor.

## What Is the "Native" Independent Variable?

On the $Y=$ Editor, all functions must be defined in terms of the current graph mode's "native" independent variable.

| Graph Mode | Native Independent Variable |
| :--- | :---: |
| Function | x |
| Parametric | t |
| Polar | $\theta$ |
| Sequence | n |
| 3D | $\mathrm{x}, \mathrm{y}$ |
| Differential Equation | t |

## Copying from the Home Screen to the $\mathbf{Y}=$ Editor

If you have an expression on the Home screen, you can use any of the following methods to copy it to the $\mathrm{Y}=$ Editor.

| Method | Description |
| :---: | :---: |
| Copy and paste | 1. Highlight the expression on the Home screen. Press F1 and select 5:Copy. <br> 2. Display the $Y=$ Editor, highlight the desired function, and press ENTER. <br> 3. Press F1 and select 6:Paste. Then press ENTER. <br> Note: Instead of using F1 5 or F1 6 to copy and paste, use: $\bullet$ [COPY] or $\bullet$ [PASTE] |
| STOD | Store the expression to a $\mathrm{Y}=$ function name. |
|  | $2 x^{\wedge} 3+3 x^{\wedge} 2-4 x+12 \rightarrow y 1(x)$ |
|  | $\qquad$ Use the complete function name: $\mathrm{y} 1(\mathrm{x})$, not just y 1 . |
|  | Note: To copy an expression from the Home screen's history area to the entry line, use the auto-paste feature or copy and paste. |
| Define command | Define the expression as a user-defined $Y=$ function. |
|  | Define $\mathrm{y} 1(\mathrm{x})=2 \mathrm{x}^{\wedge} 3+3 \mathrm{x}^{\wedge} 2-4 \mathrm{x}+12$ |
|  | Note: Define is available from the Home screen's F4 toolbar menu. |


| Method | Description |
| :---: | :---: |
| 2nd [RCL] | If the expression is already stored to a variable: <br> 1. Display the $Y=$ Editor, highlight the desired function, and press ENTER. |
|  | 2. Press [2nd [RCL]. Type the variable name that contains the expression, and press ENTER twice. <br> Important: To recall a function variable such as $\mathbf{f 1}(\mathbf{x})$, type only f 1 , not the full function name. |
|  | 3. Press ENTER to save the recalled expression in the $Y=$ Editor's function list. |
|  | Note: 2nd [RCL] is useful if an expression is stored to a variable or function that does not correspond to the $Y=$ Editor, such as a1 or $\mathbf{f 1}(\mathbf{x})$. |

## Graphing Directly from the Home Screen

The Graph command lets you graph an expression from the Home screen without using the $Y=$ Editor. Unlike the $Y=$ Editor, Graph lets you specify an expression in terms of any independent variable, regardless of the current graphing mode.

| If the expression is in | Use the Graph command <br> as shown in this example: |
| :--- | :--- |
| terms of: |  |

The native independent variable

| Graph $1.25 \mathrm{x} * \cos (\mathrm{x})$ |
| :--- |
| For function graphing, x is the native <br> variable. |

If the expression is in Use the Graph command
terms of:
as shown in this example:

| Anon-native independent <br> variable | Graph 1.25a* $\cos (\mathrm{a}), \mathrm{a}$ |
| :--- | :--- |
|  | Specify the independent variable; <br> otherwise, you may get an error. |

Note: Graph uses the current Window variable settings and is available from the Home screen's F4 toolbar menu.

Graph does not work with sequence graphs or differential equations. For parametric, polar, and 3D graphs, use the following variations.

| In PARAMETRIC graphing mode: | Graph xExpr, yExpr, $t$ |
| :--- | :--- |
| In POLAR graphing mode: | Graph expr, $\theta$ |
| In 3D graphing mode: | Graph expr, $x, y$ |

Note: To create a table from the Home screen, use the Table command. It is similar to Graph. Both share the same expressions.

Graph does not copy the expression to the $Y=$ Editor. Instead, it temporarily suspends any functions selected on the $Y=$ Editor. You can trace, zoom, or show and edit Graph expressions on the Table screen, just the same as $Y=$ Editor functions.

## Clearing the Graph Screen

Each time you execute Graph, the new expression is added to the existing ones. To clear the graphs:

- Execute the CIrGraph command (available from the Home screen's F4 Other toolbar menu).
- Or -
- Display the $\mathrm{Y}=$ Editor. The next time you display the Graph screen, it will use the functions selected on the $\mathrm{Y}=$ Editor.


## Extra Benefits of User-Defined Functions

You can define a user-defined function in terms of any independent variable. For example:

$$
\text { Define f1(aa)=1.25aa cos(aa). }
$$

| Define $\mathrm{f} 1(\mathrm{aa})=1.25 a \mathrm{aa} * \cos (\mathrm{aa})$ |
| :--- | :--- |
| Graph $\mathrm{f} 1(\mathrm{x})$ |


\[\)|  Refers to the function by using the native  |
| :--- |
|  independent variable.  |

\]

and:

Define $\mathrm{f} 1(\mathrm{aa})=1.25 \mathrm{aa} * \cos (\mathrm{aa})$ $\mathrm{f} 1(\mathrm{x}) \rightarrow \mathrm{y} 1(\mathrm{x})$

## Graphing a Piecewise Defined Function

To graph a piecewise function, you must first define the function by specifying boundaries and expressions for each piece. The when function is extremely useful for
two-piece functions. For three or more pieces, it may be easier to create a multistatement, user-defined function.

## Using the When Function

To define a two-piece function, use the syntax:
when(condition, trueExpression, falseExpression)
For example, suppose you want to graph a function with two pieces.

| When: | Use expression: |
| :--- | :--- |
| $x<0$ | $-x$ |
| $x \geq 0$ | $5 \cos (x)$ |

In the $Y=$ Editor:
The function is "pretty printed" in this form.

Enter the function in this
 form.

For three or more pieces, you can use nested when functions.

Note: To enter when, type it or use the CATALOG.

| When: | Use expression: |  |
| :--- | :--- | :--- |
| $x<-\pi$ | $4 \sin (x)$ | $2 x+6$ |

where:
$y 1(x)=$ when $\left(x<0\right.$, when $\left.(x<-\pi, 4 * \sin (x), 2 x+6), 6-x^{\wedge} 2\right)$
This nested function is in effect when $\mathrm{x}<0$.

Nested functions quickly become complex and difficult to visualize.

## Using a Multi-Statement, User-Defined Function

For three or more pieces, you may want to create a multi-statement, user-defined function.

For example, consider the previous three-piece function.

| When: | Use expression: |
| :--- | :--- |
| $x<-\pi$ | $4 \sin (x)$ |
| $x \geq-\pi$ and $x<0$ | $2 x+6$ |
| $x \geq 0$ | $6-x^{2}$ |

Note: For information about similarities and differences between functions and programs, refer to Programming.

A multi-statement, user-defined function can have many of the control and decisionmaking structures (If, Elself, Return, etc.) used in programming. When creating the structure of a function, it may be helpful to visualize it first in a block form.
(1) Func
If $x<-\pi$ Then
Return 4 *sin(x)
ElseIf $x>=-\pi$ and $x<0$ Then
Return $2 x+6$
Else
Return 6-x^2
EndIf
(1) EndFunc
(1) Func and EndFunc must begin and end the function.

When entering a multi-statement function on the $\mathrm{Y}=$ Editor or Home screen, you must enter the entire function on a single line.

| Use a colon (:) to separate each statement. |
| :--- |
| Func:If $\mathrm{x}<-\pi$ Then:Return $4 * \sin (\mathrm{x}):$... : Endlf:EndFunc |

In the $Y=$ Editor:


## From the Home Screen or a Program

From the Home screen, you can also use the Define command to create a multistatement, user-defined function.

Information is available on copying a function from the Home screen to the $\mathrm{Y}=$ Editor.
From the Program Editor, you can create a user-defined function. For example, use the Program Editor to create a function named $\mathbf{f 1}(\mathbf{x x})$. In the $Y=$ Editor, set $\mathbf{y 1}(\mathbf{x})=\mathbf{f 1}(\mathbf{x})$.

## Graphing a Family of Curves

By entering a list in an expression, you can plot a separate function for each value in the list. (You cannot graph a family of curves in SEQUENCE or 3D graphing mode.)

## Examples Using the $\mathrm{Y}=$ Editor

Enter the expression $\{\mathbf{2 , 4 , 6 \}} \boldsymbol{\operatorname { s i n }}(\mathbf{x})$ and graph the functions.
Note: Enclose list elements in braces (2nd [1] and 2nd [!]) and separate them with commas.


Graphs three functions: $2 \sin (x), 4 \sin (x), 6 \sin (x)$

Enter the expression $\{\mathbf{2 , 4 , 6 \}} \boldsymbol{\operatorname { s i n }}(\{1,2,3\} \mathbf{x})$ and graph the functions.


Graphs three functions: $2 \sin (x), 4 \sin (2 x), 6 \sin (3 x)$

Note: The commas are shown in the entry line but not in the function list.

## Example Using the Graph Command

Similarly, you can use the Graph command from the Home screen or a program.

```
graph {2,4,6}sin(x)
graph {2,4,6}sin({1,2,3}x)
```


## Simultaneous Graphs with Lists

When the graph format is set for Graph Order = SIMUL, the functions are graphed in groups according to the element number in the list.

|  | For these example functions, the TI-89 Titanium / Voyage ${ }^{\text {TM }} 200$ graphing calculator graphs three groups. |
| :---: | :---: |
|  |  |
|  |  |
|  | - $2 \sin (x), x+4, \cos (x)$ |
|  | - $4 \sin (x), 2 x+4$ |
|  | - $6 \sin (x), 3 x+4$ |

The functions within each group are graphed simultaneously, but the groups are graphed sequentially.

Note: To set graph formats from the $\mathrm{Y}=$ Editor, Window Editor, or Graph screen, press: - 1

## When Tracing a Family of Curves

Pressing $\Theta$ or $\Theta$ moves the trace cursor to the next or previous curve in the same family before moving to the next or previous selected function.

## Using the Two-Graph Mode

In two-graph mode, the calculator's graph-related features are duplicated, giving you two independent graphing calculators. The two-graph mode is only available in split screen mode. For more information about split screens, refer to Split Screens.

## Setting the Mode

Several mode settings affect the two-graph mode, but only two settings are required. Both are on Page 2 of the MODE dialog box.

1. Press MODE. Then press F2 to display Page 2.
2. Set the following required modes.

- Split Screen = TOP-BOTTOM or LEFT-RIGHT
- Number of Graphs = 2


3. Optionally, you can set the following modes.

Page 1: - Graph = Graph mode for top or left side of the split

Page 2: - Split 1 App = application for top or left side

- Split 2 App = application for bottom or right side
- Graph 2 = Graph mode for bottom or right side

4. Press ENTER to close the dialog box.

## The Two-Graph Screen

A two-graph screen is similar to a regular split screen.


## Independent Graph-Related Features

Both Graph 1 and Graph 2 have independent:

- Graph modes (FUNCTION, POLAR, etc.). Other modes such as Angle, Display Digits, etc., are shared and affect both graphs.
- Window Editor variables.
- Table setup parameters and Table screens.
- Graph formats such as Coordinates, Axes, etc.
- Graph screens.
- $Y=$ Editors. However, both graphs share common function and stat plot definitions.

Note: The $\mathrm{Y}=$ Editor is completely independent only when the two sides use different graphing modes (as described below).

Independent graph-related applications (Y= Editor, Graph screen, etc.) can be displayed on both sides of the screen at the same time.

Non-graph-related applications (Home screen, Data/Matrix Editor, etc.) are shared and can be displayed on only one side at a time.

## The $\mathbf{Y}=$ Editor in Two-Graph Mode

Even in two-graph mode, there is actually only one $Y=$ Editor, which maintains a single function list for each Graph mode setting. However, if both sides use the same graphing mode, each side can select different functions from that single list.

- When both sides use different graphing modes, each side shows a different function list.

- When both sides use the same graphing mode, each side shows the same function list.
- You can use F4 to select different functions and stat plots (indicated
 by $\checkmark$ ) for each side.
- If you set a display style for a function, that style is used by both sides. 2nd [F6]
- Suppose Graph 1 and Graph 2 are set for function graphing. Although both sides show the same function list, you can select $(\checkmark)$ different functions for graphing

Note: If you make a change on the active $Y=$ Editor (redefine a function, change a style, etc.), that change is not reflected on the inactive side until you switch to it.

## Using a Split Screen

For more complete information about split screens, refer to Split Screens.

- To switch from one graph side to the other, press 2nd [ $\boxplus$ ] (second function of APPS).
- To display different applications:
- Switch to the applicable graph side and display the application as you normally would.
- or -
- Use MODE to change Split 1 App and/or Split 2 App.
- To exit two-graph mode:
- Use MODE to set Number of Graphs = 1, or exit the split screen by setting Split Screen = FULL.
- Or -
- Press 2nd [QuIT] twice. This always exits a split screen and returns to a full-sized Home screen.

Note: You can display non-graph-related applications (such as the Home screen) on only one side at a time.

## Remember that the Two Sides Are Independent

In two-graph mode, the two sides may appear to be related when, in fact, they are not. For example:


After the two-graph mode is set up, graph-related operations refer to the active graph side. For example:
$10 \rightarrow x$ max
affects either Graph 1 or Graph 2, depending on which is active when you execute the command.

To switch the active sides, press 2nd [ $\boxplus$ ] or use the switch function, switch(1) or switch(2).

## Drawing a Function or Inverse on a Graph

For comparison purposes, you may want to draw a function over your current graph. Typically, the drawn function is some variation of the graph. You can also draw the inverse of a function. (These operations are not available for 3D graphs.)

## Drawing a Function, Parametric, or Polar Equation

Execute DrawFunc, DrawParm, or DrawPol from the Home screen or a program. You cannot draw a function or equation interactively from the Graph screen.

> DrawFunc expression
> DrawParm expression 1, expression $2[$, tmin $][$, tmax $][$, tstep $]$
> DrawPol expression $[, \theta$ min $][, \theta \max ][, \theta$ step $]$

For example:

1. Define $\mathbf{y}(\mathbf{x})=. \mathbf{x}^{3}-2 x+6$ on the $Y=$ Editor, and graph the function.

2. On the Graph screen, press: [nd [F6] and select 2:DrawFunc.

To display the Home screen and put DrawFunc in the entry line, press: 2nd [F6] 2

的需

3. On the Home screen, specify the function to draw.
4. Press ENTER to draw the function on the Graph screen.

You cannot trace, zoom, or perform a math operation on a drawn function.

DrawFunc y1 (x)-6


Note: To clear the drawn function, press
F4

- or -

2nd [F6] and select 1:ClrDraw

## Drawing the Inverse of a Function

Execute Drawlnv from the Home screen or a program. You cannot draw an inverse function interactively from the Graph screen.

Drawlnv expression
For example, use the graph of $\mathbf{y} 1(\mathbf{x})=.1 \mathbf{x}^{3} \mathbf{- 2 x + 6}$ as shown above.

1. On the Graph screen, press: [2nd [F6] and select 3:Drawinv

To display the Home screen and put Drawlnv in the entry line, press:
2nd [F6] 3
2. On the Home screen, specify the inverse function.

Drawlnv y1(x)

Press ENTER.
The inverse is plotted as $(\mathbf{y}, \mathbf{x})$ instead of ( $\mathbf{x}, \mathrm{y}$ ).


## Drawing a Line, Circle, or Text Label on a Graph

You can draw one or more objects on the Graph screen, usually for comparisons. For example, draw a horizontal line to show that two parts of a graph have the same $y$ value. (Some objects are not available for 3D graphs.)

## Clearing All Drawings

A drawn object is not part of the graph itself. It is drawn "on top of" the graph and remains on the screen until you clear it.

From the Graph screen:

- 2nd [F6]
and select 1:CIrDraw.
- or -
- Press F4 to regraph.


Note: You can also enter CIrDraw on the Home screen's entry line.
You can also do anything that causes the Smart Graph feature to redraw the graph (such as change the Window variables or deselect a function on the $Y=$ Editor).

## Drawing a Point or a Freehand Line

From the Graph screen:

1. 2nd [F7] and select 1:Pencil.
2. Move the cursor to the applicable location.

| To draw a: | Do this: |
| :--- | :--- |
| Point (pixel-sized) | Press ENTER. |
| Freehand line | Press and hold $\uparrow$, and move the cursor to draw <br> the line. |

Note: When drawing a freehand line, you can move the cursor diagonally.
After drawing the point or line, you are still in Pencil mode.

- To continue drawing, move the cursor to
 another point.
- To quit, press ESC.

Note: If you start drawing on a white pixel, the pencil draws a black point or line. If you start on a black pixel, the pencil draws a white point or line (which can act as an eraser).

## Erasing Individual Parts of a Drawing Object

From the Graph screen:

1. 2nd [F7]
and select 2:Eraser. The cursor is shown as a small box.
2. Move the cursor to the applicable location.

| To erase: | Do this: |
| :--- | :--- |
| Area under the box | Press ENTER. |
| Along a freehand line | Press and hold $\boxed{1}$, and move the cursor to <br> erase the line. |

Note: These techniques also erase parts of graphed functions.
After erasing, you are still in Eraser mode.

- To continue erasing, move the box cursor to another location.

- To quit, press ESC.


## Drawing a Line Between Two Points

From the Graph screen:

1. $2 \mathrm{nd}[\mathrm{FF}]$ and select 3:Line.
2. Move the cursor to the 1st point, and press ENTER.
3. Move to the 2nd point, and press ENTER. (As you move, a line extends from the 1st point to the cursor.)

Note: Use 2nd to move the cursor in larger increments; [2nd (1), etc.

After drawing the line, you are still in Line mode.

- To continue drawing another line, move
 the cursor to a new 1st point.
- To quit, press ESC.


## Drawing a Circle

From the Graph screen:

1. $2 n d$ [ $\mathrm{F7}]$ and select 4:Circle.
2. Move the cursor to the center of the circle, and press ENTER.
3. Move the cursor to set the radius, and press ENTER.


Note: Use 2nd to move the cursor in larger increments; 2nd (1), etc.

## Drawing a Horizontal or Vertical Line

From the Graph screen:

1. 2 nd [ $\mathrm{F7}]$
and select 5:Horizontal or 6:Vertical. A horizontal or vertical line and a flashing cursor are displayed on the screen.
If the line is initially displayed on an axis, it may be difficult to see. However, you can easily see the flashing cursor.
2. Use the cursor pad to move the line to the appropriate position. Then press ENTER.

After drawing the line, you are still in "line" mode.

- To continue, move the cursor to another location.

- To quit, press ESC.

Note: Use 2nd to move the cursor in larger increments; 2nd (1), etc.

## Drawing a Tangent Line

To draw a tangent line, use the F5 Math toolbar menu. From the Graph screen:

1. Press F5 and select A:Tangent.
2. As necessary, use $\Theta$ and $\Theta$ to select the applicable function.
3. Move the cursor to the tangent point, and press ENTER.

The tangent line is drawn, and its equation is displayed.


Note: To set the tangent point, you can also type its $\mathbf{x}$ value and press ENTER.

## Drawing a Line Based on a Point and a Slope

To draw a line through a specified point with a specified slope, execute the DrawSIp command from the Home screen or a program. Use the syntax:

DrawSIp $x, y$, slope
You can also access DrawSip from the Graph screen.

1. $2 n d[F 6]$
and select 6:DrawSIp. This switches to the Home screen and puts DrawSIp in the entry line.
2. Complete the command, and press ENTER.

DrawSIp 4,0,6.37

The calculator automatically switches to the Graph screen and draws the line.


## Typing Text Labels

From the Graph screen:

1. 2 nd [ $\mathrm{F7}]$
and select 7:Text.
2. Move the text cursor to the location where you want to begin typing.
3. Type the text label.

After typing the text, you are still in "text" mode.

- To continue, move the cursor to
 another location.
- To quit, press ENTER or ESC.

Note: The text cursor indicates the upper-left corner of the next character you type.

## From the Home Screen or a Program

Commands are available for drawing any of the objects described in this section. There are also commands (such as PxIOn, PxILine, etc.) that let you draw objects by specifying exact pixel locations on the screen.

For a list of the available drawing commands, refer to "Drawing on the Graph Screen" in Programming.

## Saving and Opening a Picture of a Graph

You can save an image of the current Graph screen in a PICTURE (or PIC) variable. Then, at a later time, you can open that variable and display the image. This saves the image only, not the graph settings used to produce it.

## Saving a Picture of the Whole Graph Screen

A picture includes any plotted functions, axes, tick marks, and drawn objects. The picture does not include lower and upper bound indicators, prompts, or cursor coordinates.

Display the Graph screen as you want to save it; and then:

1. Press F1 and select $\mathbf{2}$ :Save Copy As.
2. Specify the type (Picture), folder, and a unique variable name.
3. Press ENTER. After typing in an input box such as Variable, you must press ENTER twice.


Important: By default, Type = GDB (for graph database). You must set Type $=$ Picture .

## Saving a Portion of the Graph Screen

You can define a rectangular box that encloses only the portion of the Graph screen that you want to save．

1．［－End［F7］
and select 8：Save Picture．
A box is shown around the outer edge of the screen．

Note：You cannot save a portion of a 3D graph．

2．Set the 1st corner of the box by moving its top and left sides．Then press ENTER．

Note：Use $\Theta$ and $\Theta$ to move the top or bottom，and use（1）and（1）to move the sides．

3．Set the 2 nd corner by moving the bottom and right sides．Then press ENTER．

4．Specify the folder and a unique variable name．

5．Press ENTER．After typing in an input box such as Variable，you must press ENTER twice．

鹵
1：Fiㅌル́al 2：Er・ヨ三に 3：Line 4：Cicicic 6：Horizontal gl G：Petit
EFEME Picture
路



Note：When saving a portion of a graph，Type is automatically fixed as Picture．

## Opening a Graph Picture

When you open a graph picture, it is superimposed over the current Graph screen. To display only the picture, use the $\mathrm{Y}=$ Editor to deselect any other functions before opening the graph picture.

From the Graph screen:

1. Press $\mathbb{F 1}$ and select 1:Open.
2. Select the type (Picture), folder, and variable that contain the graph picture you want to open.

Note: If a variable name is not shown on the dialog box, there are no graph pictures in the folder.
3. Press ENTER.


Important: By default, Type = GDB (for graph database). Be sure to set Type = Picture.

A graph picture is a drawing object. You cannot trace any curve on a picture.

## For Pictures Saved from a Portion of the Graph Screen

When you press F1 and select 1:Open, the picture is superimposed starting at the upperleft corner of the Graph screen. If the picture was saved from a portion of the Graph screen, it may appear shifted from the underlying graph.

You can specify which screen pixel to use as the upper-left corner.

## Deleting a Graph Picture

Unwanted Picture variables take up calculator memory. To delete a variable, use the VAR-LINK screen (2nd [VAR-LINK]) as described in Memory and Variable Management.

## From a Program or the Home Screen

To save (store) and open (recall) a graph picture, use the StoPic, RcIPic, AndPic, XorPic, and RplcPic commands as described in the Technical Reference module.

To display a series of graph pictures as an animation, use the CyclePic command. For an example, refer to CyclePic Command.

## Animating a Series of Graph Pictures

As described earlier in this module, you can save a picture of a graph. By using the CyclePic command, you can flip through a series of graph pictures to create an animation.

## CyclePic Command

Before using CyclePic, you must have a series of graph pictures that have the same base name and are sequentially numbered starting with 1 (such as pic1, pic2, pic3, . . . ).

To cycle the pictures, use the syntax:

CyclePic picNameString, $n$ [,wait] [,cycles] [,direction]
(1) (2) © © ©
(1) base name of pictures in quotes, such as "pic"
(2) \# of pictures to cycle
(3) seconds between
(4) \# of times to repeat cycle

## Example

This example program (named cyc) generates 10 views of a 3D graph, with each view rotated $10^{\circ}$ further around the $Z$ axis. For information about each command, refer to the

Technical Reference module. For information about using the Program Editor, refer to Programming.

| Program Listing | Every Other Graph from |
| :--- | :--- |
|  | Program |

: cyc ()
: Prgm
:local I

: OSet mode and Window variables
: setMode ("graph", "3d")
: 70 $\rightarrow$ eye $\phi$
$:-10 \rightarrow x m i n$

: $10 \rightarrow \mathrm{xmax}$
:14 $\rightarrow$ xgrid
$:-10 \rightarrow$ ymin
$: 10 \rightarrow$ ymax

:14 $\rightarrow$ ygrid
$:-10 \rightarrow z m i n$
$: 10 \rightarrow$ zmax
$: 1 \rightarrow$ zscl

: ©Define the function
$:\left(x^{\wedge} 3 * y-y^{\wedge} 3 * x\right) / 390 \rightarrow_{z 1}(x, y)$
: ©Generate pics and rotate
:For i,1,10,1

: i*10 $\rightarrow$ eye $\theta$
: DispG
: StoPic \#("pic" \& string(i))
: EndFor
: ODisplay animation
: CyclePic "pic",10,.5,5,-1
: EndPrgm

Comments start with ©. Press:

## - 1

Note: Due to its complexity, this program takes several minutes to run.
After entering this program on the Program Editor, go to the Home screen and enter cyc().

## Saving and Opening a Graph Database

A graph database is the set of all elements that define a particular graph. By saving a graph database as a GDB variable, you can recreate that graph at a later time by opening its stored database variable.

## Elements in a Graph Database

A graph database consists of:

- Mode settings (MODE) for Graph, Angle, Complex Format, and Split Screen (only if you are using the two-graph mode).
- All functions in the $\mathrm{Y}=$ Editor $(\square[\mathrm{Q}=])$, including display styles and which functions are selected.
- Table parameters ( $\square$ [TBLSET]), Window variables ( $\square$ [WINDOW]), and graph formats: F1 9
- or -
- 1

A graph database does not include drawn objects or stat plots.
Note: In two-graph mode, the elements for both graphs are saved in a single database.

## Saving the Current Graph Database

From the $Y=$ Editor, Window Editor, Table screen, or Graph screen:

1. Press F1 and select 2:Save Copy As.
2. Specify the folder and a unique variable name.
3. Press ENTER. After typing in an input box such as Variable, you must press ENTER twice.


Note: If you start from the Graph screen, be sure to use Type=GDB.

## Opening a Graph Database

Caution: When you open a graph database, all information in the current database is replaced. You may want to store the current graph database before opening a stored database.

From the Y= Editor, Window Editor, Table screen, or Graph screen:

1. Press F1 and select 1:Open.
2. Select the folder and variable that contain the graph database you want to open.
3. Press ENTER.


## Deleting a Graph Database

Unused GDB variables take up calculator memory. To delete them, use the VAR-LINK screen (2nd [VAR-LINK]) described in Memory and Variable Management.

## From a Program or the Home Screen

You can save (store) and open (recall) a graph database by using the StoGDB and RcIGDB commands as described in the Technical Reference module.

## Split Screens

## Setting and Exiting the Split Screen Mode

To set up a split screen, use the MODE dialog box to specify the applicable mode settings. After you set up the split screen, it remains in effect until you change it.

## Setting the Split Screen Mode

1. Press MODE to display the MODE dialog box.
2. Because the modes related to split screens are listed on the second page of the MODE dialog box, either:

- Use $\odot$ to scroll down.
- or -
- Press F2 to display Page 2.

3. Set the Split Screen mode to either of the following settings. For the procedure used to change a mode setting, refer to Operating the Calculator.

Split Screen Settings
TOP-BOTTOM
LEFT-RIGHT

| Hine | When you set Split Screen = TOP-BOTTOM or LEFT-RIGHT, previously dimmed modes such as Split 2 App become active. |
| :---: | :---: |
|  |  |
|  |  |
| - |  |
| EStersive |  |

## Setting the Initial Applications

Before pressing ENTER to close the MODE dialog box, you can use the Split 1 App and Split 2 App modes to select the applications you want to use.

1:Home
2: $=$ Endior
S: Window Editor* 4: Grosh
5: Table
G: DatalMatrix Editor 7: Frogram Editor B+TEXE Editor

| Mode | Specifies the application in the: |
| :--- | :--- |
| Split 1 App | Top or left part of the split screen. |
| Split 2 App | Bottom or right part of the split screen. |

If you set Split 1 App and Split 2 App to the same application, the calculator exits the split screen mode and displays the application full screen.

You can open different applications after the split screen is displayed.
Note: In two-graph mode, described in Additional Graphing Topics, the same application can be in both parts of a split screen.

| Mode | Description |
| :--- | :--- |
| Number of Graphs | Lets you set up and display two |
| Note: Leave this set to 1 unless you | independent sets of graphs. |
| have read the applicable section | This is an advanced graphing feature |
| in Additional Graphing Topics. | as described in "Using the Two- |
|  | Graph Mode" in Additional Graphing |
|  | Topics. |

## Split Screens and Pixel Coordinates

The calculator has commands that use pixel coordinates to draw lines, circles, etc., on the Graph screen. The following charts show how the Split Screen and Split Screen Ratio mode settings affect the number of pixels available on the Graph screen.

## Note:

- For a list of drawing commands, refer to "Drawing on the Graph Screen" in Programming.
- Due to the border that indicates the active application, split screens have a smaller displayable area than a full screen.


## TI-89 Titanium:

|  |  | Split 1 App |  | Split 2 App |  |
| :--- | :---: | :---: | :---: | :---: | :---: |
| Split | Ratio | $\mathbf{x}$ | $\mathbf{y}$ | $\mathbf{x}$ | $\mathbf{y}$ |
| FULL | N/A | $0-158$ | $0-76$ | N/A | N/A |


| Split | Ratio | $\mathbf{x}$ | $\mathbf{y}$ | $\mathbf{x}$ | $\mathbf{y}$ |
| :--- | :---: | :---: | :---: | :---: | :---: |
| 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 ${ }^{\text {TM }} 200$ :

|  |  | Split 1 App |  | Split 2 App |  |
| :--- | :---: | :---: | :---: | :---: | :---: |
| Split | Ratio | $\mathbf{x}$ | $\mathbf{y}$ | $\mathbf{x}$ | $\mathbf{y}$ |
| 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 | $1: 1$ | $0-116$ | $0-98$ | $0-116$ | $0-98$ |
|  | $1: 2$ | $0-76$ | $0-98$ | $0-156$ | $0-98$ |
|  | $2: 1$ | $0-156$ | $0-98$ | $0-76$ | $0-98$ |

## Exiting the Split Screen Mode

Method 1: Press MODE to display the MODE dialog box. Then set Split Screen = FULL. When you press ENTER to close the dialog box, the full-sized screen shows the application specified in Split 1 App.

Method 2: Press 2nd [QuIT] twice to display a full-sized Home screen.

## When You Turn Off the Calculator

Turning the calculator off does not exit the split screen mode.

If the calculator is turned off: | When you turn the calculator on |
| :--- |
| again: |

| When you press [2nd [0FF] | The split screen is still in effect, but the <br> Home screen is always displayed in <br> place of the application that was active <br> when you pressed 2nd [0FF]. |
| :--- | :--- |

By the Automatic Power Down ${ }^{\text {TM }}$ The split screen is just as you left it. (APD ${ }^{\text {TM }}$ ) feature, or when you
press [OFF].

## Split-Screen Status Indicators on the Apps Desktop

To return to the Apps desktop, press APPS. The split-screen status appears at the top of the Apps desktop with the names of the open Apps and the portions of the screen in which each App is displayed.

Note: The Apps desktop always appears in the full-screen view.

## Split-screen indicator Names of open Apps



| Split screen indicator | Description |
| :---: | :---: |
| $\frac{1}{2}$ | Top-bottom split screen |
|  | - 1 indicates the application that will appear in the top portion of the screen. |
|  | - $\mathbf{2}$ indicates the application that will appear in the bottom portion of the screen. |
|  | The highlighted numeral indicates the active portion of the split screen. |
| 12 | Left-right split screen |
|  | - $\mathbf{1}$ indicates the application that will appear in the left portion of the screen. |
|  | - $\mathbf{2}$ indicates the application that will appear in the right portion of the screen. |
|  | The highlighted numeral indicates the active portion of the split screen. |

## Selecting the Active Application

With a split screen, only one of the two applications can be active at a time. You can easily switch between existing applications, or you can open a different application.

## The Active Application

- The active application is indicated by a thick border.
- The toolbar and status line, which are always the full width of the display, are associated with the active application.
- For applications that have an entry line (such as the Home screen and $Y=$ Editor), the entry line is the full width of the display only when that application is active.



## Switching between Applications

Press 2nd [ $\ddagger$ ] (second function of APPS) to switch from one application to the other.


## Opening a Different Application

Method 1: 1. Use [2nd [ $\boxplus$ ] to switch to the application you want to replace.
2. Use APPS or (such as $\square$ [winDOW]) to select the new application.

If you select an application that is already displayed, the calculator switches to that application.

Method 2: 3. Press MODE and then F2.
4. Change Split 1 App and/or Split 2 App.

If you set Split 1 App and Split 2 App to the same application, the calculator exits the split screen mode and displays the application full screen.

Note: In two-graph mode, described in Additional Graphing Topics, the same application can be in both parts of a split screen.

## Using 2nd QUIT to Display the Home Screen

Note: Pressing 2nd [QUIT] twice always exits the split screen mode.

| If the Home screen: | Pressing [2nd [QUIT]: |
| :--- | :--- |
| Is not already displayed | Opens the Home screen in place of the <br> active application. |
| Is displayed, but is not the active <br> application | Switches to the Home screen and <br> makes it the active application. |
| Is the active application | Exits the split screen mode and <br> displays a full-sized Home screen. |

## When Using a Top-Bottom Split

When you select a TOP-BOTTOM split, remember that the entry line and the toolbar are always associated with the active application. For example:

Entry line is for the active $\mathrm{Y}=$ Editor, not the Graph screen.

Toolbar is for the active Graph screen, not the $Y=$ Editor.


Note: Both Top-Bottom and Left-Right splits use the same methods to select an application.

## Data/Matrix Editor

## Overview of List, Data, and Matrix Variables

To use the Data/Matrix Editor effectively, you must understand list, data, and matrix variables.

## List Variable

A list is a series of items (numbers, expressions, or character strings) that may or may not be related. Each item is called an element. In the Data/Matrix Editor, a list variable:

- Is shown as a single column of elements, each in a separate cell.
- Must be continuous; blank or empty cells are not allowed within the list.

- Can have up to 999 elements.
(1) Column title and header cells are not saved as part of the list.

If you enter more than one column of elements in a list variable, it is converted automatically into a data variable.

On the Home screen (or anywhere else you can use a list), you can enter a list as a series of elements enclosed in braces \{ \} and separated by commas.

Although you must use commas to separate elements on the entry line, spaces separate the elements in the history area.

To refer to a specified element in a list, use the format shown to the right.

(1) Name of list variable
(2) Element number (or index number)

Note: After creating a list in the Data/Matrix Editor, you can use the list in any application (such as the Home screen).

## Data Variable

A data variable is essentially a collection of lists that may or may not be related. In the Data/Matrix Editor, a data variable:

- Can have up to 99 columns.
- Can have up to 999 elements in each column. Depending on the kind of data, all
 columns may not have to be the same length.
- Must have continuous columns; blank or empty cells are not allowed within a column.

Note: For stat calculations, columns must have the same length.

From the Home screen or a program, you can use the NewData command to create a data variable that consists of existing lists.
(1) Name of data variable to create
(2) Names of existing list

Although you cannot directly display a data variable on the Home screen, you can display a specified column or element.

(1) Name of data variable
(2) Column number
(3) Column number
(4) Element number in the column

For example:
(1) Displays column 1 of the variable data1.
(2) Displays element 1 in column 1 of the variable data1.


## Matrix Variable

A matrix is a rectangular array of elements. When you create a matrix in the Data/Matrix Editor, you must specify the number of rows and columns (although you can add or delete rows and columns later). In the Data/Matrix Editor, a matrix variable:

- Looks similar to a data variable, but all columns must have the same length.
- Is initially created with 0 in each cell. You can then enter the applicable value in place of 0 .
From the Home screen or a program, you can use STOD to store a matrix with either of the equivalent methods shown to the right.
(1) row 1
(2) row 2
(3) row 1

4 row 2
Although you enter the matrix as shown above, it is pretty printed in the history area in traditional matrix form.


## Starting a Data/Matrix Editor Session

Each time you start the Data/Matrix Editor, you can create a new variable, resume using the current variable (the variable that was displayed the last time you used the Data/Matrix Editor), or open an existing variable.

## Creating a New Data, Matrix, or List Variable

1. Press APPS and then select.
2. Select 3:New.

3. Specify the applicable information for the new variable.


| Item | Lets you: |
| :---: | :---: |
| Type | Select the type of variable to create. Press (1) to display a menu of available types. |
| Folder | Select the folder in which the new variable will be stored. Press (1) to display a menu of existing folders. For information about folders, refer to the Calculator Home Screen module. |


| Item | Lets you: |  |
| :---: | :---: | :---: |
| Variable | Type a new variable name. <br> If you specify a variable that already exists, an error message will be displayed when you press ENTER. When you press ESC or ENTER to acknowledge the error, the NEW dialog box is redisplayed. |  |
| Row dimension and Col dimension | If Type = Matrix, type the number of rows and columns in the matrix. |  |

Note: If you do not type a variable name, your calculator displays the Home screen.
4. Press ENTER (after typing in an input box such as Variable, press ENTER twice) to create and display an empty variable in the Data/Matrix Editor.

## Using the Current Variable

You can leave the Data/Matrix Editor and go to another application at any time. To return to the variable that was displayed when you left the Data/Matrix Editor, launch Data/Matrix Editor again and select 1:Current.

## Creating a New Variable from the Data/Matrix Editor

From the Data/Matrix Editor:

1. Press F1 and select 3:New.
2. Specify the type, folder, and variable name. For a matrix, also specify the number of rows and columns.


## Opening Another Variable

You can open another variable at any time.

1. From the Data/Matrix Editor, press F1 and select 1:Open.

- or -

From any application, launch Data/Matrix Editor again and select 2:Open.
2. Select the type, folder, and variable to open.
3. Press ENTER.


Note: Variable shows the first existing variable in alphabetic order. If there are no existing variables, nothing is displayed.

## Deleting a Variable

Because all Data/Matrix Editor variables are saved automatically, you can accumulate quite a few variables, which take up memory.

To delete a variable, use the VAR-LINK screen ([2nd [VAR-LINK]). For information about VAR-LINK, refer to Memory and Variable Management.

## Entering and Viewing Cell Values

If you create a new variable, the Data/Matrix Editor is initially blank (for a list or data variable) or filled with zeros (for a matrix). If you open an existing variable, the values in that variable are displayed. You can then enter additional values or edit the existing ones.

## The Data/Matrix Editor Screen

A blank Data/Matrix Editor screen is shown below. When the screen is displayed initially, the cursor highlights the cell at row 1 , column 1 .
(1) Variable type
(2) Column headers

3 Row numbers
(4) Row and column number of highlighted cell
(5 Column title cells, used to type a title for each
 column

When values are entered, the entry line shows the full value of the highlighted cell.
Note: Use the title cell at the very top of each column to identify the information in that column.

## Entering or Editing a Value in a Cell

You can enter any type of expression in a cell (number, variable, function, string, etc.).

1. Move the cursor to highlight the cell you want to enter or edit.
2. Press ENTER or F3 to move the cursor to the entry line.
3. Type a new value or edit the existing one.
4. Press ENTER to enter the value into the highlighted cell.

When you press ENTER, the cursor automatically moves to highlight the next cell so that you can continue entering or editing values. However, the variable type affects the direction that the cursor moves.

Note: To enter a new value, you can start typing without pressing ENTER or F3 first. However, you must use ENTER or F3 to edit an existing value.

| Variable Type | After pressing ENTER, the cursor moves: |
| :--- | :--- |
| List or data | Down to the cell in the next row. |
| Matrix | Right to the cell in the next column. From the last cell in a <br> row, the cursor automatically moves to the first cell in the <br> next row. This lets you enter values for row1, row2, etc. |


| To move the cursor: | Press: |
| :---: | :---: |
| One cell at a time | $\bigcirc, \odot,(1)$ or ${ }_{\text {© }}$ ) |
| One page at a time | 2nd and then $\odot$, $\odot$, (1), or (1) |
| Go to row 1 in the current column or to the last row that contains data for any column on the screen, respectively. If the cursor is in or past that last row, $\rightarrow \odot$ goes to row 999. | $\begin{aligned} & \square \Theta \text { or } \\ & \theta \Theta \end{aligned}$ |

Go to column 1 or to the last column that contains data, respectively. If the cursor is
 in or past that last column, (1) goes to column 99.

Note: To enter a value from the entry line, you can also use $\odot$ or $\Theta$.
When you scroll down/up, the header row remains at the top of the screen so that the column numbers are always visible. When you scroll right/left, the row numbers remain on the left side of the screen so that they are always visible.

## How Rows and Columns Are Filled Automatically

When you enter a value in a cell, the cursor moves to the next cell. However, you can move the cursor to any cell and enter a value. If you leave gaps between cells, your device handles the gaps automatically.

- In a list variable, a cell in the gap is undefined until you enter a value for the cell.


Note: If you enter more than one column of elements in a list variable, it is converted automatically into a data variable.

- In a data variable, gaps in a column are handled the same as a list. However, if you leave a gap between columns, that column is blank.

$\rightarrow$

- In a matrix variable, when you enter a value in a cell outside the current boundaries, additional rows and/or columns are added automatically to the matrix to include the new cell. Other cells in the new rows and/or columns are filled with zeros.


Note: Although you specify the size of a matrix when you create it, you can easily add additional rows and/or columns.

## Changing the Cell Width

The cell width affects how many characters are displayed in any cell. To change the cell width in the Data/Matrix Editor:

1. To display the FORMATS dialog box, press:

F1 9

- or -
- 1


Cell width is the maximum number of characters that can be displayed in a cell. All cells have the same cell width.

Note: Remember, to see a number in full precision, you can always highlight the cell and look at the entry line.
2. With the current Cell Width setting highlighted, press (1) or © (1) to display a menu of digits (3 through 12).
3. Move the cursor to highlight a number and press ENTER. (For single-digit numbers, you can type the number and press ENTER.)
4. Press ENTER to close the dialog box.

## Clearing a Column or all Columns

This procedure erases the contents of a column. It does not delete the column.

| To clear: | Do this: |
| :--- | :--- |
| A column | 1. Move the cursor to any cell in the column. <br>  <br>  <br>  <br>  <br> 2. Press: <br> 2nd [F6] and select 5:Clear Column. <br> (This item is not available for a matrix.) |
| All columns | Press [F1 and select 8:Clear Editor. When prompted for <br> confirmation, press ENTER (or ESC to cancel). |

Note: For a list or data variable, a clear column is empty. For a matrix, a clear column contains zeros.

## Defining a Column Header with an Expression

For a list variable or a column in a data variable, you can enter a function in the column header that automatically generates a list of elements. In a data variable, you can also define one column in terms of another.

## Entering a Header Definition

In the Data/Matrix Editor:

1. Move the cursor to any cell in the column and press F4.

- or -

Move the cursor to the header cell (c1, c2, etc.) and press ENTER.

## Notes:

- ENTER is not required if you want to type a new definition or replace the existing one. However, if you want to edit the existing definition, you must press ENTER.
- To view an existing definition, press F4 or move the cursor to the header cell and look at the entry line.

2. Type the new expression, which replaces any existing definition.

If you used F4 or ENTER in Step 1, the cursor moved to the entry line and highlighted the existing definition, if any. You can also:

- Press CLEAR to clear the highlighted expression. Then type the new expression. - or -
- Press (1) or © to remove the highlighting. Then edit the old expression. Note: To cancel any changes, press ESC before pressing ENTER.

You can use an expression that: For example:

| Generates a series of numbers. | $\mathrm{c} 1=\operatorname{seq}\left(\mathrm{x}^{\wedge} 2, \mathrm{x}, 1,5\right)$ <br> $\mathrm{c} 1=\{1,2,3,4,5\}$ |
| :--- | :--- |
| Refers to another column. | $\mathrm{c} 2=2 * \mathrm{c} 1$ |
|  | $\mathrm{c} 4=\mathrm{c} 1 * \mathrm{c} 2-\sin (\mathrm{c} 3)$ |

Note: The seq function is described in the Technical Reference module. If you refer to an empty column, you will get an error message unless Auto-calculate = OFF.
3. Press ENTER, $\Theta$, or $\Theta$ to save the definition and update the columns.

(1) c1 $=\operatorname{seq}(\mathrm{x}, \mathrm{x}, 1,7)$; (2) c2=2*c1; (3 You cannot directly change a locked cell ( $\mathbf{( \mathbf { I }}$ ) since it is defined by the column header.

Note: For a data variable, header definitions are saved when you leave the Data/ Matrix Editor. For a list variable, the definitions are not saved (only their resulting cell values).

## Clearing a Header Definition

1. Move the cursor to any cell in the column and press F4.

- Or -

Move the cursor to the header cell (c1, c2, etc.) and press ENTER.
2. Press CLEAR to clear the highlighted expression.
3. Press ENTER, $\Theta$, or $\Theta$.

## Using an Existing List as a Column

Suppose you have one or more existing lists, and you want to use those existing lists as columns in a data variable.

| From the: | Do this: |
| :--- | :--- |
| Data/Matrix Editor | In the applicable column, use F4 to <br> define the column header. Refer to the <br> existing list variable. For example: <br> c1=list1 |
| Home screen or a program | Use the NewData command as <br> described in the Technical Reference <br> module. For example: |
|  | NewData datavar, listl [, list 2$][$ list $] \ldots$ |

(1) Data variable. If this data variable already exists, it will be redefined based on the specified lists.
(2) Existing list variables to copy to columns in the data variable.

Note: If you have a CBL $2^{\text {TM }}$ or CBR $^{\text {TM }}$, use these techniques for your collected lists. Use 2nd [VAR-LINK] to see existing list variables.

## To Fill a Matrix with a List

You cannot use the Data/Matrix Editor to fill a matrix with a list. However, you can use the list $>$ mat command from the Home screen or a program. For information, refer to the Technical Reference module.

## The Auto-calculate Feature

For list and data variables, the Data/Matrix Editor has an Auto-calculate feature. By default, Auto-calculate $=$ ON. Therefore, if you make a change that affects a header definition (or any column referenced in a header definition), all header definitions are recalculated automatically. For example:

- If you change a header definition, the new definition is applied automatically.
- If column 2's header is defined as c2=2*c1, any change you make in column 1 is automatically reflected in column 2.

To turn Auto-calculate off and on from the Data/Matrix Editor:

1. Press:

F1 9

- or -


2. $\square$ Change Auto-Calculate to OFF or

ON.
3. Press ENTER to close the dialog box.

If Auto-calculate $=$ OFF and you make changes as described above, the header definitions are not recalculated until you set Auto-calculate $=\mathbf{O N}$.

Note: You may want to set Auto-calculate = OFF to make changes without recalculating each time, enter a definition such as c1=c2+c3 before you enter columns 2 and 3, or override any errors in a definition until you can debug the error.

## Using Shift and CumSum Functions in a Column Header

When defining a column header, you can use the shift and cumSum functions as described below. These descriptions differ slightly from the Technical Reference module. This section describes how to use the functions in the Data/Matrix Editor. The Technical Reference module gives a more general description for the Home screen or a program.

## Using the Shift Function

The shift function copies a base column and shifts it up or down by a specified number of elements. Use [F4 to define a column header with the syntax:

(1) Column used as the base for the shift.
(2) Number of elements to shift (positive shifts up; negative shifts down).

Default is -1 .
For example, for a two-element shift up and down:

(1) c2=shift(c1,2)
(2) c3=shift(c1,-2)
(3) Shifted columns have the same length as the base
(4) Last two elements of c1 shift down and out the bottom; undefined elements shift into the top.
(5) First two elements of c1 shift up and out the top; undefined elements shift into the bottom.

Note: To enter shift, type it from the keyboard or select it from the CATALOG.

## Using the CumSum Function

The cumSum function returns a cumulative sum of the elements in a base column. Use F4 to define a column header with the syntax:

## cumSum (column)

$\square$ Column used as the base for the cumulative sum.

For example:


Note: To enter cumSum, type it, select it from the CATALOG, or press 2nd [MATH] and select it from the List submenu.

## Sorting Columns

After entering information in a data, list, or matrix variable, you can easily sort a specified column in numeric or alphabetical order. You can also sort all columns as a whole, based on a "key" column.

## Sorting a Single Column

In the Data/Matrix Editor:

1. Move the cursor to any cell in the column.

2. Press:

2nd [F6] and select 3:Sort Column.

Numbers are sorted in ascending order.
Character strings are sorted in alphabetical order.


## Sorting All Columns Based on a "Key" Column

Consider a database structure in which each column along the same row contains related information (such as a student's first name, last name, and test scores). In such a case, sorting only a single column would destroy the relationship between the columns.

In the Data/Matrix Editor:

1. Move the cursor to any cell in the "key" column.
2. In this example, move the cursor to the
 second column (c2) to sort by last name.

Note: For a list variable, this is the same as sorting a single column.
3. Press:
[2nd [F6]and select 4:Sort Col, adjust all.
Note: This menu item is not available if
 any column is locked.

When using this procedure for a data variable:

- All columns must have the same length.
- None of the columns can be locked (defined by a function in the column header). When the cursor is in a locked column, $\boldsymbol{\square}$ is shown at the beginning of the entry line.


## Saving a Copy of a List, Data, or Matrix Variable

You can save a copy of a list, data, or matrix variable. You can also copy a list to a data variable, or you can select a column from a data variable and copy that column to a list.

## Valid Copy Types

| You can copy a: | To a: |
| :--- | :--- |
| List | List or data |
| Data | Data |
| Data column | List |
| Matrix | Matrix |

Note: A list is automatically converted to a data variable if you enter more than one column of information.

## Procedure

From the Data/Matrix Editor:

1. Display the variable that you want to copy.
2. Press F1 and select 2:Save Copy As.
3. In the dialog box:

- Select the Type and Folder for the copy.
- Type a variable name for the copy.

- When available, select the column to copy from.
Note: If you type the name of an existing variable, its contents will be replaced.
(1) Column is dimmed unless you copy a data column to a list. The column information is not used for other types of copies.

4. Press ENTER (after typing in an input box such as Variable, you must press ENTER twice).

## To Copy a Data Column to a List

A data variable can have mulNotele columns, but a list variable can have only one column. Therefore, when copying from a data variable to a list, you must select the column that you want to copy.

(1) List variable to copy to.
(2) Data column that will be copied to the list. By default, this shows the column that contains the cursor.

## Statistics and Data Plots

## Overview of Steps in Statistical Analysis

This section gives an overview of the steps used to perform a statistical calculation or graph a statistical plot. For detailed descriptions, refer to the following pages.

1. Set Graph mode (MODE) to FUNCTION.
2. Enter stat data in the Data/Matrix Editor.

Note: Refer to the Data/Matrix Editor module for details on entering data in the
 Data/Matrix Editor.
3. Perform stat calculations to find stat variables or fit data to a model ([55).

4. Define and select stat plots (F2) and then (F1).

Note: You can also use the $Y=$ Editor to define and select stat plots and $\mathbf{y}(\mathbf{x})$
 functions.
5. Define the viewing window ( $\square$ [winDOw $]$ ).
6. Change the graph format if necessary.
7. F1 9


Graph the selected equations
( $\downarrow$ [GRAPH]).

## Performing a Statistical Calculation

From the Data/Matrix Editor, use the F5 Calc toolbar menu to perform statistical calculations. You can analyze one-variable or two-variable statistics, or perform several types of regression analyses.

## The Calculate Dialog Box

You must have a data variable opened. The Data/Matrix Editor will not perform statistical calculations with a list or matrix variable.

From the Data/Matrix Editor:

1. Press F5 to display the Calculate dialog box.

This example shows all items as active. On your calculator, items are active only if they are valid for the current settings of Calculation Type and Freq and Categories.

Note: If an item is not valid for the current settings, it will appear dimmed. You cannot move the cursor to a dimmed item

2. Specify applicable settings for the active items.

| Item | Description |
| :--- | :--- |
| Calculation <br> Type | Select the type of calculation. |
| x | Type the column number in the Data/Matrix Editor <br> (C1, C2, etc.) used for x values, the independent <br> variable. |
| Y | Type the column number used for y values, the <br> dependent variable. This is required for all <br> Calculation Types except OneVar. |
| Store RegEQ to | If Calculation Type is a regression analysis, you can <br> select a function name (y1(x), y2(x), etc.). This lets <br> you store the regression equation so that it will be <br> displayed in the Y= Editor. |
| Use Freq and | Select NO or YES. Note that Freq, Category, and <br> Include Categories are active only when Use Freq <br> and Categories? = YES. |
| Categories? | Type the column number that contains a "weight" <br> value for each data point. If you do not enter a <br> column number, all data points are assumed to have <br> the same weight (1). |
| Category | Type the column number that contains a category <br> value for each data point. |
| Include | If you specify a Category column, you can use this <br> item to limit the calculation to specified category <br> values. For example, if you specify $\{1,4\}$, the <br> calculation uses only data points with a category <br> value of 1 or 4. |
| Categories |  |

Note: To use an existing list variable for $\mathrm{x}, \mathrm{y}$, Freq, or Category, type the list name instead of a column number. An example using Freq, Category, and Include Categories is available.
3. Press ENTER after typing in an input box, press ENTER twice).

The results are displayed on the STAT VARS screen. The format depends on the Calculation Type. For example:


Note: Any undefined data points (shown as undef) are ignored in a stat calculation.
4. To close the STAT VARS screen, press ENTER.

## Redisplaying the STAT VARS Screen

The Data/Matrix Editor's Stat toolbar menu redisplays the previous calculation results (until they are cleared from memory).
[20] [2nd [F7]

Previous results are cleared when you:

- Edit the data points or change the Calculation Type.
- Open another data variable or reopen the same data variable (if the calculation referred to a column in a data variable). Results are also cleared if you leave and then reopen the Data/Matrix Editor with a data variable.
- Change the current folder (if the calculation referred to a list variable in the previous folder).


## Statistical Calculation Types

As described in the previous section, the Calculate dialog box lets you specify the statistical calculation you want to perform. This section gives more information about the calculation types.

## Selecting the Calculation Type

From the Calculate dialog box ([F5), highlight the current setting for the Calculation Type and press (1).

You can then select from a menu of available types.


If an item is dimmed, it is not valid for the current Calculation Type.

| Calc Type | Description |
| :--- | :--- |
| OneVar | One-variable statistics - Calculates the statistical variables. |
| TwoVar | Two-variable statistics - Calculates the statistical variables. |
| CubicReg | Cubic regression - Fits the data to the third-order <br> polynomial $\mathbf{y}=\mathbf{a x} \mathbf{x}^{3}+\mathbf{b x} \mathbf{x}^{2}+\mathbf{c x}+\mathbf{d}$. You must have at least four <br> data points. |
|  | - For four points, the equation is a polynomial fit. |
|  | For five or more points, it is a polynomial regression. |


| Calc Type | Description |
| :---: | :---: |
| LinReg | Linear regression - Fits the data to the model $\mathbf{y}=\mathbf{a x}+\mathbf{b}$ (where $a$ is the slope, and $b$ is the $y$-intercept) using a leastsquares fit and x and y . |
| LnReg | Logarithmic regression - Fits the data to the model equation $\mathbf{y = a + b} \ln (\mathbf{x})$ using a least-squares fit and transformed values $\ln (\mathbf{x})$ and $\mathbf{y}$. |
| Logistic | Logistic regression - Fits the data to the model $\mathbf{y}=\mathbf{a} /\left(\mathbf{1 + b} * \mathbf{e}^{\wedge}(\mathbf{c} * \mathbf{x})\right)+\mathbf{d}$ and updates all the system statistics variables. |
| MedMed | Median-Median - Fits the data to the model $\mathbf{y}=\mathbf{a x + b}$ (where $a$ is the slope, and $b$ is the $y$-intercept) using the medianmedian line, which is part of the resistant line technique. Summary points medx1, medy1, medx2, medy2, medx3, and medy 3 are calculated and stored to variables, but they are not displayed on the STAT VARS screen. |
| PowerReg | Power regression - Fits the data to the model equation $\mathbf{y}=\mathrm{ax}^{\mathrm{b}}$ using a least-squares fit and transformed values $\ln (\mathbf{x})$ and $\operatorname{In}(y)$. |
| QuadReg | Quadratic regression - Fits the data to the second-order polynomial $\mathbf{y}=\mathbf{a x}{ }^{2} \mathbf{+ b x}+\mathbf{c}$. You must have at least three data points. <br> - For three points, the equation is a polynomial fit. <br> - For four or more points, it is a polynomial regression. |

Calc Type Description

| QuartReg | Quartic regression — Fits the data to the fourth-order <br> polynomial $y=a x^{4}+b x^{3}+c x^{2}+d x+e . ~ Y o u ~ m u s t ~ h a v e ~ a t ~ l e a s t ~$ |
| :--- | :--- |
|  | five data points. |

- For five points, the equation is a polynomial fit.
- For six or more points, it is a polynomial regression.

| SinReg | Sinusoidal regression - Calculates the sinusoidal |
| :--- | :--- |
|  | regression and updates all the system statistics variables. |
|  | The output is always in radians, regardless of the angle |
|  | mode setting. |

Note: For TwoVar and all regression calculations, the columns that you specify for $x$ and $y$ (and optionally, Freq or Category) must have the same length.

## From the Home Screen or a Program

Use the applicable command for the calculation that you want to perform. The commands have the same name as the corresponding Calculation Type. Refer to the Technical Reference module for information about each command.

Important: These commands perform a statistical calculation but do not automatically display the results. Use the ShowStat command to show the calculation results.

## Statistical Variables

Statistical calculation results are stored to variables. To access these variables, type the variable name or use the VAR-LINK screen as described in Memory and Variable

Management. All statistical variables are cleared when you edit the data or change the calculation type. Other conditions that clear the variables are listed.

## Calculated Variables

Statistical variables are stored as system variables. However, regCoef and regeq are treated as a list and a function variable, respectively.

|  | One <br> Var | Two Var | Regressions |
| :---: | :---: | :---: | :---: |
| mean of $x$ values | $\overline{\mathrm{x}}$ | $\overline{\mathrm{X}}$ |  |
| sum of $x$ values | $\Sigma \mathrm{x}$ | $\Sigma \mathrm{x}$ |  |
| sum of $x^{2}$ values | $\Sigma x^{2}$ | $\Sigma x^{2}$ |  |
| sample std. deviation of $x$ | Sx | Sx |  |
| population std. deviation of $\mathrm{x} \dagger$ | $\sigma x$ | $\sigma x$ |  |
| number of data points | nStat | nStat |  |
| mean of y values |  | $\overline{\mathrm{y}}$ |  |
| sum of $y$ values |  | $\Sigma \mathrm{y}$ |  |
| sum of $\mathrm{y}^{2}$ values |  | $\Sigma y^{2}$ |  |
| sample standard deviation of y |  | Sy |  |
| population std. deviation of y $\dagger$ |  | $\sigma y$ |  |
| sum of $x * y$ values |  | $\Sigma x y$ |  |
| minimum of $x$ values | $\min X$ | $\min X$ |  |


|  | One <br> Var | Two Var | Regressions |
| :---: | :---: | :---: | :---: |
| maximum of $x$ values | maxX | maxX |  |
| minimum of y values |  | minY |  |
| maximum of y values |  | maxY |  |
| 1st quartile | q1 |  |  |
| median | medStat |  |  |
| 3rd quartile | q3 |  |  |
| regression equation |  |  | regeq |
| regression coefficients (a, b, c, d, e) |  |  | regCoef |
| correlation coefficient $\dagger \dagger$ |  |  | corr |
| coefficient of determination $\dagger \dagger$ |  |  | $\mathrm{R}^{2}$ |
| summary points (MedMed only) † |  |  | medx1, medy1, medx2, medy2, medx3, medy3 |

$\dagger$ The indicated variables are calculated but are not shown on the STAT VARS screen.
$\dagger \dagger$ corr is defined for a linear regression only; $\mathbf{R}^{2}$ is defined for all polynomial regressions.

## Note:

- If regeq is $4 x+7$, then regCoef is $\{47\}$. To access the "a" coefficient (the 1 st element in the list), use an index such as regCoef[1].
- 1st quartile is the median of points between $\min X$ and medStat, and 3rd quartile is the median of points between medStat and maxX.


## Defining a Statistical Plot

From the Data/Matrix Editor, you can use the entered data to define several types of statistical plots. You can define up to nine plots at a time.

## Procedure

## From the Data/Matrix Editor:

1. Press F2 to display the Plot Setup screen. Initially, none of the plots are defined.
2. Move the cursor to highlight the plot number that you want to define.
3. Press F1 to define the plot.

This example shows all items as active. On your calculator, items are active only if they are valid for the current setting of Plot Type and use Freq and Categories?.
Note: If an item is not valid for the current settings, it will appear dimmed. You
 cannot move the cursor to a dimmed item.
4. Specify applicable settings for the active items.

| Item | Description |
| :--- | :--- |
| Plot Type | Select the type of plot. |
| Mark | Select the symbol used to plot the data points: Box ( $\square$ ), <br> Cross (x), Plus ( + ), Square ( $\square$ ), or Dot ( $)$ ). |
| x | Type the column number in the Data/Matrix Editor (C1, <br> C2, etc.) used for x values, the independent variable. |
| y | Type the column number used for y values, the <br> dependent variable. This is active only for Plot Type $=$ <br> Scatter or xyline. |
| Hist. Bucket Width | Specifies the width of each bar in a histogram. |
| Freq and | Select NO or YES. Note that Freq, Category, and <br> Categories? |
| Include Categories are active only when Freq and <br> Categories? = YES. (Freq is active only for Plot Type $=$ <br> Box Plot or Histogram.) |  |
| Freq | Type the column number that contains a "weight" value <br> for each data point. If you do not enter a column <br> number, all data points are assumed to have the same <br> weight (1). |
| Category | Type the column number that contains a category value <br> for each data point. |
| Include | If you specify a Category, you can use this to limit the <br> calculation to specified category values. For example, if <br> you specify $\{1,4\}$, the plot uses only data points with a <br> category value of 1 or 4. |

## Note:

- Plots defined with column numbers always use the last data variable in the Data/Matrix Editor, even if that variable was not used to create the definition.
- To use an existing list variable for $x, y$, Freq, or Category, type the list name instead of the column number.
- An example using Freq, Category, and Include Categories is available.

5. Press ENTER (after typing in an input box, press ENTER twice).

The Plot Setup screen is redisplayed.
The plot you just defined is automatically selected for graphing.
Notice the shorthand definition for the plot.


Note: Any undefined data points (shown as undef) are ignored in a stat plot.

## Selecting or Deselecting a Plot

From Plot Setup, highlight the plot and press F4 to toggle it on or off. If a stat plot is selected, it remains selected when you:

- Change the graph mode. (Stat plots are not graphed in 3D mode.)
- Execute a Graph command.
- Open a different variable in the Data/Matrix Editor.


## Copying a Plot Definition

## From Plot Setup:

1. Highlight the plot and press [F2.
2. Press ( © ) and select the plot number that you want to copy to.
3. Press ENTER.


Note: If the original plot was selected $(\checkmark)$, the copy is also selected.

## Clearing a Plot Definition

From Plot Setup, highlight the plot and press F3. To redefine an existing plot, you do not necessarily need to clear it first; you can make changes to the existing definition. To prevent a plot from graphing, you can deselect it.

## Statistical Plot Types

When you define a plot as described in the previous section, the Plot Setup screen lets you select the plot type. This section gives more information about the available plot types.

## Scatter

Data points from $x$ and $y$ are plotted as coordinate pairs. Therefore, the columns or lists that you specify for $x$ and $y$ must be the same length.

- Plotted points are shown with the symbol that you select as the Mark.
- If necessary, you can specify the same column or list for both $x$ and $y$.



## Xyline

This is a scatter plot in which data points are plotted and connected in the order in which they appear in $x$ and $y$.


You may want to sort all the columns in the Data/Matrix Editor before plotting.
[-2nd [F6] 3 or [2nd [F6] 4

## Box Plot

This plots one-variable data with respect to the minimum and maximum data points $(\min X$ and $\max X)$ in the set.

- A box is defined by its first quartile (Q1), median (Med), and third quartile (Q3).
- Whiskers extend from $\min X$ to $\mathbf{Q 1}$ and from Q3 to maxX.

- When you select multiple box plots, they are plotted one above the other in the same order as their plot numbers.
- Use NewPlot to show statistical data as a modified box plot.
- Select Mod Box Plot as the Plot Type when you define a plot in the Data/Matrix Editor.

A modified box plot excludes points outside the interval [Q1-X, Q3+X], where $X$ is defined as 1.5 (Q3-Q1). These points, called outliers, are plotted individually beyond the box plot's whiskers, using the mark that you select.

## Histogram

This plots one-variable data as a histogram. The x axis is divided into equal widths called buckets or bars. The height of each bar (its y value) indicates how many data points fall within the bar's range.

- When defining the plot, you can specify the Hist. Bucket Width (default is 1 ) to set the width of each bar.
- A data point at the edge of a bar is counted in the bar to the right.

Number of bars $=\frac{\mathrm{xmax}-\mathrm{xmin}}{\text { Hist. Bucket Width }}$


- ZoomData ( ${ }^{\text {F2 }} 9$ from the Graph screen, $\mathrm{Y}=$ Editor, or Window Editor) adjusts xmin and xmax to include all data points, but it does not adjust the $y$ axis.
- Use [WINDOW] to set ymin = 0 and ymax = the number of data points expected in the tallest bar.
- When you trace ([F3) a histogram, the screen shows information about the traced bar.



## Using the $\mathrm{Y}=$ Editor with Stat Plots

The previous sections described how to define and select stat plots from the Data/Matrix Editor. You can also define and select stat plots from the $Y=$ Editor.

## Showing the List of Stat Plots

Press $-[Y=]$ to display the $Y=$ Editor. Initially, the nine stat plots are located "off the top" of the screen, above the $\mathbf{y}(\mathbf{x})$ functions. However, the PLOTS indicator provides some information.
 above the $\mathbf{y}(\mathbf{x})$ functions.
 Plot Setup screen.

From the $Y=$ Editor, you can perform most of the same operations on a stat plot as you can on any other $\mathbf{y}(\mathbf{x})$ function.

Note: Plots defined with column numbers always use the last data variable in the Data/Matrix Editor, even if that variable was not used to create the definition.

| To: | Do this: |
| :--- | :--- |
| Edit a plot definition | Highlight the plot and press F3. You will see the <br> same definition screen that is displayed in the <br> Data/Matrix Editor. |
| Select or deselect a plot | Highlight the plot and press F4]. |
| Turn all plots and/or <br> functions off | Press F5 and select the applicable item. You <br> can also use this menu to turn all functions on. |

Note: You can not use [2nd [F6] to set a plot's display style. However, the plot definition lets you select the mark used for the plot.

## To Graph Plots and Y= Functions

As necessary, you can select and graph stat plots and $\mathbf{y}(\mathbf{x})$ functions at the same time.

## Graphing and Tracing a Defined Stat Plot

After entering the data points and defining the stat plots, you can graph the selected plots by using the same methods you used to graph a function from the $\mathrm{Y}=$ Editor (as described in Basic Function Graphing).

## Defining the Viewing Window

Stat plots are displayed on the current graph, and they use the Window variables that are defined in the Window Editor.

Use [wiNDOW] to display the Window Editor. You can either:

- Enter appropriate values.
- or -
- Select 9:ZoomData from the F2 Zoom toolbar menu. (Although you can use any zoom, ZoomData is optimized for st plots.)

ZoomData sets the viewing window to display all statistical data points.

For histograms and box plots, only xmin and xmax are adjusted. If the top of a histogram is not shown, trace the histogram to find the value for ymax.

Note: F2 Zoom is available on the $\mathrm{Y}=$ Editor, Window Editor, and Graph screen.

## Changing the Graph Format

Press:
F1 9

- or
from the $\mathrm{Y}=$ Editor, Window Editor, or Graph
screen.


Then change the settings as necessary.

## Tracing a Stat Plot

From the Graph screen, press F3 to trace a plot. The movement of the trace cursor depends on the Plot Type.

| Plot Type | Description |
| :--- | :--- |
| Scatter or xyline | Tracing begins at the first data point. |
| Box plot | Tracing begins at the median. Press © to trace to Q1 <br> and minX. Press $\oplus$ © to trace to Q3 and maxX. |
| Histogram | The cursor moves from the top center of each bar, <br> starting from the leftmost bar. |

Note: When a stat plot is displayed, the Graph screen does not automatically pan if you trace off the left or right side of the screen. However, you can still press ENTER to center the screen on the trace cursor.

When you press $\Theta$ or $\Theta$ to move to another plot or $\mathbf{y}(\mathbf{x})$ function, tracing moves to the current or beginning point on that plot (not to the nearest pixel).

## Using Frequencies and Categories

To manipulate the way in which data points are analyzed, you can use frequency values and/or category values. Frequency values let you "weight" particular data points. Category values let you analyze a subset of the data points.

## Example of a Frequency Column

In a data variable, you can use any column in the Data/Matrix Editor to specify a frequency value (or weight) for the data points on each row. A frequency value must be an integer $\geq 0$ if Calculation Type $=$ OneVar or MedMed or if Plot Type $=$ Box Plot. For other statistical calculations or plots, the frequency value can be any number $\geq 0$.

For example, suppose you enter a student's test scores, where:

- The mid-semester exam is weighted twice as much as other tests.
- The final exam is weighted three times as much.
- In the Data/Matrix Editor, you can enter the test scores and frequency values in two columns.

| Test scores | Frequency values | These weighted scores are equivalent to the single column of scores listed to the right. |  |
| :---: | :---: | :---: | :---: |
| c1 | c2 |  | c1 |
| 85 | 1 |  | 85 |
| 97 | 1 |  | 97 |
| 92 | 2 |  | 92 (1 |
| 89 | 1 |  | 92 (1) |
| 91 | 1 |  | 89 |
| 95 | 3 |  | 91 |
|  |  |  | 95 (2) |
|  |  |  | 95 (2) |
|  |  |  | 95 (2) |

(1) Frequency of 2
(2) Frequency of 3

Note: A frequency value of 0 effectively removes the data point from analysis.
To use frequency values, specify the frequency column when you perform a statistical calculation or define a stat plot. For example:


Note: You can also use frequency values from a list variable instead of a column.

## Example of a Category Column

In a data variable, you can use any column to specify a category (or subset) value for the data points on each row. A category value can be any number.

Suppose you enter the test scores from a class that has 10th and 11th grade students. You want to analyze the scores for the whole class, but you also want to analyze categories such as 10th grade girls, 10th grade boys, 10th grade girls and boys, etc.

First, determine the category values you want to use.

| Category Value | Used to indicate: |
| :---: | :---: |
| 1 | 10th grade girl |
| 2 | 10th grade boy |
| 3 | 11th grade girl |
| 4 | 11th grade boy |

Note: You do not need a category value for the whole class. Also, you do not need category values for all 10th graders or all 11th graders since they are combinations of other categories.

In the Data/Matrix Editor, you can enter the scores and the category values in two columns.

| Test scores | Category values |
| :---: | :---: |
| c 1 | c 2 |
| 85 | 1 |
| 97 | 3 |
| 92 | 2 |
| 88 | 3 |
| 90 | 2 |
| 95 | 1 |
| 79 | 4 |
| 68 | 2 |
| 92 | 4 |
| 84 | 3 |
| 82 | 1 |

To use category values, specify the category column and the category values to include in the analysis when you perform a statistical calculation or define a stat plot.

Set this to YES.
Type the column number (or list name) that contains the category values.


Within braces \{ \}, type the category values to use, separated by commas. (Do not type a column number or list name.)

Note: You can also use category values from a list variable instead of a column.

| To analyze: | Include Categories: |
| :--- | :--- |
| 10th grade girls | $\{1\}$ |
| 10th grade boys | $\{2\}$ |
| 10th grade girls and boys | $\{1,2\}$ |
| 11th grade girls | $\{3\}$ |
| 11th grade boys | $\{4\}$ |
| 11th grade girls and boys | $\{3,4\}$ |
| all girls (10th and 11th) | $\{1,3\}$ |
| all boys (10th and 11th) | $\{2,4\}$ |

Note: To analyze the whole class, leave the Category input box blank. Any category values are ignored.

## If You Have a CBL $\mathbf{2}^{\text {TM }}$ or CBR ${ }^{\text {TM }}$

The Calculator-Based Laboratory ${ }^{\text {TM }}$ System (CBL 2) and Calculator-Based Ranger ${ }^{\text {TM }}$ System (CBR) are optional accessories, available separately, that let you collect data from a variety of real-world experiments. TI-89 Titanium, CBL 2 and CBR programs are available from the TI web site at education.ti.com.

## How CBL 2 ${ }^{\text {TM }}$ Data Is Stored

When you collect data with the CBL 2, that data is initially stored in the CBL 2 unit itself. You must then retrieve the data (transfer it to the TI-89 Titanium) by using the Get command, which is described in the Technical Reference module.

Although each set of retrieved data can be stored in several variable types (list, real, matrix, pic), using list variables makes it easier to perform statistical calculations.

When you transfer the collected information to the TI-89 Titanium, you can specify the list variable names that you want to use.

For example, you can use the CBL 2 to collect temperature data over a period of time. When you transfer the data, suppose you store:

- Temperature data in a list variable called temp.
- Time data in a list variable called time.

After you store the CBL 2 information on the TI-89 Titanium, there are two ways to use the CBL 2 list variables.

Note: For specifics about using the CBL 2 and retrieving data to the TI-89 Titanium, refer to the guidebook that comes with the CBL 2 unit.

## Referring to the CBL $2^{\text {TM }}$ Lists

When you perform a statistical calculation or define a plot, you can refer explicitly to the CBL 2 list variables. For example:


## Creating a Data Variable with the CBL $\mathbf{2}^{\text {TM }}$ Lists

You can create a new data variable that consists of the necessary CBL 2 list variables.

- From the Home screen or a program, use the NewData command.

NewData dataVar, list1 [,list2 ] [,list3 ] ...
LCBL 2 list variable names. In the new data variable, list1 will be copied to column 1, list 2 to column 2, etc.
Name of the new data variable that you want to create.

For example:
NewData temp1, time, temp
creates a data variable called templ in which time is in column 1 and temp is in column 2.

- From the Data/Matrix Editor, create a new, empty data variable with the applicable name. For each CBL 2 list that you want to include, define a column header as that list name.

For example, define column 1 as time, column 2 as temp.


Note: To define or clear a column header, use [F4]. For more information, refer to the Data/Matrix Editor module.

At this point, the columns are linked to the CBL 2 lists. If the lists are changed, the columns will be updated automatically. However, if the lists are deleted, the data will be lost.

To make the data variable independent of the CBL 2 lists, clear the column header for each column. The information remains in the column, but the column is no longer linked to the CBL 2 list.

## CBR ${ }^{\text {TM }}$

You can also use the Calculator-Based Ranger ${ }^{T M}$ (CBR) to explore the mathematical and scientific relationships between distance, velocity, acceleration, and time using data collected from activities you perform.

## Programming

## Running an Existing Program

After a program is created (as described in the remaining sections of this module), you can run it from the Home screen. The program's output, if any, is displayed on the Program I/O screen, in a dialog box, or on the Graph screen.

## Running a Program

On the Home screen:

1. Type the name of the program.
2. You must always type a set of parentheses after the name.

Some programs require you to pass an argument to the program.

Note: Use [2nd [VAR-LINK] to list existing PRGM variables.
Highlight a variable and press ENTER to paste its name to the entry line.

## 3. Press ENTER.

Note: Arguments specify initial values for a program.
When you run a program, the TI-89 Titanium automatically checks for errors. For example, the following message is displayed if you:

- Do not enter ( ) after the program name.

This error message appears if you:


- Do not enter enough arguments, if required.


To cancel program execution if an error occurs, press ESC. You can then correct any problems and run the program again.

Note: The TI-89 Titanium also checks for run-time errors that are found within the program itself.

## "Breaking" a Program

When a program is running, the BUSY indicator is displayed in the status line.

Press 0 N to stop program execution. A message is then displayed.

- To display the program in the Program Editor, press ENTER. The cursor appears at the command where the break
 occurred.
- To cancel program execution, press ESC.


## Where Is the Output Displayed?

Depending on the commands in the program, the TI-89 Titanium automatically displays information on the applicable screen.

- Most output and input commands use the Program I/O screen. (Input commands prompt the user to enter information.)
- Graph-related commands typically use the Graph screen.

After the program stops, the TI-89 Titanium shows the last screen that was displayed.

## The Program I/O Screen

On the Program I/O screen, new output is displayed below any previous output (which may have been displayed earlier in the same program or a different program). After a full page of output, the previous output scrolls off the top of the screen.


Note: To clear any previous output, enter the CIrIO command in your program. You can also execute CIrIO from the Home screen.

When a program stops on the Program I/O screen, you need to recognize that it is not the Home screen (although the two screens are similar). The Program I/O screen is used only to display output or to prompt the user for input. You cannot perform calculations on this screen.

Note: If Home screen calculations don't work after you run a program, you may be on the Program I/O screen.

## Leaving the Program I/O Screen

From the Program I/O screen:

- Press F5 to toggle between the Home screen and the Program I/O screen.
- or -
- Press ESC, 2nd [QUIT], or

HOME
[CALC HOME]to display the Home screen.

- or -
- Display any other application screen (with APPS, $\Delta[Y=]$, etc.).


## Starting a Program Editor Session

Each time you start the Program Editor, you can resume the current program or function (that was displayed the last time you used the Program Editor), open an existing program or function, or start a new program or function.

## Starting a New Program or Function

1. Press APPS and then select Program Editor.
2. Select 3:New.
3. Specify the applicable information for the new program or function.

Frosrum Ed...
1:Current. 2:Operi. sprew


| Item | Lets you: |  |
| :--- | :--- | :--- |
| Type | Select whether to create a new program <br> or function. | Select the folder in which the new program or function will be <br> stored. For information about folders, refer to the Calculator <br> Home Screen module. |
| Folder | Holen |  |

## Item Lets you:

## Variable Type a variable name for the program or function. <br> If you specify a variable that already exists, an error message <br> will be displayed when you press ENTER. When you press ESC <br> or ENTER to acknowledge the error, the NEW dialog box is redisplayed.

4. Press ENTER (after typing in an input box such as Variable, you must press ENTER twice) to display an empty "template."

This is the template for a program. Functions have a similar template.


You can now use the Program Editor as described in the remaining sections of this module.

Note: A program (or function) is saved automatically as you type. You do not need to save it manually before leaving the Program Editor, starting a new program, or opening a previous one.

## Resuming the Current Program

You can leave the Program Editor and go to another application at any time. To return to the program or function that was displayed when you left the Program Editor, launch Program Editor again and select 1:Current.

## Starting a New Program from the Program Editor

To leave the current program or function and start a new one:

1. Press F1 and select 3 :New.
2. Specify the type, folder, and variable for the new program or function.
3. Press ENTER twice.


## Opening a Previous Program

You can open a previously created program or function at any time.

1. From within the Program Editor, press F1 and select 1:Open.

- or -

From another application, launch Program Editor again and select 2:Open.
2. Select the applicable type, folder, and variable.
3. Press ENTER.


Note: By default, Variable shows the first existing program or function in alphabetical order.

## Copying a Program

In some cases, you may want to copy a program or function so that you can edit the copy while retaining the original.

1. Display the program or function you want to copy.
2. Press F1 and select 2:Save Copy As.
3. Specify the folder and variable for the copy.
4. Press ENTER twice.

## Note about Deleting a Program

Because all Program Editor sessions are saved automatically, you can accumulate quite a few previous programs and functions, which take up memory storage space.

To delete programs and functions, use the VAR-LINK screen ([2nd [VAR-LINK]). For information about VAR-LINK, refer to the Memory and Variable Management module.

## Overview of Entering a Program

A program is a series of commands executed in sequential order (although some commands alter the program flow). In general, anything that can be executed from the Home screen can be included in a program. Program execution continues until it reaches the end of the program or a Stop command.

## Entering and Editing Program Lines

On a blank template, you can begin entering commands for your new program.


Note: Use the cursor pad to scroll through the program for entering or editing commands. Use $\bullet \odot$ or $\bullet \odot$ to go to the top or bottom of a program, respectively.

You enter and edit program commands in the Program Editor by using the same techniques used to enter and edit text in the Text Editor. Refer to "Entering and Editing Text" in the Text Editor module.

After typing each program line, press ENTER. This inserts a new blank line and lets you continue entering another line. A program line can be longer than one line on the screen; if so, it will wrap to the next screen line automatically.

Note: Entering a command does not execute that command. It is not executed until you run the program.

## Entering Multi- Command Lines

To enter more than one command on the same line, separate them with a colon by pressing [2nd [:].

## Entering Comments

A comment symbol (©) lets you enter a remark in a program. When you run the program, all characters to the right of $\odot$ are ignored.

```
    :prog1()
    :Prgm
(1):ODisplays sum of 1 thru n
    :Request "Enter an integer",n
(2) : expr(n)->n:©Convert to numeric expression
    :------
```

(1) Description of the program .
(2) Description of expr.

Note: Use comments to enter information that is useful to someone reading the program code.

To enter the comment symbol, press:

- $\quad \square$
- or -
- Press F2 and select 9:©


## Controlling the Flow of a Program

When you run a program, the program lines are executed in sequential order. However, some commands alter the program flow. For example:

- Control structures such as If...Endlf commands use a conditional test to decide which part of a program to execute.
- Loops commands such as For...EndFor repeat a group of commands.


## Using Indentation

For more complex programs that :If $x>5$ Then use If...Endlf and loop structures : Disp "x is > 5" such as For...EndFor, you can make : Else the programs easier to read and : Disp "x is < or = 5" understand by using indentation. :EndIf

## Displaying Calculated Results

In a program, calculated results are not displayed unless you use an output command. This is an important difference between performing a calculation on the Home screen and in a program.

These calculations will not :12*6
display a result in a program $: \cos (\pi / 4)$
(although they will on the Home : solve ( $\left.x^{\wedge} 2-x-2=0, x\right)$
screen).
Output commands such as Disp : Disp 12*6
will display a result in a program. : Disp $\cos (\pi / 4)$
:Disp solve (x^2-x-2=0,x)


Note: A list of output commands is available.

## Getting Values into a Program

To input values into a program, you can:

- Require the users to store a value (with STO』) to the necessary variables before running the program. The program can then refer to these variables.
- Enter the values directly into : Disp 12*6 the program itself.
: $\cos (\pi / 4) \rightarrow$ maximum
- Include input commands that prompt the users to enter the necessary values when they
:Input "Enter a value",i
:Request "Enter an run the program.
- Require the users to pass one or more values to the program when they run it.

Note: A list of input commands is available.

## Example of Passing Values to a Program

The following program draws a circle on the Graph screen and then draws a horizontal line across the top of the circle. Three values must be passed to the program: $x$ and $y$ coordinates for the circle's center and the radius $r$.

- When you write the program in the Program Editor:

In the ( ) beside the program name, specify the variables that will be used to store the passed values.

Notice that the program also contains commands that set up the Graph screen.
: $\operatorname{circ}(x, y, r)(1)$
.Prgm
:FnOff
:ZoomStd
:ZoomSqr
:Circle x,y,r
:LineHorz y+r
:EndPrgm
(1) Only circ( ) is initially displayed on the blank template; be sure to edit this line.

Note: In this example, you cannot use circle as the program name because it conflicts with a command name.

Before drawing the circle, the program turns off any selected $Y=$ Editor functions, displays a standard viewing window, and "squares" the window.

- To run the program from the Home screen:

The user must specify the applicable values as arguments within the ().


The arguments, in order, are passed to the program.


Note: This example assumes that the user enters values that can be displayed by the viewing window set up by ZoomStd and ZoomSqr.

## Overview of Entering a Function

A function created in the Program Editor is very similar to the functions and instructions that you typically use from the Home screen.

## Why Create a User-Defined Function?

Functions (as well as programs) are ideal for repetitive calculations or tasks. You only need to write the function once. Then you can reuse it as many times as necessary. Functions, however, have some advantages over programs.

- You can create functions that expand on the TI-89 Titanium's built-in functions. You can then use the new functions the same as any other function.
- Functions return values that can be graphed or entered in a table; programs cannot.
- You can use a function (but not a program) within an expression. For example: $3 *$ func1(3) is valid, but not $3 * \operatorname{prog1} 1(3)$.
- Because you pass arguments to a function, you can write generic functions that are not tied to specific variable names.

Note: You can create a function from the Home screen, but the Program Editor is more convenient for complex, multi-line functions.

## Differences Between Functions and Programs

This guidebook sometimes uses the word command as a generic reference to instructions and functions. When writing a function, however, you must differentiate between instructions and functions.

A user-defined function:

- Can use the following instructions only. Any others are invalid.

| Cycle | Define | Exit |
| :--- | :--- | :--- |
| For...EndFor | Goto | If...EndIf (all forms) |
| LbI | Local | Loop...EndLoop |
| Return | While...EndWhile | $\rightarrow($ STO』 key) |

- Can use all built-in TI-89 Titanium / Voyage ${ }^{\text {TM }} 200$ functions except:

| setFold | setGraph <br> switch | setMode |
| :--- | :--- | :--- |

- Can refer to any variable; however, it can store a value to a local variable only.
- The arguments used to pass values to a function are treated as local variables automatically. If you store to any other variables, you must declare them as local from within the function.
- Cannot call a program as a subroutine, but it can call another user-defined function.
- Cannot define a program.
- Cannot define a global function, but it can define a local function.

Note: Information about local variables is available.

## Entering a Function

When you create a new function in the Program Editor, the TI-89 Titanium displays a blank "template."

Note: Use the cursor pad to scroll through the function for entering or editing commands.

Function name, which you specify when you create a new function.

Enter your commands between Func and EndFunc.

All function lines begin with a colon.


Be sure to edit this line to include any necessary arguments. Remember to use argument names in the definition that will never be used when calling the function.

If the function requires input, one or more values must be passed to the function. (A user-defined function can store to local variables only, and it cannot use instructions that prompt the user for input.)

## How to Return a Value from a Function

There are two ways to return a value from a function:

- As the last line in the function
: cube (x) (before EndFunc), calculate the value to be returned.
: Func
: $x^{\wedge} 3$
: EndFunc
- Use Return. This is useful for : cube (x) exiting a function and returning a : Func value at some point other than : If $x<0$ the end of the function. : Return 0
: $x^{\wedge} 3$
:EndFunc

Note: This example calculates the cube if $\mathbf{x} \geq \mathbf{0}$; otherwise, it returns a 0 .
The argument $x$ is automatically treated as a local variable. However, if the example needed another variable, the function would need to declare it as local by using the Local command.

There is an implied Return at the end of the function. If the last line is not an expression, an error occurs.

## Example of a Function

The following function returns the xth root of a value $y(\sqrt[x]{y})$. Two values must be passed to the function: $x$ and $y$.

Note: Because $x$ and $y$ in the function are local, they are not affected by any existing $x$ or y variable.

| Function as called from the Home Screen |  | Function as defined in the Program Editor |
| :---: | :---: | :---: |
| $3 \rightarrow x: 125 \rightarrow y$ |  |  |
| $\frac{4 * \operatorname{xroot}(3,125)}{\square}$ | 20 |  |
|  |  |  |
|  | 5 |  |

## Calling One Program from Another

One program can call another program as a subroutine. The subroutine can be external (a separate program) or internal (included in the main program). Subroutines are useful when a program needs to repeat the same group of commands at several different places.

## Calling a Separate Program

To call a separate program, use the same syntax used to run the program from the Home screen.

```
:subtest1()
: Prgm
:For i,1,4,1
: subtest2(i,i*1000) : Disp x,y
:EndFor 4 :EndPrgm
:EndPrgm
```


## Calling an Internal Subroutine

To define an internal subroutine, use the Define command with Prgm...EndPrgm. Because a subroutine must be defined before it can be called, it is a good practice to define subroutines at the beginning of the main program.

An internal subroutine is called and executed in the same way as a separate program.

```
    : subtest1()
    : Prgm
(1) :local subtest2
(2) : Define subtest2 \((x, y)=\) Prgm
: : Disp x,y
(2) : EndPrgm
    : OBeginning of main program
    : For i, 1,4,1
(3): subtest2 (i,I*1000)
    : EndFor
    : EndPrgm
```

(1) Declares the subroutine as a local variable.
(2) Defines the subroutine.
(3) Calls the subroutine.

Note: Use the Program Editor's F4 Var toolbar menu to enter the Define and Prgm...EndPrgm commands.

## Notes about Using Subroutines

At the end of a subroutine, execution returns to the calling program. To exit a subroutine at any other time, use the Return command.

A subroutine cannot access local variables declared in the calling program. Likewise, the calling program cannot access local variables declared in a subroutine.

Lbl commands are local to the programs in which they are located. Therefore, a Goto command in the calling program cannot branch to a label in a subroutine or vice versa.

## Using Variables in a Program

Programs use variables in the same general way that you use them from the Home screen. However, the "scope" of the variables affects how they are stored and accessed.

## Scope of Variables

| Scope | Description |
| :--- | :--- |
| System (Global) | Variables with reserved names that are created <br> automatically to store data about the state of the |
| Variables | calculator. For example, Window variables ( $\mathbf{x m i n}, \mathbf{x m a x}$, <br> ymin, ymax, etc.) are globally available from any folder. |
|  | You can always refer to these variables by using the <br> variable name only, regardless of the current folder. |
|  | - $\quad$A program cannot create system variables, but it can <br> use the values and (in most cases) store new values. |


| Scope | Description |
| :--- | :--- |
| Folder VariablesVariables that are stored in a particular folder.  <br> - If you store to a variable name only, it is stored in the <br> current folder. For example:  <br> 5 $\rightarrow$ start  |  |
|  | - If you refer to a variable name only, that variable must |
| be in the current folder. Otherwise, it cannot be found |  |
| (even if the variable exists in a different folder). |  |

Note: If a program has local variables, a graphed function cannot access them. For example:

```
Local a
5>a
Graph a*cos(x)
```

may display an error or an unexpected result (if a is an existing variable in the current folder).

## Circular Definition Errors

When evaluating a user-defined function or running a program, you can specify an argument that includes the same variable that was used to define the function or create the program. However, to avoid Circular definition errors, you must assign a value for $x$ or $i$ variables that are used in evaluating the function or running the program.
For example:
(1) $x+1 \rightarrow x$

- or -

For i,i,10,1
(1) Disp i

EndFor
(1) Causes a Circular definition error message if x or i does not have a value. The error does not occur if $x$ or $i$ has already been assigned a value.

| Command | Description |
| :---: | :---: |
| STOD key | Stores a value to a variable. As on the Home screen, pressing STOص enters a $\rightarrow$ symbol. |
| Archive | Moves specified variables from RAM to user data archive memory. |
| BldData | Lets you create a data variable based on the graph information entered in the $Y=$ Editor, Window Editor, etc. |
| CopyVar | Copies the contents of a variable. |
| Define | Defines a program (subroutine) or function variable within a program. |
| DelFold | Deletes a folder. All variables in that folder must be deleted first. |
| DelVar | Deletes a variable. |
| getFold | Returns the name of the current folder. |
| getType | Returns a string that indicates the data type (EXPR, LIST, etc.) of a variable. |
| Local | Declares one or more variables as local variables. |
| Lock | Locks a variable so that it cannot be accidentally changed or deleted without first being unlocked. |
| MoveVar | Moves a variable from one folder to another. |
| NewData | Creates a data variable whose columns consist of a series of specified lists. |
| NewFold | Creates a new folder. |


| Command | Description |
| :--- | :--- |
| NewPic | Creates a picture variable based on a matrix. |
| Rename | Renames a variable. |
| Unarchiv | Moves specified variables from user data archive memory <br> to RAM. |
| Unlock | Unlocks a locked variable. |

Note: The Define, DelVar, and Local commands are available from the Program Editor's F4 Var toolbar menu.

## Using Local Variables in Functions or Programs

A local variable is a temporary variable that exists only while a user-defined function is being evaluated or a user-defined program is running.

## Example of a Local Variable

The following program segment shows a For...EndFor loop (which is discussed later in this module). The variable $i$ is the loop counter. In most cases, the variable $i$ is used only while the program is running.
(1) :Local I
: For i, 0,5,1
: Disp I
: EndFor
: Disp i
(1) Declares variable $i$ as local.

Note: As often as possible, use local variables for any variable that is used only within a program and does not need to be stored after the program stops.

If you declare variable $i$ as local, it is deleted automatically when the program stops so that it does not use up memory.

## What Causes an Undefined Variable Error Message?

An Undefined variable error message displays when you evaluate a user-defined function or run a user-defined program that references a local variable that is not initialized (assigned a value).

This example is a multi-statement function, rather than a program. Line breaks are shown here, but you would type the text in the entry line as one continuous line, such as: Define fact(n)=Func:Local... where the ellipsis indicates the entry line text continues offscreen.

For example:

```
    Define fact(n)=Func:
(1)Local m:
While n>1:
    n*m}->\textrm{m}:\textrm{n}-1->\textrm{n}
EndWhile:
Return m:
EndFunc
```

(1) Local variable $m$ is not assigned an initial value.

In the example above, the local variable $m$ exists independently of any variable $m$ that exists outside of the function.

## You Must Initialize Local Variables

All local variables must be assigned an initial value before they are referenced.

```
    Define fact(n)=Func:
(1) Local m: 1>m:
While n>1:
    n*m}->\textrm{m}:\textrm{n}-1->\textrm{n}
EndWhile:
Return m:
EndFunc
```

(1) 1 is stored as the initial value for $m$.

The calculator cannot use a local variable to perform symbolic calculations.

## To Perform Symbolic Calculations

If you want a function or program to perform symbolic calculations, you must use a global variable instead of a local. However, you must be certain that the global variable does not already exist outside of the program. The following methods can help.

- Refer to a global variable name, typically with two or more characters, that is not likely to exist outside of the function or program.
- Include DelVar within the function or program to delete the global variable, if it exists, before referring to it. (DelVar does not delete locked or archived variables.)


## String Operations

Strings are used to enter and display text characters. You can type a string directly, or you can store a string to a variable.

## How Strings Are Used

A string is a sequence of characters enclosed in "quotes." In programming, strings allow the program to display information or prompt the user to perform some action. For example:

```
Disp "The result is",answer
- Or -
Input "Enter the angle in degrees",ang1
- Or -
"Enter the angle in degrees">strl
Input str1,ang1
```

Some input commands (such as InputStr) automatically store user input as a string and do not require the user to enter quotation marks.

A string cannot be evaluated mathematically, even if it appears to be a numeric expression. For example, the string " 61 " represents the characters " 6 " and " 1 ", not the number 61.

Although you cannot use a string such as "61" or " $2 x+4$ " in a calculation, you can convert a string into a numeric expression by using the expr command.

## String Commands

Note: See the Technical Reference module for syntax for all commands and functions.

| Command | Description |
| :--- | :--- |
| \# | Converts a string into a variable name. This is called <br> indirection. |
| $\boldsymbol{\text { char }}$ | Appends (concatenates) two strings into one string. <br> character code. This is the opposite of the ord command. |
| $\mathbf{\text { cim }}$ | Returns the number of characters in a string. |
| expr | Converts a string into an expression and executes that <br> expression. This is the opposite of the string command. <br> Important: Some user input commands store the entered <br> value as a string. Before you can perform a mathematical <br> operation on that value, you must convert it to a numeric <br> expression. |
| format | Returns an expression as a character string based on the <br> format template (fixed, scientific, engineering, etc.) |
| inString | Searches a string to see if it contains a specified substring. If <br> so, inString returns the character position where the first <br> occurrence of the substring begins. |
| left | Returns a specified number of characters from the left side <br> (beginning) of a string. |


| Command | Description |
| :--- | :--- |
| mid | Returns a specified number of characters from any position <br> within a string. |
| ord | Returns the character code of the first character within a <br> string. This is the opposite of the char command. |
| right | Returns a specified number of characters from the right side <br> (end) of a string. |
| rotate | Rotates the characters in a string. The default is -1 (rotate <br> right one character). |
| shift | Shifts the characters in a string and replaces them with <br> spaces. The default is -1 (shift right one character and <br> replace with one space). Examples: shift("abcde",2) "cde " <br> and shift("abcde") $\Rightarrow$ " abcd" |
| string | Converts a numeric expression into a string. This is the <br> opposite of the expr command. |

## Conditional Tests

Conditional tests let programs make decisions. For example, depending on whether a test is true or false, a program can decide which of two actions to perform. Conditional tests are used with control structures such as If...Endlf and loops such as While...EndWhile (described later in this module).

## Entering a Test Operator

- Type the operator directly from the keyboard.
- or -
- Press 2nd [MATH] and select 8:Test. Then select the operator from the menu.
- or -
- Display the built-in functions. Press: CATALOG The test operators are listed near
 the bottom of the F2 Built-in menu.


## Relational Tests

Relational operators let you define a conditional test that compares two values. The values can be numbers, expressions, lists, or matrices (but they must match in type and dimension).

| Operator | True if: | Example |
| :---: | :--- | :--- |
| $>$ | Greater than | $\mathrm{a}>8$ |
| $<$ | Less than | $\mathrm{a}<0$ |
| $\geq$ | Greater than or equal to | $\mathrm{a}+\mathrm{b} \geq 100$ |
| $\leq$ | Less than or equal to | $\mathrm{a}+6 \leq \mathrm{b}+1$ |
| $=$ | Equal | list1=list2 |
| $\neq$ | Not equal to | mat1 $\neq$ mat2 |

Note: From the keyboard, you can type:
$>=$ for $\geq$
<= for $\leq$
/= for $\neq$
(To get the / character, press $\dagger$. .)

## Boolean Tests

Boolean operators let you combine the results of two separate tests.

| Operator | True if: | Example |
| :--- | :--- | :--- |
| and | Both tests are true | $a>0$ and $a \leq 10$ |
| or | At least one test is true | $a \leq 0$ or $b+c>10$ |
| xor | One test is true and the <br> other is false | $a+6<b+1$ xor $c<d$ |

## The Not Function

The not function changes the result of a test from true to false and vice versa. For example:

```
not }x>2 is true if
    is false if
    x\leq2
    x>2
```

Note: If you use not from the Home screen, it is shown as ~in the history area. For example, not $\mathbf{x > 2}$ is shown as $\sim(x>2)$.

## Using If, Lbl, and Goto to Control Program Flow

An If...Endlf structure uses a conditional test to decide whether or not to execute one or more commands. Lbl (label) and Goto commands can also be used to branch (or jump) from one place to another in a program.

## F2 Control Toolbar Menu

To enter If...Endlf structures, use the Program Editor's F2 Control toolbar menu.


The If command is available directly from the F2 menu.

To see a submenu that lists other If structures, select 2:If...Then.

When you select a structure such as
If...Then...Endlf, a template is inserted at the cursor location.

:If | Then ©
:EndIf
(1) The cursor is
positioned so that you can enter a conditional test.

## If Command

To execute only one command if a conditional test is true, use the general form:

```
    : If \(x>5\)
(1): Disp "x is greater than 5"
(2) : Disp \(x\)
```

(1) Executed only if $x>5$; otherwise, skipped.
(2) Always displays the value of $x$.

In this example, you must store a value to $x$ before executing the if command.
Note: Use indentation to make your programs easier to read and understand.

## If...Then...EndIf Structures

To execute one group of commands if a conditional test is true, use the structure:

```
    :If x>5 Then
(1): Disp "x is greater than 5"
(1): 2* }\mp@subsup{\}{X}{
(2) :EndIf
    :Disp x
```

(1) Executed only if $\mathrm{x}>5$.
(2) Displays value of:

- $2 x$ if $x>5$
- $x$ if $x \leq 5$

Note: Endlf marks the end of the Then block that is executed if the condition is true.

## If...Then...Else... Endlf Structures

To execute one group of commands if a conditional test is true and a different group if the condition is false, use this structure:

```
    :If x>5 Then
(1) : Disp "x is greater than 5"
(1): 2* 
    :Else
(2) : Disp "x is less than or
(2) equal to 5"
    : 5*x>x
    :EndIf
(3):Disp x
```

(1) Executed only if $x>5$.
(2) Executed only if $x \leq 5$.
(3) Displays value of:

- $2 x$ if $x>5$
- $5 x$ if $x \leq 5$


## If...Then...Elself... EndIf Structures

A more complex form of the If command lets you test a series of conditions. Suppose your program prompts the user for a number that corresponds to one of four options. To test for each option (If Choice=1, If Choice = 2, etc.), use the If...Then...Elself...Endlf structure.

Refer to the Technical Reference module for more information and an example.

## Lbl and Goto Commands

You can also control the flow of your program by using Lbl (label) and Goto commands.
Use the Lbl command to label (assign a name to) a particular location in the program.

## LbI labelName

L_ name to assign to this location (use the same naming convention as a variable name)

You can then use the Goto command at any point in the program to branch to the location that corresponds to the specified label.

Goto labelName
L_ specifies which LbI command to branch to

Because a Goto command is unconditional (it always branches to the specified label), it is often used with an If command so that you can specify a conditional test. For example:

```
    :If x>5
(1): Goto GT5
(2) :Disp x
    :--------
    :--------
    :Lbl GT5
    :Disp "The number was > 5"
```

(1) If $x>5$, branches directly to label GT5.
(2) For this example, the program must include commands (such as Stop) that prevent LbI GT5 from being executed if $x \leq 5$.

## Using Loops to Repeat a Group of Commands

To repeat the same group of commands successively, use a loop. Several types of loops are available. Each type gives you a different way to exit the loop, based on a conditional test.

## F2 Control Toolbar Menu

To enter most of the loop-related commands, use the Program Editor's F2 Control toolbar menu.


When you select a loop, the loop command and its corresponding End command are inserted at the cursor location.
:For | ©
:EndFor
(1) If the loop requires arguments, the cursor is positioned after the command.

You can then begin entering the commands that will be executed in the loop.
Note: A loop command marks the start of the loop. The corresponding End command marks the end of the loop.

## For...EndFor Loops

A For...EndFor loop uses a counter to control the number of times the loop is repeated. The syntax of the For command is:

Note: The ending value can be less than the beginning value, but the increment must be negative.

For(variable, begin, end [, increment])
(1)
(2) 3
4
(1) variable used as a counter
(2) counter value used the first time For is executed
(3) exits the loop when variable exceeds this value
(4) added to the counter each subsequent time For is executed

When For is executed, the variable value is compared to the end value. If variable does not exceed end, the loop is executed; otherwise, program control jumps to the command following EndFor.


Note: The For command automatically increments the counter variable so that the program can exit the loop after a certain number of repetitions.

At the end of the loop (EndFor), program control jumps back to the For command, where variable is incremented and compared to end.

For example:

```
    :For i,0,5,1
(1 : Disp I
    :EndFor
(2):Disp i
```

(1) Displays 0, 1, 2, 3, 4, and 5 .
(2) Displays 6 . When variable increments to 6 , the loop is not executed.

Note: You can declare the counter variable as local if it does not need to be saved after the program stops.

## While...EndWhile Loops

A While...EndWhile loop repeats a block of commands as long as a specified condition is true. The syntax of the While command is:

## While condition

When While is executed, the condition is evaluated. If condition is true, the loop is executed; otherwise, program control jumps to the command following EndWhile.


Note: The While command does not automatically change the condition. You must include commands that allow the program to exit the loop.

At the end of the loop (EndWhile), program control jumps back to the While command, where condition is re-evaluated.

To execute the loop the first time, the condition must initially be true.

- Any variables referenced in the condition must be set before the While command. (You can build the values into the program or prompt the user to enter the values.)
- The loop must contain commands that change the values in the condition, eventually causing it to be false. Otherwise, the condition is always true and the program cannot exit the loop (called an infinite loop).

For example:
(1) : $0 \rightarrow \mathrm{x}$
:While $x<5$
(2) : Disp $x$

3 : $x+1 \rightarrow x$
:EndWhile
(4) : Disp x
(1) Initially sets x .
(2) Displays $0,1,2,3$, and 4 .
(3) Increments $x$.
(4) Displays 5 . When x increments to 5 , the loop is not executed.

## Loop...EndLoop Loops

A Loop...EndLoop creates an infinite loop, which is repeated endlessly. The Loop command does not have any arguments.


Typically, the loop contains commands that let the program exit from the loop. Commonly used commands are: If, Exit, Goto, and Lbl (label). For example:

```
    :0->x
    :Loop
    : Disp x
    : x+1->x
(1): If }x>
    : Exit
    :EndLoop
(2):Disp x
```

(1) An If command checks the condition.
(2) Exits the loop and jumps to here when x increments to 6 .

Note: The Exit command exits from the current loop.
In this example, the If command can be anywhere in the loop.

| When the If command is: | The loop is: |
| :--- | :--- |
| At the beginning of the loop | Executed only if the condition is true. |
| At the end of the loop | Executed at least once and repeated <br> only if the condition is true. |

The If command could also use a Goto command to transfer program control to a specified Lbl (label) command.

## Repeating a Loop Immediately

The Cycle command immediately transfers program control to the next iteration of a loop (before the current iteration is complete). This command works with For...EndFor, While...EndWhile, and Loop...EndLoop.

## Lbl and Goto Loops

Although the Lbl (label) and Goto commands are not strictly loop commands, they can be used to create an infinite loop. For example:


As with Loop...EndLoop, the loop should contain commands that let the program exit from the loop.

## Configuring the TI-89 Titanium

Programs can contain commands that change the configuration of the calculator. Because mode changes are particularly useful, the Program Editor's Mode toolbar menu makes it easy to enter the correct syntax for the setMode command.

## Configuration Commands

| Command | Description |
| :--- | :--- |
| getConfg | Returns a list of calculator characteristics. |
| getFold | Returns the name of the current folder. |
| getMode | Returns the current setting for a specified mode. |
| getUnits | Returns a list of default units. |
| setFold | Sets the current folder. |
| setGraph | Sets a specified graph format (Coordinates, Graph Order, <br> etc.). |
| setMode | Sets any mode except Current Folder. |
| setTable | Sets a specified table setup parameter (tbIStart, $\Delta$ tbl, etc.) |
| setUnits | Sets default units for displayed results. |
| switch | Sets the active window in a split screen, or returns the <br> number of the active window. |

Note: The parameter/mode strings used in the setMode( ), getMode( ), setGraph( ), and setTable( ) functions do not translate into other languages when used in a program. See the Technical Reference module.

## Entering the SetMode Command

In the Program Editor:

1. Position the cursor where you want to insert the setMode command.
2. Press:
[2nd [F6] to display a list of modes.
Note: The Mode menu does not let you set the Current Folder mode. To set this
 mode, use the setFold command.
3. Select a mode to display a menu of its valid settings.
4. Select a setting.

The correct syntax is :setMode("Graph","FUNCTION") inserted into your program.

## Getting Input from the User and Displaying Output

Although values can be built into a program (or stored to variables in advance), a program can prompt the user to enter information while the program is running. Likewise, a program can display information such as the result of a calculation.

## F3 I/O Toolbar Menu

To enter most of the commonly used input/output commands, use the Program Editor's F3 I/O toolbar menu.


To see a submenu that lists additional commands, select 1:Dialog.

## Input Commands

| Command | Description |
| :--- | :--- |
| getKey | Returns the key code of the next key pressed. See the <br> Technical Reference module for a listing of key codes. |
| Input | Prompts the user to enter an expression. The expression is <br> treated according to how it is entered. For example: |
|  | A numeric expression is treated as an expression. <br> Input can also display the Graph screen and let the user <br> update the variables xc and yc (rc and $\theta$ c in polar mode) by <br> positioning the graph cursor. |
| InputStr | Prompts the user to enter an expression. The expression is <br> always treated as a string; the user does not need to enclose <br> the expression in "quotes". |
| PopUp | Displays a pop-up menu box and lets the user select an <br> item. |
| Prompt | Prompts the user to enter a series of expressions. As with <br> Input, each expression is treated according to how it is <br> entered. |

## Command Description

| Request | Displays a dialog box that prompts the user to enter an <br> expression. Request always treats the entered expression <br> as a string. |
| :--- | :--- |

Note: String input cannot be used in a calculation. To convert a string to a numeric expression, use the expr command.

## Output Commands

| Command | Description |
| :--- | :--- |
| CIrIO | Clears the Program I/O screen. |
| Disp | Displays an expression or string on the Program I/O screen. <br> Disp can also display the current contents of the Program <br> I/O screen without displaying additional information. |
| DispG | Displays the current contents of the Graph screen. |
| DispHome | Displays the current contents of the Home screen. |
| DispTbI | Displays the current contents of the Table screen. |
| Output | Displays an expression or string starting at specified <br> coordinates on the Program I/O screen. |
| Format | Formats the way in which numeric information is displayed. <br> PauseSuspends program execution until the user presses ENTER. <br> Optionally, you can display an expression during the pause. <br> A pause lets users read your output and decide when they <br> are ready to continue. |

Command Description

| Text | Displays a dialog box that contains a specified character <br> string. |
| :--- | :--- |

## Notes:

- In a program, simply performing a calculation does not display the result. You must use an output command.
- After Disp and Output, the program immediately continues. You may want to add a Pause command.


## Graphical User Interface Commands

## Command Description

Dialog... Defines a program block (consisting of Title, Request, etc., EndDlog commands) that displays a dialog box.

Toolbar... Defines a program block (consisting of Title, Item, etc., EndTbar commands) that replaces the toolbar menus. The redefined toolbar is in effect only while the program is running and only until the user selects an item. Then the original toolbar is redisplayed.

CustmOn... Activates or removes a custom toolbar. CustmOff

Custom... Defines a program block that displays a custom toolbar when EndCustm the user presses 2nd [CATALOG]. That toolbar remains in effect until the user presses 2nd [CATALOG] again or changes applications.

## Command Description

| DropDown | Displays a drop-down menu within a dialog box. |
| :--- | :--- |
| Item | Displays a menu item for a redefined toolbar. |
| Request | Creates an input box within a dialog box. |
| Text | Displays a character string within a dialog box. |
| Title | Displays the title of a dialog box or a menu title within a <br> toolbar. |

## Notes:

- When you run a program that sets up a custom toolbar, that toolbar is still available even after the program has stopped.
- Request and Text are stand-alone commands that can also be used outside of a dialog box or toolbar program block.


## Creating a Custom Menu

The custom menu feature lets you create your own toolbar menu. A custom menu can contain any available function, instruction, or set of characters. The calculator has a default custom menu that you can modify or redefine.

## Turning the Custom Menu On and Off

When you create a custom menu, you can let the user turn it on and off manually, or you can let a program turn it on and off automatically.

| To: | Do this: |
| :---: | :---: |
| Turn on the custom menu | From the Home screen or any other application: <br> - Press 2nd [CATALOG]. <br> From the Home screen or a program: <br> - Execute the CustmOn command. |
| Turn off the custom menu | From any application: <br> - Press 2nd [CATALOG] again. <br> - or - <br> - Go to a different application. <br> Using the default custom menu on the Home screen: <br> 1. Select the Tools menu: 2nd [F7] Select 3:CustmOff. <br> This pastes CustmOff in the entry line. <br> 2. Press ENTER. <br> You can also use CustmOff in a program. |

Note: When the custom menu is turned on, it replaces the normal toolbar menu. Unless a different custom menu has been created, the default custom menu is displayed.

## Defining a Custom Menu

To create a custom menu, use the following general structure.

## Custom

: Title title of F1 menu:



Title title of F2 menu
Title title of F3 menu

Note: When the user selects a menu item, the text defined by that Item command is pasted to the current cursor location.

For example:
:Custom
:Title "Vars"
:Item "L1":Item "M1":Item "Prgm1":Item "Func1":Item "Data1"
:Item "Text1":Item "Pic1":Item "GDB1":Item "Str1"
(1) :Title "f(x)"

(1):Item " $f(x+h)$ ": Item "Define $f(x)="$
:Title "Solve"
:Item "Solve(":Item " and ":Item "\{x,y\}"
:Item "Solve( and ,\{x,y\})"

```
(2) :Title "Units"
(2) :Item "_m/_s^2":Item "_ft/_s^2":Item "_m":Item "_ft":Item "_l"
(2) :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
Note: The following may be slightly different than the default custom menu on your calculator.


FIT FET FST F4T FET FGT F7T
 (1)

(2)

Note: See how "_\o\C" and "_lo\F" display as ${ }^{\circ} \mathrm{C}$ and ${ }^{\circ} \mathrm{F}$ in the menu. Similarly, see the international accented characters.

To modify the default custom menu, use 3:Restore custom default (as described below) to get the commands for the default menu. Copy those commands, use the Program

Editor to create a new program, and paste them into the blank program. Then modify the commands as necessary.

Note: This inserts all the commands on a single line. You do not need to split them into separate lines.

You can create and use only one custom menu at a time. If you need more, write a separate program for each custom menu. Then run the program for the menu you need.

## Restoring the Default Custom Menu

To restore the default:

1. From the Home screen's normal menu (not the custom menu), select Clean Up: 2nd [F6]
2. Select 3:Restore custom default.

This pastes the commands used to create the
 default menu into the entry line.
3. Press ENTER to execute the commands and restore the default.

When you restore the default, any previous custom menu is erased. If the previous menu was created with a program, you can run the program again if you want to reuse the menu later.

## Creating a Table or Graph

To create a table or a graph based on one or more functions or equations, use the commands listed in this section.

## Table Commands

| Command | Description |
| :--- | :--- |
| DispTbI | Displays the current contents of the Table screen. |
| setTable | Sets the Graph <-> Table or Independent table <br> parameters. (To set the other two table parameters, you <br> can store the applicable values to the tbIStart and $\Delta$ tbl <br> system variables.) |
| Table | Builds and displays a table based on one or more <br> expressions or functions. |

## Graphing Commands

| Command | Description |
| :--- | :--- |
| CIrGraph | Erases any functions or expressions that were graphed <br> with the Graph command. |
| Define | Creates a user-defined function. |
| DispG | Displays the current contents of the Graph screen. |
| FnOff | Deselects all (or only specified) Y= functions. |
| FnOn | Selects all (or only specified) Y= functions. |


| Command | Description |
| :--- | :--- |
| Graph | Graphs one or more specified expressions, using the <br> current graphing mode. |
| Input | Displays the Graph screen and lets the user update the <br> variables xc and yc (rc and $\theta$ c in polar mode) by positioning <br> the graph cursor. |
| NewPlot | Creates a new stat plot definition. |
| PlotsOff | Deselects all (or only specified) stat data plots. |
| PlotsOn | Selects all (or only specified) stat data plots. |
| setGraph | Changes settings for the various graph formats <br> (Coordinates, Graph Order, etc.). |
| setMode | Sets the Graph mode, as well as other modes. |
| Style | Sets the display style for a function. |
| Trace | Lets a program trace a graph. <br> ZoomBox <br> - to -Perform all of the Zoom operations that are available from <br> the F2 toolbar menu on the Y= Editor, Window Editor, and <br> ZoomTrig |

Note: More information is available about using setMode. r

## Graph Picture and Database Commands

| Command | Description |
| :--- | :--- |
| AndPic | Displays the Graph screen and superimposes a stored <br> graph picture by using AND logic. |


| Command | Description |
| :--- | :--- |
| CyclePic | Animates a series of stored graph pictures. |
| NewPic | Creates a graph picture variable based on a matrix. |
| RcIGDB | Restores all settings stored in a graph database. |
| RcIPic | Displays the Graph screen and superimposes a stored <br> graph picture by using OR logic. |
| RpIcPic | Clears the Graph screen and displays a stored graph <br> picture. |
| StoGDB | Stores the current graph settings to a graph database <br> variable. |
| StoPic | Copies the Graph screen (or a specified rectangular <br> portion) to a graph picture variable. |
| XorPic | Displays the Graph screen and superimposes a stored <br> graph picture by using XOR logic. |

Note: For information about graph pictures and databases, also refer to Additional Graphing Topics.

## Drawing on the Graph Screen

To create a drawing object on the Graph screen, use the commands listed in this section.

## Pixel vs. Point Coordinates

When drawing an object, you can use either of two coordinate systems to specify a location on the screen.

- Pixel coordinates - Refer to the pixels that physically make up the screen. These are independent of the viewing window because the screen is always: 159 (0 to 158) pixels wide and 77 ( 0 to 76 ) pixels tall.
- Point coordinates - Refer to the coordinates in effect for the current viewing window (as defined in the Window Editor).


Pixel coordinates (independent of viewing window)


Point coordinates (for standard viewing window)

Note: For information about pixel coordinates in split screens, refer to the Data/Matrix Editor module.

Many drawing commands have two forms: one for pixel coordinates and one for point coordinates.

Note: Pixel commands start with PxI, such as PxIChg.

## Erasing Drawn Objects

| Command | Description |
| :--- | :--- |
| CIrDraw | Erases all drawn objects from the Graph screen. |

## Drawing a Point or Pixel

| Command | Description |
| :--- | :--- |
| PtChg or <br> PxIChg | Toggles (inverts) a pixel at the specified coordinates. <br> PtChg, which uses point coordinates, affects the pixel <br> closest to the specified point. If the pixel is off, it is turned <br> on. If the pixel is on, it is turned off. |
| PtOff or | Turns off (erases) a pixel at the specified coordinates. <br> PtOff, which uses point coordinates, affects the pixel <br> closest to the specified point. |
| PtOn or Turns on (displays) a pixel at the specified coordinates. <br> PtOn, which uses point coordinates, affects the pixel <br> closest to the specified point. <br> PtTest or <br> PxITest Returns true or false to indicate if the specified coordinate <br> is on or off, respectively. <br> PtText or Displays a character string at the specified coordinates. <br> PxIText  |  |

## Drawing Lines and Circles

| Command | Description |
| :--- | :--- |
| Circle or <br> PxICrcl | Draws, erases, or inverts a circle with a specified center <br> and radius. |
| DrawSIp | Draws a line with a specified slope through a specified <br> point. |
| Line or <br> PxILine | Draws, erases, or inverts a line between two sets of <br> coordinates. |

Command Description

| LineHorz or | Draws, erases, or inverts a horizontal line at a specified |
| :--- | :--- |
| PxIHorz | row coordinate. |


| LineTan | Draws a tangent line for a specified expression at a <br> specified point. (This draws the tangent line only, not the <br> expression.) |
| :--- | :--- |

LineVert or Draws, erases, or inverts a vertical line at a specified PxiVert column coordinate.

## Drawing Expressions

| Command | Description |
| :--- | :--- |
| DrawFunc | Draws a specified expression. |
| DrawInv | Draws the inverse of a specified expression. |
| DrawParm | Draws a parametric equation using specified expressions <br> as its $x$ and y components. |
| DrawPol | Draws a specified polar expression. |
| DrwCtour | Draws contours in 3D graphing mode. |
| Shade | Draws two expressions and shades the areas where <br> expression1 < expression2. |

## Accessing Another TI-89 Titanium, a CBL 2, or a CBR

If you link two graphing calculators (described in the Connectivity module), programs on both units can transmit variables between them. If you link a TI-89 Titanium to a Calculator-Based Laboratory ${ }^{\text {TM }}$ (CBL $2^{\text {TM }}$ ) or a Calculator-Based Ranger ${ }^{\text {TM }}$ (CBR ${ }^{\text {TM }}$ ), a program on the TI-89 Titanium can access the CBL 2 or CBR.

## F3 I/O Toolbar Menu

Use the Program Editor's F3 I/O toolbar menu to enter the commands in this section.

1. Press F3 and select 8 :Link.
2. Select a command.


## Accessing Another TI-89 Titanium

When two calculators are linked, one acts as a receiving unit and the other as a sending unit.

| Command | Description |
| :---: | :---: |
| GetCalc | Executed on the receiving unit. Sets up the unit to receive a variable via the I/O port. <br> - After the receiving unit executes GetCalc, the sending unit must execute SendCalc. <br> - After the sending unit executes SendCalc, the sent variable is stored on the receiving unit (in the variable name specified by GetCalc). |
| SendCalc | Executed on the sending unit. Sends a variable to the receiving unit via the I/O port. <br> - Before the sending unit executes SendCalc, the receiving unit must execute GetCalc. |

Note: For a sample program that synchronizes the receiving and sending units so that GetCalc and SendCalc are executed in the proper sequence, refer to "Transmitting Variables under Program Control" in Connectivity.

## Accessing a CBL 2 or CBR

For additional information, refer to the manual that comes with the CBL 2 or CBR unit.

| Command | Description |
| :--- | :--- |
| Get | Gets a variable from an attached CBL 2 or CBR and <br> stores it in the graphing calculator. |

Command Description

Send
Sends a list variable from the graphing calculator to the CBL 2 or CBR.

## Debugging Programs and Handling Errors

After you write a program, you can use several techniques to find and correct errors. You can also build an error-handling command into the program itself.

## Run-Time Errors

The first step in debugging your program is to run it. The graphing calculator automatically checks each executed command for syntax errors. If there is an error, a message indicates the nature of the error.

- To display the program in the Program Editor, press ENTER. The cursor appears in the approximate area of the error.

- To cancel program execution and return to the Home screen, press ESC.

If your program allows the user to select from several options, be sure to run the program and test each option.

## Debugging Techniques

Run-time error messages can locate syntax errors but not errors in program logic. The following techniques may be useful.

- During testing, do not use local variables so that you can check the variable values after the program stops. When the program is debugged, declare the applicable variables as local.
- Within a program, temporarily insert Disp and Pause commands to display the values of critical variables.
- Disp and Pause cannot be used in a user-defined function. To temporarily change the function into a program, change Func and EndFunc to Prgm and EndPrgm. Use Disp and Pause to debug the program. Then remove Disp and Pause and change the program back into a function.
- To confirm that a loop is executed the correct number of times, display the counter variable or the values in the conditional test.
- To confirm that a subroutine is executed, display messages such as Entering subroutine and Exiting subroutine at the beginning and end of the subroutine.


## Error-Handling Commands

| Command | Description |
| :--- | :--- |
| Try...EndTry | Defines a program block that lets the program execute a <br> command and, if necessary, recover from an error <br> generated by that command. |
| CIrErr | Clears the error status and sets the error number in system <br> variable Errornum to zero. |
| PassErr | Passes an error to the next level of the Try...EndTry block. |

## Example: Using Alternative Approaches

The example in the Previews module shows a program that prompts the user to enter an integer, sums all integers from 1 to the entered integer, and displays the result. This section gives several approaches that you can use to achieve the same goal.

## Example 1

This example uses InputStr for input, a While...EndWhile loop to calculate the result, and Text to display the result.

```
    :prog1 ()
    : Prgm
(1) : InputStr "Enter an integer", n
(2) \(: \operatorname{expr}(\mathrm{n}) \rightarrow \mathrm{n}\)
    \(: 0 \rightarrow\) temp: \(1 \rightarrow\) I
(3):While isn
: : temp+i \(\rightarrow\) temp
: : i+1 \(\rightarrow\) I
(3) : EndWhile
4 : Text "The answer is "\&string (temp)
    : EndPrgm
```

(1) Prompts for input on Program I/O screen.
(2) Converts string entered with InputStr to an expression.

3 Loop calculation.
(4) Displays output in a dialog box.

Note: For $\leq$, type 0 (zero). For \&, press:

- ® (times)


## Example 2

This example uses Prompt for input, LbI, and Goto to create a loop, and Disp to display the result.

```
    :prog2()
    :Prgm
(1) : Prompt n
    :0->temp: 1->I
(2):Lbl top
: : temp+i->temp
    : i+1->I
    : If i\leqn
(2) : Goto top
3 :Disp temp
    :EndPrgm
```

(1) Prompts for input on Program I/O screen.
(2) Loop calculation.
(3) Displays output on Program I/O screen.

Note: Because Prompt returns n as a number, you do not need to use expr to convert n .

## Example 3

This example uses Dialog...EndDlog to create dialog boxes for input and output. It uses Loop...EndLoop to calculate the result.

```
    :prog3()
    :Prgm
(1):Dialog
    : Title "Enter an integer"
    : Request "Integer",n
(1) :EndDlog
(2): }\operatorname{expr}(n)->
    :0->temp:0->I
3 : Loop
: : temp+i->temp
    : i+1->I
    : If i>n
    : Exit
3 : EndLoop
4):Dialog
    : Title "The answer is"
    : Text string(temp)
4 : EndDlog
    :EndPrgm
```

(1) Defines a dialog box for input.
(2) Converts string entered with Request to an expression.
(3) Loop calculation.
(4) Defines a dialog box for output.

## Example 4

This example uses built-in functions to calculate the result without using a loop.

```
    :prog4 ()
    : Prgm
(1) :Input "Enter an integer",n
(2) \(: \operatorname{sum}(\operatorname{seq}(i, i, 1, n)) \rightarrow\) temp
(3):Disp temp
    :EndPrgm
```

(1) Prompts for input on Program I/O.
(2) Calculates sum.
(3) Displays output on Program I/O screen.

Note: Because Input returns $n$ as a number, you do not need to use expr to convert $n$.

| Function | Used in this example to: |
| :---: | :---: |
| seq | Generate the sequence of integers from 1 to n . |
|  | seq(expression, var, low, high [,step]) |
|  | (1) (2) 3 8 $\mathbf{3}^{4}$ |
|  | (1) expression used to generate the sequence <br> (2) variable that will be incremented <br> (3) initial and final values of var <br> 4 increment for var ; if omitted, uses 1 |
| sum | Sum the integers in the list generated by seq. |

## Assembly-Language Programs

You can run programs written for the TI-89 Titanium in assembly language. Typically, assembly-language programs run much faster and provide greater control than the keystroke programs that you write with the built-in Program Editor.

## Where to Get Assembly-Language Programs

Assembly-language programs, as well as keystroke programs, are available on the Texas Instruments web site at education.ti.com.

The programs available from this site provide additional functions or features that are not built into the TI-89 Titanium. Check the Texas Instruments web site for up-to-date information.

After downloading a program from the web to your computer, use a USB cable or TI-GRAPH LINK ${ }^{\text {TM }}$ computer-to-calculator cable and TI Connect software to send the program to your TI-89 Titanium.

For Flash App installation instructions, see education.ti.com/guides.

## Note about TI-GRAPH LINK

If you have a TI-GRAPH LINK ${ }^{\text {TM }}$ computer-to-calculator cable and software for the TI-89 or TI-92 Plus, be aware that the TI-GRAPH LINK software is not compatible with the TI-89 Titanium. The cable, however, works with all units. Use TI Connect software on your computer.

You can purchase computer-to-calculator and unit-to-unit cables from the TI Online Store at education.ti.com/buy.

## Running an Assembly-Language Program

After a TI-89 Titanium assembly-language program is stored on your unit, you can run the program from the Home screen just as you would any other program.

- If the program requires one or more arguments, type them within the (). Refer to the program's documentation to find out about required arguments.

- If the program is not in the current folder, be sure to specify the pathname.

You can call an assembly-language program from another program as a subroutine, delete it, or use it the same as any other program.

## Shortcuts to Run a Program

On the Home screen, you can use keyboard shortcuts to run up to nine user-defined or assembly-language programs. However, the programs must have the following names.

| On Home screen, press: | To run a program, if any, named: |
| :---: | :---: |
|  | kbdprgm1 ( ) |
| $\vdots$ | $\vdots$ |
|  | 9 |

The programs must be stored in the MAin folder. Also, you cannot use a shortcut to run a program that requires an argument.

If you have a program with a different name and you would like to run it with a keyboard shortcut, copy or rename the existing program to kbdprgm1( ), etc.

## You Cannot Edit an Assembly-Language Program

You cannot use your TI-89 Titanium to edit an assembly-language program. The built-in Program Editor will not open assembly-language programs.

## Displaying a List of Assembly-Language Programs

To list the assembly-language programs stored in memory:

1. Display the VAR-LINK screen ([2nd [VAR-LINK]).
2. Press F2 View.
3. Select the applicable folder (or All folders) and set Var Type = Assembly.

4. Press ENTER to display the list of assembly-language programs.


Note: Assembly-language programs have an ASM data type.

## For Information about Writing an Assembly-Language Program

The information required to teach a novice programmer how to write an assemblylanguage program is beyond the scope of this book. However, if you have a working knowledge of assembly language, please check the Texas Instruments web site (education.ti.com) for specific information about how to access TI-89 Titanium features.

The graphing calculator also includes an Exec command that executes a string consisting of a series of Motorola 68000 op-codes. These codes act as another form of an assembly-language program. Check the Texas Instruments web site for available information.

Note: You must use a computer to write assembly-language programs. You cannot create assembly-language programs from the calculator keyboard.

Warning: Exec gives you access to the full power of the microprocessor. Please be aware that you can easily make a mistake that locks up the calculator and causes you to lose your data. We suggest you make a backup of the calculator contents before attempting to use the Exec command.

## Text Editor

## Starting a Text Editor Session

Each time you start the Text Editor, you can start a new text session, resume the current session (the session that was displayed the last time you used the Text Editor), or open a previous session.

## Starting a New Session

1. Press APPS and then select.
2. Select 3:New.

The NEW dialog box is displayed.

That Editor
1:Current. 2: Open. SREW.
3. Specify a folder and text variable that you want to use to store the new session.


| Item | Description |
| :--- | :--- |
| Type | Automatically set as Text and cannot be changed. |
| Folder | Shows the folder in which the text variable will be stored. <br>  <br>  <br>  <br>  <br> For information about folders, refer to the Calculator Home <br> Screen module. <br> To use a different folder, press © $(\square)$ to display a menu of <br> existing folders. Then select a folder. |


| Item | Description |
| :--- | :--- |
| Variable | Type a variable name. <br> If you specify a variable that already exists, an error <br> message will be displayed when you press ENTER. When <br> you press ESC or ENTER to acknowledge the error, the |
|  | NEW dialog box is redisplayed. |

4. Press ENTER (after typing in an input box such as Variable, you must press ENTER twice) to display an empty Text Editor screen.


You can now use the Text Editor as described in the remaining sections of this module.

Note: Your session is saved automatically as you type. You do not need to save a session manually before leaving the Text Editor, starting a new session, or opening a previous one.

## Resuming the Current Session

You can leave the Text Editor and go to another application at any time. To return to the session that was displayed when you left the Text Editor, launch Text Editor again and select 1:Current.

## Starting a New Session from the Text Editor

To leave the current Text Editor session and start a new one:

1. Press F1 and select 3:New.
2. Specify a folder and text variable for the

## Tir

 new session.
3. Press ENTER twice.

## Opening a Previous Session

You can open a previous Text Editor session at any time.

1. From within the Text Editor, press F1 and select 1:Open.

- or -

From any application, launch Text Editor again and select 2:Open.
2. Select the applicable folder and text variable.
3. Press ENTER.

| DFEH |  |
| :---: | :---: |
| TuFst Tixt |  |
| Folder: main' |  |
| Warintic: test 9 |  |
| Enter=0K | ESC=CNACEL |

Note: By default, Variable shows the first existing text variable in alphabetic order.

## Copying a Session

In some cases, you may want to copy a session so that you can edit the copy while retaining the original.

1. Display the session you want to copy.
2. Press F1 and select 2:Save Copy As.
3. Specify the folder and text variable for the copied session.
4. Press ENTER twice.

## Note about Deleting a Session

Because all Text Editor sessions are saved automatically, you can accumulate quite a few previous sessions, which take up memory storage space.

To delete a session, use the VAR-LINK screen (2nd [VAR-LINK]) to delete that session's text variable. For information about VAR-LINK, refer to Memory and Variable Management.

## Entering and Editing Text

After beginning a Text Editor session, you can enter and edit text. In general, use the same techniques that you have already used to enter and edit information on the Home screen's entry line.

## Typing Text

When you create a new Text Editor session, you see an empty screen. When you open a previous session or return to the current session, you see the existing text for that session.

All text paragraphs begin with a space and a colon.

The beginning space is used in command scripts and lab reports.


You do not need to press ENTER at the end of each line. At the end of a line, the next character you type wraps to the next line. Press ENTER only when you want to start a new paragraph.

As you reach the bottom of the screen, previous lines scroll off the top of the screen.
Using a USB cable and TI Connect ${ }^{\text {TM }}$ software with the TI-89 Titanium, you can use the computer keyboard to type a text file and then send that file to the TI-89 Titanium. This is useful if you need to create a lengthy text file.

For information about obtaining cables or updated TI Connect ${ }^{\text {TM }}$ software, check the TI web site at education.ti.com, or contact Texas Instruments at TI-Cares ${ }^{\mathrm{TM}}$.

## Notes:

- Use the cursor pad to scroll through a session or position the text cursor.
- Press 2 nd $\Theta$ or 2nd $\Theta$ to scroll up or down one screen at a time, and $\bullet \Theta$ or $\bullet \Theta$ to go to the top or bottom of the text session.


## Typing Alphabetic Characters

| To: | Press: |
| :---: | :---: |
| Type a single lowercase alpha character. | [ alpha and then the letter key (status line shows d) |
| Type a single uppercase alpha character. | ( $\dagger$ and then the letter key (status line shows $\bullet$ ) |
| Type a space. |  |
| Turn on lowercase alphalock. | ( 2 2nd [ $\mathrm{a}-\mathrm{lock}]$ (status line shows [-m) |
| Turn on uppercase ALPHA-lock. |  |
| Turn off alphalock. | (0] alpha (turns off upper- and lowercase lock) |

Note: On the TI-89 Titanium, you do not need alpha or alpha-lock to type x, y, z, or t. But you must use $\dagger$ or uppercase ALPHA-lock for X, Y, Z, or T. On the TI-89 Titanium, alpha-lock is always turned off when you change applications, such as going from the Text Editor to the Home screen.

On the $\mathrm{TI}-89$ Titanium, while either type of alpha-lock is on:

- To type a period, comma, or other character that is the primary function of a key, you must turn alpha-lock off.
- To type a second function character such as 2nd [i], you do not need to turn alphalock off. After you type the character, alpha-lock remains on.


## Deleting Characters

| To delete: | Press: |
| :--- | :--- |
| The character to the left of the cursor | $\square$ or F1 7 |
| The character to the right of the cursor | $[D E L]$ (same as |
| All characters to the right of the cursor <br> through the end of the paragraph | CLEAR |
| All characters in the paragraph (regardless <br> of the cursor's position in that paragraph) | CLEAR CLEAR |

Note: If there are no characters to the right of the cursor, CLEAR erases the entire paragraph.

## Highlighting Text

| To: | Do this: |
| :---: | :---: |
| Highlight text | Move the cursor to the beginning or end of the text. Hold 1 and press: <br> - (1) or (1) to highlight characters to the left or right of the cursor, respectively. <br> - $\Theta$ or $\Theta$ to highlight all characters up to the cursor position on the next or previous line, respectively. |
|  |  |

Note: To remove highlighting without replacing or deleting, move the cursor.

## Replacing or Deleting Highlighted Text

To: Do this:

Replace highlighted text Type the new text.
Delete highlighted text Press $\square$.

## Cutting, Copying, and Pasting Text

Cutting and copying both place highlighted text into the clipboard of the TI-89 Titanium. Cutting deletes the text from its current location (used to move text) and copying leaves the text.

1. Highlight the text you want to move or copy.
2. Press F1.
3. Select the applicable menu item.

- To move the text, select 4:Cut.
- or -

- To copy the text, select 5:Copy.

Note: You can press:
:- $\quad$ [CUT], $\bullet$ [COPY], $\bullet$ [PASTE]
to cut, copy, and paste without having to use the F1 toolbar menu.
4. Move the text cursor to the location where you want to insert the text.
5. Press $\mathbb{F 1}$ and then select 6:Paste.

You can use this general procedure to cut, copy, and paste text:

- Within the same text session.
- From one text session to another. After cutting or copying text in one session, open the other session and then paste the text.
- From a text session to a different application. For example, you can paste the text into the Home screen's entry line.


## Finding Text

From the Text Editor:

1. Place the text cursor at any location preceding the text you want to search for. All searches start at the current cursor location.
2. Press F5.

3. Type the search text.

The search is not case sensitive. For example: CASE, case, and Case have the same effect.

Note: The FIND dialog box retains the last search text you entered. You can type over it or edit it.
4. Press ENTER twice.

If the search text is: The cursor:
Found Moves to beginning of the search text.
Not found Does not move.

## Inserting or Overtyping a Character

By default, the TI-89 Titanium is in insert mode. To toggle between insert and overtype mode, press [2nd [INS].

| If the Tl-89 Titanium is in: | The next character you type: |
| :--- | :--- |
| Inserfe mode | Will be inserted at the cursor. |
| Thin cursor between |  |
| characters |  |

Note: Look at the shape of the cursor to see if you're in insert or overtype mode.

## Clearing the Text Editor

To erase all existing paragraphs and display an empty text screen, press F1 and then select 8:Clear Editor.

## Entering Special Characters

You can use the CHAR menu to select any special character from a list. You can also type certain commonly used characters from the keyboard. To see which characters are available from the keyboard, you can display a map that shows the characters and their corresponding keys.

## Selecting Characters from the CHAR Menu

1. Press 2nd [Char].
2. Select the applicable category.

A menu lists the characters in that category.
3. Select a character. You may need to scroll through the menu.


Note: For accented characters, select International. Commonly used international characters are also available from the default custom menu (2nd [Custom]).

## Displaying the Keyboard Map

The keyboard map shows several shortcuts that let you enter certain special characters from the keyboard. It also shows some shortcuts for other calculator features.

The keyboard map does not display all available shortcuts. Refer to the inside front and the inside back covers of this guidebook for a complete list of shortcut keys.

To access the shortcuts, first press the 2nd key. Some special characters are marked on the keyboard, but most are not.

## On the TI-89 Titanium:

- Press EEE to display the keyboard map.
- Press ESC to exit the map.



## TI-89 Titanium Keyboard map

To access the TI-89 Titanium shortcuts, first press the key.

## TI-89 Titanium keyboard map feature shortcuts:

- GREEK ( $\triangle$ ) - Accesses the Greek character set (described later in this section).
- SYSDATA $(\bullet \square)$ - Copies the current graph coordinates to the system variable sysdata.
- FMT ( $\square \square)$ — Displays the FORMATS dialog box.
- KBDPRGM1 - $9(\square 1$ through $\bullet 9)$ - If you have user-defined or assemblylanguage programs named kbdprgm1() through kbdprgm9(), these shortcuts run the corresponding program.
- OFF ( $\triangle$ [0FF]) — Similar to 2nd [0FF] except:
- You can use [0FF] if an error message is displayed.
- When you turn the TI-89 Titanium on again, it will be exactly as you left it.
- HOMEDATA $(\square)$ - Copies the current graph coordinates to the Home screen's history area.


## Typing Special Symbols from the Keyboard

Note: To help you find the applicable keys, these maps show only the special symbols.

## On the TI-89 Titanium:

$\rightarrow$ Press $\square$ and then the key for the symbol. For example: $\square$ (times) displays \&


These special symbols are not affected by whether Alpha-Lock is on or off.

## Typing Greek Letters from the Keyboard

Press the key combination that accesses the Greek character set on your calculator. Then select the applicable alpha character on the keyboard to enter a Greek letter.

## On the TI-89 Titanium:

- Press $\square$ to access the Greek character set.


Note: If you press a key combination that does not access a Greek letter, you get the normal letter for that key. Your calculator does not display a map of Greek letters; the map shown here is for reference only.

Several keys let you access lowercase and uppercase Greek letters. For example:

## On the TI-89 Titanium:

- Press $\square$ to access the Greek character set.
- Press $\quad \square$ alpha + letter to access lowercase Greek letters. Example:
- $\square$ alpha [W] displays $\omega$
- Press $\square \square+$ letter to access uppercase Greek letters. Example:
$\rightarrow \square \square$ [W] displays $\Omega$
The exact keys that you press on the TI-89 Titanium depend on whether alpha-lock is on or off. For example:

| On the TI-89 Titanium, if: | Then: |
| :---: | :---: |
| Alpha-lock is off. | $\rightarrow \square \mathbf{X}$ or $\rightarrow$ alpha $\mathbf{X}$ displays $\xi$. (alpha is not required for $\mathrm{X}, \mathrm{Y}, \mathrm{Z}$, or T .) <br> - alpha W displays $\omega$. <br> $\rightarrow \square \mathbf{W}$ displays $\Omega$. <br> ( $\uparrow$ is used for uppercase letters.) |
| Lowercase alpha-lock (2nd [a-lock]) is on. | $\rightarrow$ X displays $\xi$. <br> $\square \square$ W displays $\omega$. <br> - 1 T W displays $\Omega$. |
| Uppercase ALPHA-LOCK <br> ( 1 [a-lock]) is on. | - X displays $\xi$. <br> - $\square \mathbf{W}$ displays $\Omega$. <br> - 0 W displays $\Omega$. |

Important: If you press alpha on the TI-89 Titanium to access a Greek letter while alphalock is on, it turns alpha-lock off.

## For a List of All Special Characters

For a list of all special characters, refer to the Technical Reference module.

## Entering and Executing a Command Script

By using a command script, you can use the Text Editor to type a series of command lines that can be executed at any time on the Home screen. This lets you create interactive example scripts in which you predefine a series of commands and then execute them individually.

## Inserting a Command Mark

In the Text Editor:

1. Place the cursor on the line for the command.
2. Press F2 to display the Command toolbar menu.
3. Select 1:Command.
$\mathbf{C}$ is displayed at the beginning of the text line (to the left of the colon).
Note: This does not insert a new line for
the command, it simply marks an existing
Note: This does not insert a new line for
the command, it simply marks an existing line as a command line.
4. Type a command just as you would on the Home screen.

The line can contain only the command, with no additional text.

Note: You can mark a line as a command either before or after typing the command on that line.

You can type multiple commands on the same line if you type a colon to separate the commands.

## Deleting a Command Mark

This deletes only the C mark; it does not delete the command text itself.

1. Place the cursor anywhere on the marked line.
2. Press F2 and select 4:Clear command.

## Executing a Command

To execute a command, you must first mark the line with a C. If you execute a line that is not marked with $\mathbf{C}$, it will be ignored.

1. Place the cursor anywhere on the command line.
2. Press F4.

The command is copied to the entry line on the Home screen and executed. The Home screen is displayed temporarily during execution, and then the Text Editor is redisplayed.

After execution, the cursor moves to the next line in the script so that you can continue to execute a series of commands.

Note: To examine the result on the Home screen, use a split screen or press [ HOME

## Splitting the Text Editor/ Home Screen

With a split screen, you can view your command script and see the result of an executed command at the same time.

| To: | Press: |  |
| :--- | :--- | :--- |
| Split the screen | F3 and select  <br>  1:Script view. |  |
|  |  |  |
| Return to a full screen <br> Text Editor | F3 and select |  |

You can also use MODE to set up a split screen manually. However, F3 sets up a Text Editor/Home screen split much easier than MODE.

- The active application is indicated by a thick border. (By default, the Text Editor is the active application.)
- To switch between the Text Editor and the Home screen, press 2nd [ $\boxplus$ ] (second function of APPS).


## Creating a Script from Your Home Screen Entries

From the Home screen, you can save all the entries in the history area to a text variable. The entries are automatically saved in a script format so that you can open the text variable in the Text Editor and execute the entries as commands.

For information, refer to "Saving the Home Screen Entries as a Text Editor Script" in the Calculator Home Screen module.

## Example

1. Type your script. Press F2 and select 1:Command to mark the command lines.
2. Press F3 and select 1:Script view.

| Fir FET FFT F4 Fid |  |
| :---: | :---: |
| OWindou for ${ }^{\text {creaph }}$ |  |
| $0: x^{*} 3-2 x-2+x-1+f(x)$ |  |
| C:zeros (f(x), $x$ ) |  |
| $\mathrm{E}=\mathrm{d}(\mathrm{f}(\mathrm{x}), \mathrm{x})+\mathrm{df}(\mathrm{x})$ |  |
| C: zeros (df (x), x) |  |
| $0: d(d f(x), x)+d d f(x)$ |  |
| C: $-4+\mathrm{xmini} 4+\times \mathrm{max}$ |  |
| C: -10 |  |
| -6-sph |  |
| MAINT Fifictilla FUMEC |  |

3. Move the cursor to the first command line. Then press F4 to execute the command.
Note: Some commands take longer to execute. Wait until the Busy indicator disappears before pressing F4 again.

4. Continue using F4 to execute each command, but stop just before executing the Graph command.
5. Execute the Graph command.

Note: In this example, the Graph command displays the Graph screen in place of the Home screen.
6. Press F3 and select 2:Clear split to return
 to a full screen Text Editor.

## Numeric Solver

## Displaying the Solver and Entering an Equation

After you display the Numeric Solver, start by entering the equation that you want to solve.

## Displaying the Numeric Solver

To display the Numeric Solver, press APPS and then select.
$f(x)=0$
Numeric so...

The Numeric Solver screen shows the last entered equation, if any.

## Entering an Equation

On the eqn: line, type in your equation.

Enter Eqtation
력: $\quad \exists=(\mathrm{m} 2-m 1) /(\mathrm{m} \overline{2}+m 1) * g$

| You can: | For example: |
| :--- | :--- |
| Type an equation directly. | $\mathrm{a}=(\mathrm{m} 2-\mathrm{m} 1) /(\mathrm{m} 2+\mathrm{m} 1) * g$ <br> $\mathrm{a}+\mathrm{b}=\mathrm{c}+\sin (\mathrm{d})$ |

## You can:

Refer to a function or equation defined elsewhere.

## Notes:

- Do not use system function names (such as $\mathbf{y 1}(\mathbf{x})$ or $\mathbf{r 1}(\theta)$ ) as simple variables ( $\mathbf{y} \mathbf{1}$ or $\mathbf{r} \mathbf{1}$ ).
- Be careful with implied multiplication. For example, $a(m 2+m 1)$ is treated as a function reference, not as $a *(m 2+m 1)$.


## For example:

Suppose you defined $\mathbf{y 1}(\mathbf{x})$ on either the:

- $Y=$ Editor:
$\mathrm{y} 1(\mathrm{x})=1.25 \mathrm{x} * \cos (\mathrm{x})$
- or -
- Home screen:

Define $\mathbf{y 1}(\mathrm{x})=1.25 \mathrm{x} * \cos (\mathrm{x})$
In the Numeric Solver, you then would enter:

| $\mathrm{y} 1(\mathrm{x})=0$ or $\mathrm{y} 1(\mathrm{t})=\mathbf{0}$, etc. |
| ---: | :--- |
| The argument does | not have to match the one used to define the function or equation.


| Type an expression without an $=$ | e+f-ln(g) |
| :--- | :--- |
| sign. | After you press ENTER, the expression |
| Note: When you define the | is set equal to a system variable called |
| variables, you can either define |  |
| exp or solve for it. | exp and entered as: <br> exp=e $\mathbf{f}-\ln (\mathrm{g})$ |
| Recall a previously entered | Refer to the applicable heading later in <br> equation or open a saved <br> this section. | equation.

Note: After you press ENTER the current equation is stored automatically to the system variable eqn.

## Recalling Previously Entered Equations

Your most recently entered equations (up to 11 with the default setting) are retained in memory. To recall one of these equations:

1. From the Numeric Solver screen, press F5.

A dialog box displays the most recently entered equation.
2. Select an equation.

- To select the displayed equation, press ENTER.
- To select a different equation, press (1) to display a list. Then select the one you want.
Note: You can specify how many equations are retained. From the Numeric Solver, press F1 and select 9:Format (or use 目 (1). Then select a number from 1 through 11.

3. Press ENTER.


Only unique equations are listed. If you reenter the same equation 5 times, it appears only once.

## Saving Equations for Future Use

Because the number of equations that you can recall with F5 Eqns is limited, a particular equation may not be retained indefinitely.

To store the current equation for future use, save it to a variable.

1. From the Numeric Solver screen, press F1 and select 2:Save Copy As.

2. Specify a folder and a variable name for the equation.
3. Press ENTER twice.


Note: An equation variable has an EXPR data type, as shown on the MEMORY and VAR-LINK screens.

## Opening a Saved Equation

To open a previously saved equation variable:

1. From the Numeric Solver screen, press F1 and select 1:Open.

2. Select the applicable folder and equation variable.
3. Press ENTER.


Variable eqn contains the current equation; it always appears alphabetically in the list.

## Defining the Known Variables

After you type an equation in the Numeric Solver, enter the applicable values for all variables except the unknown variable.

## Defining the List of Variables

After typing your equation on the eqn: line, press ENTER or $\Theta$.

The screen lists the variables in the order they appear in the equation. If a variable is already defined, its value is shown. You can edit these variable values.

Note: If an existing variable is locked or archived, you cannot edit its value.

Enter a number or expression for all variables except the one you want to solve for.

## Notes and Common Errors

- If you define a variable:
- In terms of another variable in the equation, that variable must be defined first.
- In terms of another variable that is not in the equation, that variable must already have a value; it cannot be undefined.
- As an expression, it is evaluated when you move the cursor off the line.

```
F17 FE
\exists=(mz-m1)/(mZ+m1)*g
g=9/3
m}2=10
m1=
g=9.8
bound=<-1.E14, 1. E14)
```

Since $a$ is defined in terms of g , you must define $g$ before a. When you move the cursor to another line, $\mathrm{g} / 3$ is evaluated.

The expression must evaluate to a real number

- If the equation contains a variable already defined in terms of other variables, those other variables are listed.

Note: When you assign a value to a variable in the Numeric Solver, that variable is defined globally. It still exists after you leave the solver.

- If you refer to a previously defined function, any variables used as arguments in the function call are listed, not the variables used to define the function.

|  |
| :---: |
| If $f(a, b)$ was defined previously as $\left.\sqrt{( } a^{\wedge} 2+b^{\wedge} 2\right)$ and your equation contains $f(x, y)$, then $x$ and $y$ are listed, not a and b. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |



```
x MEx=2* 人+2*!- 
    x=
    y=
```



In the standard viewing window, $x$ max=10.

- If you see the error shown to the right, delete the entered variable value. Then edit the equation to use a different variable.

Note: This error occurs if you use a reserved name incorrectly or refer to an undefined system function as a simple variable without parentheses.


For example, $\mathrm{y} 1(\mathrm{x})$ is undefined and you use y1.

## Editing the Equation

In the Numeric Solver, press $\Theta$ until the cursor is on the equation. The screen automatically changes to show only the eqn: line. Make your changes, and then press ENTER or $\Theta$ to return to the list of variables.

## Specifying an Initial Guess and/or Bounds (Optional)

To find a solution more quickly or to find a particular solution (if multiple solutions exist), you can optionally:

- Enter an initial guess for the unknown variable. The guess must be within the specified bounds.
- Enter lower and upper bounds close to the solution.


Initial guess must be within the bounds.

For the bounds, you can also enter variables or expressions that evaluate to appropriate values (bound=\{lower,upper\}) or a valid list variable that contains a two-element list (bound=list). The bounds must be two floating point elements with the first one less than or equal to the second one.

Note: You can also select an initial guess graphically.

## Solving for the Unknown Variable

After you type an equation in the Numeric Solver and enter values for the known variables, you are ready to solve for the unknown variable.

## Finding the Solution

With all known variables defined:

1. Move the cursor to the unknown variable.


Put the cursor on the variable you want to solve for.


Note: To stop (break) a calculation, press 0 N . The unknown variable shows the value being tested when the break occurred.

Using the solution and your entered values, the left and right sides of the equation are evaluated separately. left-rt shows the difference, which indicates the solution's accuracy. The smaller the value, the more accurate the solution. If the solution is precise, left-rt=0.
If you: Do this:
Want to solve for other Edit the equation or variable values. values

| Want to find a different <br> solution for an equation <br> with multiple solutions | Enter an initial guess and/or a new set of <br> bounds close to the other solution. |
| :--- | :--- |

See the message: Press ESC. The unknown variable shows the value being tested when the error occurred.


- The left-rt value may be small enough for you to accept the result.
EGCCHCEL - If not, enter a different set of bounds.

Note: An iterative process is used to solve an equation. If the iterative process cannot converge on a solution, this error occurs.

## Graphing the Solution

You can graph an equation's solutions any time after defining the known variables, either before or after you solve for the unknown variable. By graphing the solutions, you can
see how many solutions exist and use the cursor to select an accurate initial guess and bounds.

## Displaying the Graph

In the Numeric Solver, leave the cursor on the unknown variable. Press F3 and select:

1:Graph View

- or -

3:ZoomStd

- or -

4:ZoomFit

## FST

1: Brapfo View
 3: ZoomSta
4: ZoomFit
Graph View uses the current Window variable values.

For information about
ZoomStd and ZoomFit, refer to
Basic Function
Graphing.
The graph is shown in a split screen, where:

- The unknown variable is plotted on the $x$ axis.
- left-rt is plotted on the $y$ axis.

Solutions for the equation exist at left-rt=0, where the graph crosses the $x$ axis.

Note: For more information, refer to the Split Screens module.


The current graph format settings are used.

You can explore the graph by using the free-moving cursor, tracing, zooming, etc., as described in Basic Function Graphing.

## How the Graph Affects Various Settings

When you use the Numeric Solver to display a graph:

- The following modes are changed automatically to these settings:

| Mode | Setting |
| :--- | :--- |
| Graph | FUNCTION |
|  | Any functions selected in the <br>  <br>  <br>  <br> Split Scditor will not be graphed. |
| Number of Graphs | LEFT-RIGHT |

Note: If you were previously using different mode settings, you will need to reselect those settings manually.

- All stat plots are deselected.
- After you leave the Numeric Solver, the Graph screen may continue to display the equation's solution, ignoring any selected $\mathrm{Y}=$ functions. If so, display the $\mathrm{Y}=$ Editor and then return to the Graph screen. Also, the graph is reset when you change the Graph mode or use CIrGraph from the Home screen (F4 5) or a program.


## Selecting a New Initial Guess from the Graph

To use the graph cursor to select an initial guess:

1. Move the cursor (either free-moving or trace) to the point that you want to use as the new guess.
2. Use [2nd [ $\omega$ ] to make the Numeric Solver screen active.
3. Make sure the cursor is on the unknown variable, and press F4.

Note: Cursor coordinate xc is the unknown variable value, and yc is the left-rt value.
4. Press F2 to re-solve the equation.

| Eric |  |
| :---: | :---: |
|  |  |
|  |  |
| left-rt | -4. |
|  |  |
| -F4] sets the grap |  |
| cursor's xc value as |  |
| an initial guess and |  |
| the yc value as left-rt. |  |
| The graph's xmin and |  |
| max values are set |  |
| as the bounds. |  |

## Returning to a Full Screen

From the split screen:

- To display the Numeric Solver full screen, use [2nd [ $\oiint$ ] to make the solver screen active, press F3, and then select 2:Clear Graph View.
- or -
- To display the Home screen, press [2nd [QuIT] twice.


## Clearing Variables Before Leaving the Numeric Solver

When you solve an equation, its variables still exist after you leave the Numeric Solver. If the equation contains single-character variables, their values may inadvertently affect later symbolic calculations. Before leaving the Numeric Solver, you may want to:

1. Press:
[2nd [F6]
to clear all single-character variables in the current folder.
2. Press ENTER to confirm the action.

The screen returns to the solver's eqn: line.
Note: Any time you want to clear single-character variables listed in the solver, use:
[20] [F6].

## Number Bases

## Entering and Converting Number Bases

Regardless of the Base mode, you must always use the appropriate prefix when entering a binary or hexadecimal number.

## Entering a Binary or Hexadecimal Number

To enter a binary number, use the form:


To enter a hexadecimal number, use the form:


Note: You can type the $\mathbf{b}$ or $\mathbf{h}$ in the prefix, as well as hex characters $\mathbf{A}-\mathbf{F}$, in uppercase or lowercase.

If you enter a number without the $\mathbf{0 b}$ or $\mathbf{O h}$ prefix, such as $\mathbf{1 1}$, it is always treated as a decimal number. If you omit the $\mathbf{O h}$ prefix on a hexadecimal number containing $\mathbf{A - F}$, all or part of the entry is treated as a variable.

## Converting between Number Bases

Use the conversion operator

| integerExpression $\downarrow$ Bin | For $\downarrow$ ，press 2nd［ $\downarrow$ ．Also，you can |
| :--- | :--- |
| integerExpression Dec | select base conversions from the |
| integerExpression $\downarrow$ Hex | MATH／Base menu． |

For example，to convert 256 from decimal to binary：

$$
256>\operatorname{Bin}
$$

Note：If your entry is not an integer，a Domain error is displayed．

To convert 101110 from binary to hexadecimal：
Ob101110 • Hex

## Alternate Method for Conversions

Instead of using $>$ ，you can：
1．Use MODE to set the Base mode to the base that you want to convert to．

For a binary or hex entry，you must use the 0 b or Oh prefix．

| －256．Bin | 日61000000000 |  |
| :---: | :---: | :---: |
| －B1011101Hex |  | Bh2E |
| 吅10111口ト下ox |  |  |
|  | FUNIT | $2 / 20$ |

Results use the 0 b or Oh prefix to identify．

If Base mode $=$ BIN：

| － 256 |  | 回1回回 |
| :---: | :---: | :---: |
| 256 |  |  |
| －1／il｜ | Fiñ AUTD | FUW［： |

2. From the Home screen, type the number

If Base mode = HEX: that you want to convert (using the correct prefix) and press ENTER.

| - 保10110 |  |  | 92E |
| :---: | :---: | :---: | :---: |
| 0101110 |  |  |  |
| - Mind | Find | FUNA: | $1 \% 0$ |

## Performing Math Operations with Hex or Bin Numbers

For any operation that uses an integer number, you can enter a hexadecimal or binary number. Results are displayed according to the Base mode. However, results are restricted to certain size limits when Base $=$ HEX or BIN.

## Setting the Base Mode for Displayed Results

1. Press MODE F2 to display Page 2 of the MODE screen.
2. Scroll to the Base mode, press (1), and select the applicable setting.
3. Press ENTER to close the MODE screen.


The Base mode controls the displayed format of integer results only.

Note: The Base mode affects output only. You must always use the $\mathbf{0 h}$ or $\mathbf{0 b}$ prefix to enter a hex or binary number.

Fractional and floating-point results are always shown in decimal form.

## Dividing When Base = HEX or BIN

When Base=HEX or BIN, a division result is displayed in hexadecimal or binary form only if the result is an integer.

To ensure that division always produces an integer, use intDiv( ) instead of $\dot{\square}$.

If Base mode $=$ HEX:


Oh prefix in result identifies the base.

If Base mode $=$ HEX:


Press ENTER to display the result in APPROXIMATE form.

## Size Limitations When Base $=$ HEX or BIN

When Base=HEX or BIN, an integer result is stored internally as a signed, 32-bit binary number, which uses the range (shown in hexadecimal and decimal):


If a result's magnitude is too large to be stored in a signed, 32 -bit binary form, a symmetric modulo operation brings the result into the range. Any number greater than Oh7FFFFFFF is affected. For example, Oh80000000 through OhFFFFFFFF become negative numbers.

## Comparing or Manipulating Bits

The following operators and functions let you compare or manipulate bits in a binary number. You can enter an integer in any number base. Your entries are converted to binary automatically for the bitwise operation, and results are displayed according to the Base mode.Boolean Operations

## Operator with syntax Description

not integer Returns the one's complement, where each bit is flipped.

Returns the two's complement, which is the one's complement +1 .

| Operator with syntax | Description |
| :--- | :--- |
| integer 1 and integer 2 | In a bit-by-bit and comparison, the result is 1 if <br> both bits are 1; otherwise, the result is 0. The <br> returned value represents the bit results. |
| integer1 or integer 2 | In a bit-by-bit or comparison, the result is 1 if <br> either bit is 1; the result is 0 only if both bits are <br> 0. The returned value represents the bit results. |
| integer 1 xor integer 2 | In a bit-by-bit xor comparison, the result is 1 if <br> either bit (but not both) is $1 ;$ the result is 0 if both <br> bits are 0 or both bits are 1. The returned value <br> represents the bit results. |

Note: You can select these operators from the MATH/Base menu. For an example using each operator, refer to the Technical Reference module.

Suppose you enter:

## Oh7AC36 and 0h3D5F

Internally, the hexadecimal integers are converted to a signed, 32-bit binary number.

Then corresponding bits are compared.

If Base mode $=$ HEX:

| - bh7PCS6 and bh305F |  |
| :---: | :---: |
|  | Gh2C16 |
| 鸟730.36 and Gh3d5f |  |
| Mifle kind iuto func |  |

If Base mode $=\mathrm{BIN}$ :


Oh7AC36 = Ob00000000000001111010110000110110
and and
Oh3D5F Ob00000000000000000011110101011111
Ob00000000000000000010110000010110 = Oh2C16
-Leading zeros are not shown in the result.

Note: If you enter an integer that is too large to be stored in a signed, 32-bit binary form, a symmetric modulo operation brings the value into the range.

The result is displayed according to the Base mode.

## Rotating and Shifting Bits

| Function with syntax | Description |
| :---: | :---: |
| ```rotate(integer) - or - rotate(integer,#ofRotations)``` | If \#ofRotations is: <br> omitted - bits rotate once to the right (default is -1 ). <br> - negative - bits rotate the specified number of times to the right. <br> - positive - bits rotate the specified number of times to the left. <br> In a right rotation, the rightmost bit rotates to the leftmost bit; vice versa for a left rotation. |

Function with syntax Description
shift(integer) If \#ofShifts is:

- or -
shift(integer,\#ofShifts)
- omitted - bits shift once to the right (default is -1 ).
- negative - bits shift the specified number of times to the right.
- positive - bits shift the specified number of times to the left.
In a right shift, the rightmost bit is dropped and 0 or 1 is inserted to match the leftmost bit. In a left shift, the leftmost bit is dropped and 0 is inserted as the rightmost bit.

Suppose you enter:

## shift(0h7AC36)

| If Base mode $=\mathrm{HEX}$ : |  |
| :---: | :---: |
| - $=$ hift(9h7ac36) | ansocie |
|  |  |

Internally, the hexadecimal integer is converted to a signed, 32-bit binary number.

Then the shift is applied to the binary number.
If Base mode = BIN:


Each bit shifts to the right.

```
7AC36 = 0b00000000000001111010110000110110
Inserts 0 if leftmost bit is 0, Dropped
or 1 if leftmost bit is }
b00000000000000111101011000011011 = Oh3D61B
    Leading zeros are not shown in the result.
```

The result is displayed according to the Base mode.
Note: If you enter an integer that is too large to be stored in a signed, 32-bit binary form, a symmetric modulo operation brings the value into the range.

## Memory and Variable Management

## Checking and Resetting Memory

The MEMORY screen shows the amount of memory (in bytes) used by all variables in each data type, regardless of whether the variables are stored in RAM or the user data archive. You can also use this screen to reset the memory.

## Displaying the MEMORY Screen

Press 2nd [MEM]. (The numbers on your MEMORY screen may vary from those shown.)


Prgm/Asn: Includes programs written for the Tl-89 Titanium as well as any assembly-language programs you have loaded.
History: Size of history pairs saved in the Home screen's history area.
FlashApp: Size of Flash applications.
RAM free: Free space in RAM.
Flash ROM free: Free space in Flash ROM.
Note: To display the size of individual variables and determine if they are in the user data archive, use the VAR-LINK screen.

To close the screen, press ENTER. To reset the memory, use the following procedure.

## Resetting the Memory

From the MEMORY screen:

## 1. Press F1.

2. Select the applicable item.


| Item | Description |
| :--- | :--- |
| RAM | 1:All RAM: Resetting RAM erases all data and <br> programs from RAM. |
|  | 2:Default: Resets all system variables and modes to |
| their original factory settings. This does not affect any |  |
|  | user-defined variables, functions, or folders. |
| Flash ROM | 1:Archive: Resetting Archive erases all data and <br>  <br>  <br>  <br>  <br>  <br> programs from Flash ROM. <br>  <br>  <br>  <br>  <br>  <br>  <br> 2:Flash Apps: Resetting Flash Apps erases all Flash from Flash ROM. <br> 3:Both: Resetting both erases all data, programs, and <br> Flash applications from Flash ROM. |
| All Memory | Resetting will delete all data, programs, and Flash <br> applications from RAM and Flash ROM. |
|  |  |

Important: To delete individual (instead of all) variables, use VAR-LINK.
3. When prompted for confirmation, press ENTER.

The TI-89 Titanium displays a message when the reset is complete.
Note: To cancel the reset, press ESC instead of ENTER.
4. Press ENTER to acknowledge the message.

## Displaying the VAR-LINK Screen

The VAR-LINK screen lists the variables and folders that are currently defined. After displaying the screen, you can manipulate the variables and/or folders.

## Displaying the VAR-LINK Screen

Press 2nd [VAR-LINK]. By default, the VAR-LINK screen lists all user-defined variables in all folders and with all data types.

(1) Folder names (alphabetically listed)
(2) Shows installed Flash applications
(3 Size in bytes
(4) Data type
© Variable names (alphabetically listed)

| This... | Indicates this... |
| :---: | :--- |
| $\rightarrow$ | Collapsed folder view (to right of folder name). |
| $\checkmark$ | Expanded folder view (to right of folder name). |


| This... | Indicates this... |
| :---: | :--- |
| $\boldsymbol{\nabla}$ | You can scroll for more variables and/or folders (in bottom <br> left corner of screen). |
| $\boldsymbol{\sigma}$ | If selected with F4]. |
| $\mathbf{B}$ | Locked |
| $\boldsymbol{n}$ | Archived |

To scroll through the list:

- Press $\Theta$ or $\Theta$. (Use 2nd $\Theta$ or 2nd $\Theta$ to scroll one page at a time.)
- or -
- Type a letter. If there are any variable names that start with that letter, the cursor moves to highlight the first of those variable names.

Note: Type a letter repeatedly to cycle through the names that start with that letter.

## Variable Types as Listed on VAR-LINK

| Type | Description |
| :--- | :--- |
| ASM | Assembly-language program |
| DATA | Data |
| EXPR | Expression (includes numeric values) |
| FUNC | Function |
| GDB | Graph database |
| LIST | List |
| MAT | Matrix |
| PIC | Picture of a graph |
| PRGM | Program |
| STR | String |
| TEXT | Text Editor session |

Types not listed above are miscellaneous data types used by software applications.

## Closing the VAR-LINK Screen

To close the VAR-LINK screen and return to the current application, use ENTER or ESC as described below.

| Press: | To: |
| :--- | :--- |
| ENTER | Paste the highlighted variable or folder name to the cursor <br> location in the current application. |


| Press: | To: |
| :--- | :--- |
| ESC | Return to the current application without pasting the <br> highlighted name. |
|  |  |

## Manipulating Variables and Folders with VAR-LINK

On the VAR-LINK screen, you can show the contents of a variable. You can also select one or more listed items and manipulate them by using the operations in this section.

## Showing the Contents of a Variable

You can show all variable types except ASM, DATA, GDB, and variables created by Flash Apps. For example, you must open a DATA variable in the Data/Matrix Editor.

1. On VAR-LINK, move the cursor to highlight the variable.
2. Press:
[2] 2nd [F6]

If you highlight a folder, the screen shows
 the number of variables in that folder.
3. To return to VAR-LINK, press any key.

Note: You cannot edit the contents from this screen.

## Selecting Items from the List

For other operations, select one or more variables and/or folders.

| To select: | Do this: |
| :---: | :---: |
| A single variable or folder | Move the cursor to highlight the item, then press (F4). |
| A group of variables or folders | Highlight each item and press F4]. A $\checkmark$ is displayed to the left of each selected item. (If you select a folder, all variables in that folder are selected.) Use F4 to select or deselect an item. |
| All folders and all variables | Press ( © to expand the folder, then press F5 All and select 1:Select All. |
| 再開 | Choosing 3:Select Current selects the last set of items transmitted to your unit during the current |
|  | VAR-LINK session. |
| 5:Collare hll | Choosing 4:Expand All or 5:Collapse All expands or collapses your folders or Flash applications. |

Note: Press either © © or © to toggle between expanded or collapsed view when you have a folder highlighted.

## Folders and Variables

Folders give you a convenient way to manage variables by organizing them into related groups.

The TI-89 Titanium has one built-in folder named MAIN. Unless you create other folders and designate a user-created folder as the current folder, all variables are stored in the

MAIN folder by default. A system variable or a variable with a reserved name can be stored in the MAIN folder only.

## Example of variables that can be stored in MAIN only

Window variables
(xmin, xmax, etc.)
Table setup variables
(TbIStart, $\Delta$ Tbl, etc.)
$\mathrm{Y}=$ Editor functions
( $\mathbf{y} 1(\mathrm{x})$, etc.)

By creating additional folders, you can store independent sets of user-defined variables (including user-defined functions). For example, you can create separate folders for different TI-89 Titanium applications (Math, Text Editor, etc.) or classes. You can store a user-defined variable in any existing folder.

The user-defined variables in one folder are independent of the variables in any other folder. Therefore, folders can store separate sets of variables with the same names but different values.


You cannot create a folder within another folder.
The system variables in the MAIN folder are always directly accessible, regardless of the current folder.

Note: User-defined variables are stored in the "current folder" unless you specify otherwise.

## Creating a Folder from the VAR-LINK Screen

1. Press 2nd [VAR-LINK].
2. Press F1 Manage and select 5:Create Folder.
CREATE NEW FDLDER
3. Type a unique folder name up to eight characters, and press ENTER twice.

After you create a new folder from VAR-LINK, that folder is not automatically set as the current folder.

## Creating a Folder from the Home Screen

Enter the NewFold command on the Home screen.
NewFold folderName
$\square$ Folder name to create. This new folder is set automatically as the current folder.

## Setting the Current Folder from the Home Screen

Enter the setFold function on the Home screen.

```
setFold (folderName)
```

setFold is a function, which requires you to enclose the folder name in parentheses.

When you execute setFold, it returns the name of the folder that was previously set as the current folder.

## Setting the Current Folder from the MODE Dialog Box

1. Press MODE.
2. Highlight the Current Folder setting.
3. Press ( $(1)$ to display a menu of existing folders.

Note: To cancel the menu or exit the
 dialog box without saving any changes, press ESC.
4. Select the applicable folder. Either:

- Highlight the folder name and press ENTER.
- or -
- Press the corresponding number or letter for that folder.

5. Press ENTER to save your changes and close the dialog box.

## Renaming Variables or Folders

Remember, if you use F4 to select a folder, the variables in that folder are selected automatically. As necessary, use F4] to deselect individual variables.

1. On VAR-LINK, select the variables and/or folders.
2. Press F1 Manage and select 3:Rename.
3. Type a unique name, and press ENTER twice.

If you selected multiple items, you are prompted to enter a new name for each
 one.

## Using Variables in Different Folders

You can access a user-defined variable or function that is not in the current folder. Specify the complete pathname instead of only the variable name.

A pathname has the form:
folderName $\backslash$ variableName

- or -
folderName $\backslash$ functionName

For example:

If Current Folder = MAIN


## Folders and Variables

| MAIN |
| :--- |
| $a=1$ |
| $f(x)=x^{3}+x^{2}+x$ |


| - 4 - 3 |  |  |
| :---: | :---: | :---: |
| -4•matha 168 |  |  |
| - f(5) 155 |  |  |
| - mathヤf(5) |  | 120 |
| Matや. ${ }^{\text {ma }}$ |  |  |
| Mill | FUNC | $4 \times 200$ |

$$
\begin{aligned}
& \text { MATH } \\
& \begin{array}{l}
a=42 \\
f(x)=3 x^{2}+4 x+25
\end{array}
\end{aligned}
$$

To see a list of existing folders and variables, press 2nd [VAR-LINK]. On the VAR-LINK screen, you can highlight a variable and press ENTER to paste that variable name to the open application's entry line. If you paste a variable name that is not in the current folder, the pathname (folderNamelvariableName) is pasted.

## Listing Only a Specified Folder and/or Variable Type, or Flash application

If you have a lot of variables, folders, or Flash applications, it may be difficult to locate a particular variable. By changing VAR-LINK's view, you can specify the information you want to see.

From the VAR-LINK screen:

## 1. Press F2 View.

2. Highlight the setting you want to change, and press (1). This displays a menu of
 valid choices. (To cancel a menu, press ESC.)

View - Allows you to choose variables, Flash applications, or system variables to view.


Note: To list system variables (window variables, etc.), select 3:System

Folder - Always lists 1:All and 2:main, but lists other folders only if you have created them.


Var Type - Lists the valid variable types.
$\downarrow$ - indicates that you can scroll for additional variable types.

3. Select the new setting.
4. When you are back on the VAR-LINK VIEW screen, press ENTER.

The VAR-LINK screen is updated to show only the specified folder, variable type, or Flash application.

## Copying or Moving Variables from One Folder to Another

You must have at least one folder other than MAIN. You cannot use VAR-LINK to copy variables within the same folder.

1. On VAR-LINK, select the variables.
2. Press F1 Manage and select 2:Copy or 4:Move.
3. Select the destination folder.

4. Press ENTER. The copied or moved variables retain their original names.

Note: To copy a variable to a different name in the same folder, use STO. (such as $\mathrm{a} 1 \rightarrow \mathrm{a} 2$ ) or the CopyVar command from the Home screen.

## Locking or Unlocking Variables Folders, or Flash Applications

When a variable is locked, you cannot delete, rename, or store to it. However, you can copy, move, or display its contents. When a folder is locked, you can manipulate the variables in the folder (assuming the variables are not locked), but you cannot delete the folder. When a Flash application is locked, you cannot delete it.

1. On VAR-LINK, select the variables, folders, or Flash application.
2. Press F1 Manage and select 6:Lock or 7:UnLock.

E indicates a locked variable or folder in RAM.
a: indicates an archived variable, which is locked automatically.


## Deleting a Folder from the VAR-LINK Screen

When you delete a folder from the VAR-LINK screen, all of the variables in that folder are also deleted. You cannot delete the MAIN folder.

1. Press 2nd [VAR-LINK].
2. Press [F4] to select the folder(s) to delete. (The folder's variables become selected automatically.)

3. Press F1 1:Delete or $\square$.

4. Press ENTER to confirm the deletion of the folder and all its variables.


## Deleting a Variable or a Folder from the Home Screen

Before deleting a folder from the Home screen, you must first delete all the variables stored in that folder.

- To delete a variable, enter the DeIVar command on the calculator Home screen.

DeIVar var1 [, var2] [, var3] ...

- To delete an empty folder, enter the DelFold command on the calculator Home screen.

DelFold folder1 [, folder2] [, folder3] ...
Note: You cannot delete the MAIN folder.

## Pasting a Variable Name to an Application

Suppose you are typing an expression on the Home screen and can't remember which variable to use. You can display the VAR-LINK screen, select a variable from the list, and paste that variable name directly onto the Home screen's entry line.

## Which Applications Can You Use?

From the following applications, you can paste a variable name to the current cursor location.

- Home screen, $\mathrm{Y}=$ Editor, Table Editor, or Data/Matrix Editor - The cursor must be on the entry line.
- Text Editor, Window Editor, Numeric Solver, or Program Editor - The cursor can be anywhere on the screen.

You can also paste a variable name to the current cursor location in many Flash applications.

## Procedure

Starting from an application listed above:

1. Position the cursor where you want to insert the variable name.
2. Press 2nd [VAR-LINK].
3. Highlight the applicable variable.

Note: You can also highlight and paste folder names.
4. Press ENTER to paste the variable name.

Note: This pastes the variable's name, not its contents. Use 2nd [RCL], instead of 2nd [VAR-LINK], to recall a variable's contents.
5. Finish typing the expression.


```
sin(a1|
```

$\square$

## Archiving and Unarchiving a Variable

To archive or unarchive one or more variables interactively, use the VAR-LINK screen. You can also perform these operations from the Home screen or a program.

## Why Would You Want to Archive a Variable?

The user data archive lets you:

- Store data, programs, or any other variables to a safe location where they cannot be edited or deleted inadvertently.
- Create additional free RAM by archiving variables. For example:
- You can archive variables that you need to access but do not need to edit or change, or variables that you are not using currently but need to retain for future use.

Note: You cannot archive variables with reserved names or system variables.

- If you acquire additional programs for your TI-89 Titanium, particularly if they are large, you may need to create additional free RAM before you can install those programs.

Additional free RAM can improve performance times for certain types of calculations.

## From the VAR-LINK Screen

To archive or unarchive:

1. Press 2nd [VAR-LINK] to display the VAR-LINK screen.
2. Select one or more variables, which can be in different folders. (You can select an entire folder by selecting the folder name.)
Note: To select a single variable, highlight it. To select multiple variables, highlight each variable and press F4 $\checkmark$.
3. Press F1 and select either:

8:Archive Variable

- or -

9:Unarchive Variable


If you select 8:Archive Variable, the variables are moved to the user data archive.
mi $=$ archived variables


You can access an archived variable just as you would any locked variable. For all purposes, an archived variable is still in its original folder; it is simply stored in the user data archive instead of RAM.

Note: An archived variable is locked automatically. You can access the variable, but you cannot edit or delete it.

## From the Home Screen or a Program

Use the Archive and Unarchiv commands:

Archive variable1, variable2, ...
Unarchiv variable1, variable2, ...

## If a Garbage Collection Message Is Displayed

If you use the user data archive extensively, you may see a Garbage Collection message. This occurs if you try to archive a variable when there is not enough free archive memory. However, the TI-89 Titanium will attempt to rearrange the archived variables to make additional room.

## Responding to the Garbage Collection Message

When you see the message to the right:

- To continue archiving, press ENTER.

- or -
- To cancel, press ESC.

After garbage collection, depending on how much additional space is freed, the variable may or may not be archived. If not, you can unarchive some variables and try again.

## Why not Perform Garbage Collection Automatically, without a Message?

The message:

- Lets you know why an archive will take longer than usual. It also alerts you that the archive may fail if there is not enough memory.
- Can alert you when a program is caught in a loop that repetitively fills the user data archive. Cancel the archive and investigate the reason.


## Why Is Garbage Collection Necessary?

The user data archive is divided into sectors. When you first begin archiving, variables are stored consecutively in sector 1 . This continues to the end of the sector. If there is not enough space left in the sector, the next variable is stored at the beginning of the next sector. Typically, this leaves an empty block at the end of the previous sector.

Each variable that you archive is stored in the first empty block large enough to hold it.
Note: An archived variable is stored in a continuous block within a single sector; it cannot cross a sector boundary.


This process continues to the end of the last sector. Depending on the size of individual variables, the empty blocks may account for a significant amount of space.

Note: Garbage collection occurs when the variable you are archiving is larger than any empty block.

## How Unarchiving a Variable Affects the Process

When you unarchive a variable, it is copied to RAM but it is not actually deleted from user data archive memory.

After you unarchive variables B and C, they continue to take up space.


Unarchived variables are "marked for deletion," meaning they will be deleted during the next garbage collection.

## If the MEMORY Screen Shows Enough Free Space

Even if the MEMORY screen shows enough free space to archive a variable, you may still get a Garbage Collection message.

This TI-89 Titanium memory screen shows free space that will be available after all "marked for deletion" variables are deleted.

When you unarchive a variable, the Flash
 ROM free amount increases immediately, but the space is not actually available until after the next garbage collection.

## The Garbage Collection Process

The garbage collection process:

- Deletes unarchived variables from the user data archive.
- Rearranges the remaining variables into consecutive blocks.



## Memory Error When Accessing an Archived Variable

An archived variable is treated the same as a locked variable. You can access the variable, but you cannot edit or delete it. In some cases, however, you may get a Memory Error when you try to access an archived variable.

## What Causes the Memory Error?

The Memory Error message is displayed if there is not enough free RAM to access the archived variable. This may cause you to ask, "If the variable is in the user data archive, why does it matter how much RAM is available?" The answer is that the following operations can be performed only if a variable is in RAM.

- Opening a text variable in the Text Editor.
- Opening a data variable, list, or matrix in the Data/Matrix Editor.
- Opening a program or function in the Program Editor.
- Running a program or referring to a function.

Note: A temporary copy lets you open or execute an archived variable. However, you cannot save any changes to the variable.

So that you don't have to unarchive variables unnecessarily, the TI-89 Titanium performs a "behind-the scenes" copy. For example, if you run a program that is in the user data archive, the TI-89 Titanium:

1. Copies the program to RAM.
2. Runs the program.
3. Deletes the copy from RAM when the program is finished.

The error message is displayed if there is not enough free RAM for the temporary copy.
Note: Except for programs and functions, referring to an archived variable does not copy it. If variable $a b$ is archived, it is not copied if you perform $6 * \mathbf{a b}$.

## Correcting the Error

To free up enough RAM to access the variable:

1. Use the VAR-LINK screen ( 2 nd $[$ VAR-LINK]) to determine the size of the archived variable that you want to access.
2. Use the MEMORY screen ([2nd [MEM]) to check the RAM free size.
3. Free up the needed amount of memory by:

- Deleting unnecessary variables from RAM.
- Archiving large variables or programs (moving them from RAM to the user data archive).
Note: Typically, the RAM free size must be larger than the archived variable.


## Connectivity

## Connecting Two Units

The TI-89 Titanium comes with a cable that lets you connect two units. Once connected, you can transmit information between two units. A USB unit-to-unit cable is included with the $\mathrm{TI}-89$ Titanium; use the calculator's USB port with this cable.

Note: The TI-89 Titanium features both a USB port and an I/O port, so you can connect TI graphing calculators with either type of link port. However, using the I/O port requires the I/O unit-to-unit cable (sold separately) or the USB Silver Edition cable (also sold separately), which is used to connect to a computer.

## Connecting before Sending or Receiving

Using firm pressure, insert one end of the cable into the link port of each unit. Either unit can send or receive, depending on how you set them up from the VAR-LINK screen.

You can link a TI-89 Titanium or Voyage ${ }^{\text {TM }} 200$ to another TI-89 Titanium, Voyage ${ }^{\text {TM }}$ 200, TI-89, or TI-92 Plus.


Two TI-89 Titanium calculators linked together


Position so that the USB symbols face each other; then insert the connector.


A TI-89 Titanium and a Voyage ${ }^{\text {TM }} 200$ linked together


A TI-89 Titanium and a TI-89 linked together

## Transmitting Variables, Flash Applications, and Folders

Transmitting variables is a convenient way to share any variable listed on the VAR-LINK screen - functions, programs, etc. You can also transmit Flash applications (Apps) and folders.

## Setting Up the Units

Flash applications will transfer only between certain units. For example, you can transfer an App from a Tl-89 Titanium to another TI-89 Titanium, or from a Tl-89 Titanium to a TI89.

During transmission, a progress bar is displayed in the status line of the receiving unit. When transmission is complete, the VAR-LINK screen is updated on the receiving unit.

Note: Before transferring a purchased App, the receiving unit must have the appropriate certificate, if required. A certificate is a file that is generated by TI. Free and concept Apps do not require a certificate.

## Rules for Transmitting Variables, Flash Applications, or Folders

Unlocked and unarchived variables that have the same name on both the sending and receiving units will be overwritten from the sending unit.

Locked variables that have the same name on both the sending and receiving units must be unlocked on the receiving unit before they can be overwritten from the sending unit. If archived variables have the same names on both the sending and receiving units, a message asks you to confirm that you will allow the variables to be overwritten.

| If you select: | What happens: |
| :--- | :--- |
| Unlocked variable | The variable is transmitted to the current folder and <br> it remains unlocked on the receiving unit. |
| Locked variable | The variable is transmitted to the current folder and <br> it remains locked on the receiving unit. |


| If you select: | What happens: |
| :--- | :--- |
| Archived variable | The variable is transmitted to the current folder and <br> it remains archived on the receiving unit. |
| Unlocked Flash <br> application | If the receiving unit has the correct certification, the <br> Flash application is transmitted. It remains <br> unlocked on the receiving unit. |
| Locked Flash <br> application | If the receiving unit has the correct certification, the <br> Flash application is transmitted. It remains locked <br> on the receiving unit. |
| Unlocked Folder | The folder and its selected contents are <br> transmitted. The folder remains unlocked on the <br> receiving unit. |
| Locked Folder | The folder and its selected contents are <br> transmitted. The folder becomes unlocked on the <br> receiving unit. |

## Canceling a Transmission

From either the sending or receiving unit:

1. Press 0 N .

An error message is displayed.
2. Press ESC or ENTER.


## Common Error and Notification Messages

| Shown on: | Message and Description: |
| :---: | :---: |
| Sending unit | This is displayed after several seconds if: <br> - A cable is not attached to the sending unit's link port. - or - <br> - A receiving unit is not attached to the other end of the cable. <br> - or - <br> - The receiving unit is not set up to receive. <br> Press ESC or ENTER to cancel the transmission. <br> Note: The sending unit may not always display this message. Instead, it may remain BUSY until you cancel the transmission. |
| Sending unit | The receiving unit does not have the correct certification for the operating system (OS) or Flash application being sent. |


| Shown on: | Message and Description: |
| :---: | :---: |
| Receiving unit | New Name is active only if you change Overwrite to NO <br> The receiving unit has a variable with the same name as the specified variable being sent. <br> - To overwrite the existing variable, press ENTER. (By default, Overwrite = YES. ) <br> - To store the variable to a different name, set Overwrite = NO. In the New Name input box, type a variable name that does not exist in the receiving unit. Then press ENTER twice. <br> - To skip this variable and continue with the next one, set Overwrite = SKIP and press ENTER. <br> - To cancel the transmission, press ESC. |
| Receiving unit | The receiving unit does not have enough memory for what is being sent. Press ESC or ENTER to cancel the transmission. |

## Deleting Variables, Flash Applications, or Folders

1. Press 2nd [VAR-LINK] to display the VAR-LINK screen.
2. Select the variables, folders, or Flash applications to delete.

- To select a single variable, Flash application, or folder, move the cursor to highlight it and press F4 to place a checkmark ( $\checkmark$ ) beside it.
- If on the default VAR-LINK screen, this selects the folder and its contents. Collapsed folders become expanded when selected.
- If selecting a Flash App (from the F7 tab), this selects the App folder and its contents. A checkmark appears beside the folder, but not beside the contents. Collapsed Flash App folders do not automatically become expanded.

Note: You cannot delete the Main folder.

- To select multiple variables, Flash applications, or folders highlight each one and press F4 to place a checkmark ( $\checkmark$ ) beside it. Use F4 again to deselect any that you do not want to transmit.
- To select all variables, Flash applications, or folders use F5 All 1:Select All.

3. Press F1 and choose 1:Delete.

- or -

Press $\square$. A confirmation message appears.
4. Press ENTER to confirm the deletion.

## Where to Get Flash Applications (Apps)

For up-to-date information about available Flash applications, check the Texas Instruments Web site at education.ti.com.

Many Apps no longer require a certificate. If you try to transfer an App from one unit to another and receive an Unlicensed OS or Flash application message, try downloading the App again from the Texas Instruments Web site at education.ti.com.

You can download a Flash application and/or certificate from the Texas Instruments Web site to a computer, and use a to install the application or certificate on your TI-89 Titanium.

For Flash App installation instructions, see education.ti.com/guides.

## Transmitting Variables under Program Control

You can use a program containing GetCalc and SendCalc to transmit a variable from one device to another.

SendCalc sends a variable to the link port, where a linked device can receive the variable. The linked device must be on the Home screen or must execute GetCalc from a program.

You can use optional parameters with the SendCalc or GetCalc command to specify either the USB port or I/O port. (See Appendix A for details.) If you do not include these parameters, the TI-89 Titanium communicates through the USB port.

## The "Chat" Program

The following program uses GetCalc and SendCalc. The program sets up two loops that let the linked devices take turns sending and receiving/displaying a variable named msg. InputStr lets each user enter a message in the msg variable

```
    :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
(1) GetCalc msg
                                Disp msg
                                    InputStr msg
                                    SendCalc msg
            EndWhile
            :Else
            : While true
4
                                    InputStr msg
                                    SendCalc msg
                                    (3
2
    \square
```



6

```
5 : : GetCalc msg
: Disp msg
: EndWhile
: EndIf
:EndPrgm
```


## Notes:

(1) Sets up this unit to receive and display the variable msg.
(2) Then lets this user enter a message in msg and send it.
(3) Loop executed by the unit that receives the first message.
(4) Lets this user enter a message in msg and send it.
(5) Then sets up this unit to receive and display msg.
(6) Loop executed by the unit that sends the first message.

To synchronize GetCalc and SendCalc, the loops are arranged so that the receiving unit executes GetCalc while the sending unit is waiting for the user to enter a message.

## Running the Program

This procedure assumes that:

- The two devices are linked with the connecting cable.
- The Chat program is loaded on both devices.
- Use each device's Program Editor to enter the program.
- or -
- Enter the program on one device and then use VAR-LINK to transmit the program variable to the other device.

To run the program on both devices:

1. On the Home screen of each device, enter chat( ).
2. When each device displays its initial prompt, respond as shown below.
On the: Type:

Device that will send the first 1 and press ENTER. message.

Device that will receive the first $\mathbf{0}$ and press ENTER. message.
3. Take turns typing a message and pressing ENTER to send the variable msg to the other device.

## Stopping the Program

Because the Chat program sets up an infinite loop on both devices, press 0 ON (on both devices) to break the program. If you press ESC to acknowledge the error message, the program stops on the Program I/O screen. Press F5 or ESC to return to the Home screen.

## Upgrading the Operating System (OS)

You can upgrade the OS on your TI-89 Titanium using your computer. You can also transfer the OS from one unit to another identical model (for example, from a TI-89 Titanium to a TI-89 Titanium or from a Voyage ${ }^{\text {TM }} 200$ to a Voyage ${ }^{\text {TM }} 200$ ).

Installing OS software resets all device memory to the original factory settings. This means that all user-defined variables (in both RAM and the user data archive), functions, programs, lists, and folders (except the Main folder) will be deleted. It is possible that Flash applications could also be deleted. You should use TI Connect software to back up your data to your computer before installing a new OS on your calculator.

See the important information concerning batteries before performing an OS upgrade.

## Important Operating System Download Information

New batteries should be installed before beginning an OS download.
When in OS download mode, the Automatic Power Down ${ }^{\text {TM }}$ (APD ${ }^{\text {TM }}$ ) feature does not function. If you leave your device in download mode for an extended time before you actually start the downloading process, your batteries may become depleted. You will then need to replace the depleted batteries with new batteries before downloading.

If you accidentally interrupt the transfer before it is complete, you will need to reinstall the OS. Again, remember to install new batteries before downloading.

## Backing Up Your Unit Before an Operating System Installation

When you install an OS upgrade, the installation process:

- Deletes all user-defined variables (in both RAM and the user data archive), functions, programs, and folders.
- Could delete all Flash applications.
- Resets all system variables and modes to their original factory settings. This is equivalent to using the MEMORY screen to reset all memory.

To retain any existing variables or Flash applications, do the following before installing the upgrade:

- Important: Install new batteries.
- Transmit the variables or Flash applications to another device.
- or -
- Use a USB cable or TI Connectivity Cable USB and TI Connect ${ }^{\text {TM }}$ software (education.ti.com/downloadticonnect) to send the variables and/or Flash applications to a computer.


## Where to Get Operating System Upgrades

For up-to-date information about available OS upgrades, check the Texas Instruments Web site at education.ti.com/downloadticonnect.

You can download an OS upgrade or Flash application from the Texas Instruments Web site to a computer, and use a USB computer cable to install the OS or application on your TI-89 Titanium.

For complete information, refer to the instructions on the web.

## Transferring the Operating System

OS software will transfer only from a TI-89 Titanium to a TI-89 Titanium, TI-89 to a TI-89, from a Voyage ${ }^{\text {TM }} 200$ to a Voyage ${ }^{\text {TM }} 200$, or from a TI-92 Plus to a Tl-92 Plus.

To transfer the Operating System (OS) from unit to unit:

1. Link two like units together, for example, a TI-89 Titanium to a TI-89 Titanium; or a Voyage ${ }^{\text {TM }} 200$ to a Voyage ${ }^{\text {TM }} 200$.
2. On the receiving and the sending unit, press [2nd [VAR-LINK] to display the VAR-LINK screen.
3. On the receiving and the sending unit, press F3 Link to display the menu options.
4. On the receiving unit, select 5 :Receive OS.

A warning message displays. Press ESC to halt the process, or press ENTER to proceed. Pressing ENTER, displays VAR-LINK: WAITING TO RECEIVE and BUSY in the status line of the receiving unit.
5. On the sending unit, select 4:Send OS.

A warning message displays. Press ESC to halt the process, or press ENTER to start the transmission.

## Important:

- For each receiving unit, remember to back up information as necessary and install new batteries.
- Be sure both the sending and receiving units are in the VAR-LINK screen.

During the transfer, the receiving unit shows how the transfer is progressing. When the transfer is complete:

- The sending unit returns to the VAR-LINK screen.
- The receiving unit returns to either the Apps desktop or the Home screen. You may need to use $\square \square$ (lighten) or $\downarrow$ (darken) to adjust the contrast.


## Do Not Attempt to Cancel an Operating System Transfer

After the transfer starts, the receiving unit's existing OS is effectively deleted. If you interrupt the transfer before it is complete, the receiving unit will not operate properly. You will then need to reinstall the OS upgrade.

## If You are Upgrading the Operating System on Multiple Units

To perform an OS upgrade on multiple units, download and install the OS into one unit and then transfer the OS upgrade from one unit to another. This method is faster than installing it on each unit via a computer. OS upgrades are released free of charge and you do not need to obtain a certificate before you download or install them.

## Error Messages

Most error messages are displayed on the sending unit. Depending on when the error occurs during the transfer process, you may see an error message on the receiving unit.

| Error Message | Description |
| :---: | :---: |
|  | The sending and receiving units are not connected properly, or the receiving unit is not set up to receive. |
|  | The certificate on the receiving unit is not valid for the operating system (OS) or App on the sending unit. You must obtain and install a valid certificate. <br> If the App no longer requires a certificate, you can download it again from the Texas Instruments Web site at education.ti.com and then install the App again on your calculator. |
|  | An error occurred during the transfer. The current OS in the receiving unit is corrupted. You must reinstall the product software from a computer. |
|  | Replace the batteries on the unit displaying this message. |

## Collecting and Transmitting ID Lists

The VAR-LINK screen F3] 6:Send ID List menu option allows collection of electronic ID numbers from individual TI-89 Titanium, Tl-89, Voyage ${ }^{\text {TM }} 200$, or TI-92 Plus devices.

## ID Lists and Group Certificates

The ID list feature provides a convenient way to collect device IDs for group purchase of commercial applications. After the IDs are collected, transmit them to Texas Instruments so a group certificate can be issued.

A group certificate allows distribution of purchased software to multiple TI-89 Titanium, TI-89, Voyage ${ }^{\text {TM }} 200$, or Tl-92 Plus units. The software can be loaded, deleted from, and reloaded to the devices as often as needed for as long as the software remains listed in the group certificate. You may add new ID numbers and/or new commercial applications to a group certificate.

## Collecting ID Lists

You can use one device to collect all of the IDs, or use several collection units and then consolidate their ID lists onto one device.

To send an ID number from one device to another, first connect two units by using a USB unit-to-unit cable or I/O unit-to-unit cable.

| Step: | On the: | Do this: |
| :---: | :---: | :---: |
| 1. | Collecting unit (Receiving unit) | $\begin{aligned} & \text { Display the Home screen. Press: } \\ & \text { [ HOME } \\ & \text { [CALC HOME] } \end{aligned}$ |
| 2. | Sending unit | a. Press 2nd [VAR-LINK] to display the VAR-LINK screen. |
|  |  | b. Press F3 Link and select 6:Send ID List. |
|  |  | The sending unit adds a copy of its unique ID number to the collection unit's ID list. The sending unit always retains its own ID number, which cannot be deleted from the device. |
| 3. | Additional units | Repeat steps 1 and 2 until all the IDs are collected onto one device. <br> Depending on available memory in the collection device, it is possible to collect over 4,000 IDs. |

## Notes:

- You cannot view the ID list on the sending or collecting units.
- Each time an ID list is successfully sent from one device to another, the ID list is automatically deleted from the sending unit.
- If an ID is collected from a device twice, the duplicate ID is automatically deleted from the list.


## Clearing the ID List

The ID list remains on the collection device after it is uploaded to the computer. You can then use the collection device to upload the list to other computers.

To clear the ID list from the collection unit:

1. Press 2nd [VAR-LINK] to display the VAR-LINK screen.
2. Press F1 Manage and select A:Clear ID List.


## Compatibility among the TI-89 Titanium, Voyage ${ }^{\text {TM }}$ 200, TI-89, and TI-92 Plus

In general, TI-89 Titanium, TI-89, Voyage ${ }^{\text {TM }} 200$, and TI-92 Plus data and programs are compatible with each other, with a few exceptions.

Most functions of the TI-89 Titanium are compatible with the TI-89, Voyage ${ }^{\text {TM }} 200$, and TI-92 Plus. The TI-89 Titanium and the TI-89 are similar, except that the TI-89 Titanium has more memory (more room for Apps and user archive) and the TI-89 Titanium has a USB port. The Voyage ${ }^{\text {TM }} 200$ is the same as the TI-92 Plus except it has more memory, and thus more room for applications (Apps).

All data is compatible among the TI-89 Titanium, TI-89, Voyage ${ }^{\text {TM }} 200$, and TI-92 Plus, but some programs written for one may not run or may not run the same on the other because of differences in the device's screen sizes and keyboards and the USB port on the TI-89 Titanium.

Other incompatibilites can occur because of different version the operating system. To download the latest version of the operating system, visit the Texas Instruments Web site at education.ti.com/downloadticonnect.

## Link Transmission Table

| $\text { To } \rightarrow$ <br> From $\downarrow$ | $\begin{aligned} & \hline \text { TI-89 } \\ & \text { Titanium } \end{aligned}$ | TI-89 | $\begin{aligned} & \text { Voyage }^{\mathrm{TM}} \\ & 200 \end{aligned}$ | TI-92 Plus |
| :---: | :---: | :---: | :---: | :---: |
| $\begin{aligned} & \hline \text { TI-89 } \\ & \text { Titanium } \end{aligned}$ | OS <br> Apps <br> Variables | Apps <br> Variables | Variables | Variables |
| TI-89 | Apps Variables | OS <br> Apps <br> Variables | Variables | Variables |
| $\begin{aligned} & \text { Voyage }^{\mathrm{TM}} \\ & 200 \end{aligned}$ | Variables | Variables | OS <br> Apps <br> Variables | Apps <br> Variables |
| TI-92 Plus | Variables | Variables | Apps <br> Variables | OS <br> Apps <br> Variables |

## Activities

## Analyzing the Pole-Corner Problem

A ten-foot-wide hallway meets a five-foot-wide hallway in the corner of a building. Find the maximum length pole that can be moved around the corner without tilting the pole.

## Maximum Length of Pole in Hallway

The maximum length of a pole $\mathbf{c}$ is the shortest line segment touching the interior corner and opposite sides of the two hallways as shown in the diagram below.

Use proportional sides and the Pythagorean theorem to find the length $\mathbf{c}$ with respect to $\mathbf{w}$. Then find the zeros of the first derivative of $\mathbf{c}(\mathbf{w})$. The minimum value of $\mathbf{c}(\mathbf{w})$ is the maximum length of the pole.


1. Define the expression for side a in terms of $w$ and store it in $a(w)$.

- Lefine $\exists(w)=w+5 \quad$ Done Define $\quad$ (w) $=$ い+5

Note: When you want to define a function, use multiple character names as you build the definition.

2. Define the expression for side $\mathbf{b}$ in terms of $\mathbf{w}$ and store it in $\mathbf{b}(\mathbf{w})$.

| Define $\exists(\mathrm{w})=\mathrm{w}+5$ <br> Lefine $b(w)=\frac{10 \cdot g(w)}{w}$ | Done |
| :---: | :---: |
|  | Done |
| Define b(w) = 10*, (w)/ |  |
| Mîlk Fifl ilta FIME | $2{ }^{2} 0$ |

3. Define the expression for side $\mathbf{c}$ in terms of $\mathbf{w}$ and store it in $\mathbf{c}(\mathbf{w})$.
Enter: Define $\left.\mathbf{c}(\mathbf{w})=\sqrt{( } \mathbf{a}(\mathbf{w})^{\wedge} \mathbf{2 + b}(\mathbf{w})^{\boldsymbol{\wedge}} \mathbf{2}\right)$

4. Use the zeros() function to compute the zeros of the first derivative of $\mathrm{c}(\mathrm{w})$ to find the minimum value of $c(w)$.
Note: The maximum length of the pole is the minimum value of $\mathbf{c}(\mathbf{w})$.
5. Compute the exact maximum length of the pole.
Enter: c (2nd [ANS])
6. Compute the approximate maximum length of the pole.
Result: Approximately 20.8097 feet.
Note: Use the auto-paste feature to copy the result from step 4 to the entry line inside the parentheses of c() and press

- EnTER.




## Deriving the Quadratic Formula

This activity shows you how to derive the quadratic formula:

$$
\mathrm{x}=\frac{-b \pm \sqrt{b^{2}-4 a c}}{2 a}
$$

Detailed information about using the functions in this example can be found in Symbolic Manipulation.

## Performing Computations to Derive the Quadratic Formula

Perform the following steps to derive the quadratic formula by completing the square of the generalized quadratic equation.

1. Clear all one-character variables in the current folder.

## [ 2nd [F6]

Choose 1:Clear a-z and press ENTER to confirm.
2. On the Home screen, enter the generalized quadratic equation: $a x^{2}+b x+c=0$.


Note: This example uses the result of the last answer to perform computations on the $\mathrm{TI}-89$ Titanium. This feature reduces keystroking and chances for error.
4. Divide both sides of the equation by the leading coefficient a.
Note: Continue to use the last answer (2nd [ANS]) as in step 3 in steps 4 through 9.
5. Use the expand( ) function to expand the result of the last answer.


6. Complete the square by adding ((b/a)/2) ${ }^{2}$ to both sides of the equation.
8. Multiply both sides of the equation by $4 a^{2}$.
9. Take the square root of both sides of the equation with the constraint that $\mathbf{a}>0$ and $b>0$ and $x>0$.

| $(2 \cdot a \cdot x+b)^{2}=-\left(4 \cdot a \cdot c-b^{2}\right)$ |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: |
| - $\sqrt{(2 \cdot a \cdot x+b)^{2}=-\left(4 \cdot a \cdot c-b^{2}\right.}$ |  |  |  |  |
| $2 \cdot a \cdot x+b=\sqrt{b^{2}-4 \cdot a \cdot c}$ |  |  |  |  |
|  |  |  |  |  |
| Hilled | Eifl illa | FUNAC |  | 浚 |

10. Solve for $\mathbf{x}$ by subtracting $\mathbf{b}$ from both sides and then dividing by $\mathbf{2 a}$.

Note: This is only one of the two general quadratic solutions due to the constraint in step 9.


## Exploring a Matrix

This activity shows you how to perform several matrix operations.

## Exploring a $\mathbf{3 x} \mathbf{3}$ Matrix

Perform these steps to generate a random matrix, augment and find the identity matrix, and then solve to find an invalid value of the inverse.

1. On the Home screen, use RandSeed to set the random number generator seed to the factory default, and then use randMat( ) to create a random $3 \times 3$ matrix and store it in a.
2. Replace the $[2,3]$ element of the matrix with the variable $\mathbf{x}$, and then use the augment( ) function, to augment the $3 \times 3$ identity to $\mathbf{a}$ and store the result in $\mathbf{b}$.

3. Use rref( ) to "row reduce" matrix b:

The result will have the identity matrix in the first three columns and $\mathbf{a}^{\wedge}-1$ in the last three columns.


Note: Use the cursor in the history area to scroll the result.
4. Solve for the value of $\mathbf{x}$ that will cause the inverse of the matrix to be invalid.

Enter:
solve(getDenom( 2nd [ANS] [1,4] )=0,x)
Result: $\mathbf{x = - 7 0 / 1 7}$


Note: Use the cursor in the history area to scroll the result.

## Exploring $\boldsymbol{\operatorname { c o s }}(\mathrm{x})=\boldsymbol{\operatorname { s i n }}(\mathrm{x})$

This activity uses two methods to find where $\boldsymbol{\operatorname { c o s }}(\mathbf{x})=\boldsymbol{\operatorname { s i n }}(\mathbf{x})$ for the values of $\mathbf{x}$ between 0 and $3 \pi$.

## Method 1: Graph Plot

Perform the following steps to observe where the graphs of the functions $\mathbf{y 1}(\mathbf{x})=\boldsymbol{\operatorname { c o s }}(\mathrm{x})$ and $\mathbf{y 2}(\mathbf{x})=\boldsymbol{\operatorname { s i n }}(\mathbf{x})$ intersect.

1. In the $Y=$ Editor, set $\mathbf{y} 1(x)=\cos (x)$ and $2(x)=\sin (x)$.
2. In the Window Editor, set $\mathbf{x m i n}=0$ and $\mathbf{x m a x}=3 \pi$.
3. Press F2 and select A:ZoomFit.
4. Find the intersection point of the two functions.

Note: Press F5 and select 5:Intersection. Respond to the screen prompts to select the two curves, and the lower and upper

5. Note the $\mathbf{x}$ and $\mathbf{y}$ coordinates. (Repeat steps 4 and 5 to find the other intersections.)

## Method 2: Symbolic Manipulation

Perform the following steps to solve the equation $\boldsymbol{\operatorname { s i n }}(\mathbf{x})=\boldsymbol{\operatorname { c o s }}(\mathbf{x})$ with respect to $\mathbf{x}$.

1. On the Home screen, enter solve( $\boldsymbol{\operatorname { s i n }}(\mathbf{x})=$ $\boldsymbol{\operatorname { c o s }}(\mathrm{x}), \mathbf{x})$.
The solution for x is where @ $\mathbf{n} \mathbf{1}$ is any integer.
2. Using the ceiling() and floor() functions, find the ceiling and floor values for the intersection points as shown.
Note: Move the cursor into the history
 area to highlight the last answer. Press ENTER to copy the result of the general solution.
3. Enter the general solution for x and apply the constraint for @n1 as shown.
Compare the result with Method 1.
Note: To get the with operator:


## Finding Minimum Surface Area of a Parallelepiped

This activity shows you how to find the minimum surface area of a parallelepiped having a constant volume $\mathbf{V}$. Detailed information about the steps used in this example can be found in Symbolic Manipulation and 3D Graphing.

## Exploring a 3D Graph of the Surface Area of a Parallelepiped

Perform the following steps to define a function for the surface area of a parallelepiped, draw a 3D graph, and use the Trace tool to find a point close to the minimum surface area.

1. On the Home screen, define the function $\mathbf{s a}(\mathbf{x}, \mathbf{y}, \mathbf{v})$ for the surface area of a parallelepiped.

Enter: define $\mathbf{s a}(\mathbf{x}, \mathbf{y}, \mathbf{v})=\mathbf{2} * \mathbf{x} * \mathbf{y}+\mathbf{2 v} / \mathbf{x}+\mathbf{2 v} / \mathbf{y}$
2. Select the 3D Graph mode. Then enter the function for $\mathbf{z 1}(\mathbf{x}, \mathbf{y})$ as shown in this example with volume $\mathbf{v}=\mathbf{3 0 0}$.

3. Set the Window variables to:
eye= $[60,90,0]$
$\mathrm{x}=\quad[0,15,15]$
$y=\quad[0,15,15]$
$z=[260,300]$
ncontour= [5]
4. Graph the function and use Trace to go to the point close to the minimum value of the surface area function.


## Finding the Minimum Surface Area Analytically

Perform the following steps to solve the problem analytically on the Home screen.

1. Solve for $\mathbf{x}$ and $\mathbf{y}$ in terms of $\mathbf{v}$.

Enter: solve( $\mathbf{d}(\mathbf{s a}(\mathbf{x}, \mathbf{y}, \mathbf{v}), \mathbf{x})=\mathbf{0}$ and $\mathbf{d}(\mathbf{s a}(\mathbf{x}, \mathrm{y}, \mathrm{v}), \mathrm{y})=\mathbf{0},\{\mathrm{x}, \mathrm{y}\})$
2. Find the minimum surface area when the value of $v$ equals 300 .

Enter: $\mathbf{3 0 0} \rightarrow \mathbf{v}$
Enter: sa(v^(1/3), $\left.\mathbf{v}^{\wedge}(1 / 3), \mathbf{v}\right)$


Note: Press ENTER to obtain the exact result in symbolic form. Press $\square$ ENTER to obtain the approximate result in decimal form.

## Running a Tutorial Script Using the Text Editor

This activity shows you how to use the Text Editor to run a tutorial script.

## Running a Tutorial Script

Perform the following steps to write a script using the Text Editor, test each line, and observe the results in the history area on the Home screen.

1. Open the Text Editor, and create a new variable named demo1.


Note: The command symbol $C$ is accessed from the F2 1:Command toolbar menu.
2. Type the following lines into the Text Editor.
: Compute the maximum value of $f$ on the closed interval $[a, b]$
: assume that $f$ is differentiable on $[\mathrm{a}, \mathrm{b}]$
C : define $f(x)=x^{\wedge} 3-2 x^{\wedge} 2+x-7$
C : $1 \rightarrow a: 3.22 \rightarrow b$
C: $\mathrm{d}(\mathrm{f}(\mathrm{x}), \mathrm{x}) \rightarrow \mathrm{df}(\mathrm{x})$
C: zeros $(\mathrm{df}(\mathrm{x}), \mathrm{x})$
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. Press F3 and select 1:Script view to show the Text Editor and the Home screen on a split-screen. Move the cursor to the first line in the Text Editor.

4. Press F4 repeatedly to execute each line in the script one at a time.

Note: Press F4 and select 2:Clear split to go back to a full-sized Text Editor screen.

5. To see the results of the script on a fullsized screen, go to the Home screen.

Note: Press 2nd [QUIT] twice to display the Home screen.


## Decomposing a Rational Function

This activity examines what happens when a rational function is decomposed into a quotient and remainder. Detailed information about the steps used in this example can be found in Basic Function Graphing and Symbolic Manipulation.

## Decomposing a Rational Function

To examine the decomposition of the rational function $f(x)=\left(x^{3}-10 x^{2}-x+50\right) /(x-2)$ on a graph:

1. On the Home screen, enter the rational function as shown below and store it in a function $f(x)$.


Enter: $\left(x^{\wedge} 3-10 x^{\wedge} 2-x+50\right) /(x-2) \rightarrow f(x)$
Note: Actual entries are displayed in reverse type in the example screens.
2. Use the proper fraction function (propFrac) to split the function into a quotient and remainder.
3. Copy the last answer to the entry line.
-or-
Enter: 16/(x-2)+x^2-8*x-17


Note: Move the cursor into the history area to highlight the last answer. Press ENTER to copy it to the entry line.
4. Edit the last answer in the entry line. Store the remainder to $\mathbf{y 1}(\mathbf{x})$ and the quotient to $\mathbf{y 2}(\mathbf{x})$ as shown.


Enter: $\mathbf{1 6 / ( x - 2 ) \rightarrow y 1 ( x ) : ~} \mathbf{x}^{\wedge} \mathbf{2 - 8 *} \mathbf{x - 1 7 \rightarrow y 2 ( x )}$
5. In the $\mathbf{Y}=$ Editor, select the thick graphing style for $\mathbf{y 2} \mathbf{2 ( x )}$.
6. Add the original function $f(x)$ to $\mathbf{y} 3(\mathbf{x})$ and select the square graphing style.
7. In the Window Editor, set the window variables to:
$\mathrm{x}=$ [-10,15,10]
$y=[-100,100,10]$
8. Draw the graph.

Note: Be sure the Graph mode is set to Function.


Observe that the global behavior of the $f(x)$ function is basically represented by the quadratic quotient $\mathbf{y 2} \mathbf{2} \mathbf{x}$ ). The rational expression is basically a quadratic function as $\mathbf{x}$ gets very large in both the positive and negative directions.

The lower graph is $\mathbf{y} \mathbf{3}(\mathbf{x})=\mathbf{f}(\mathbf{x})$ graphed separately using the line style.


## Studying Statistics: Filtering Data by Categories

This activity provides a statistical study of the weights of high school students using categories to filter the data.

## Filtering Data by Categories

Each student is placed into one of eight categories depending on the student's sex and academic year (freshman, sophomore, junior, or senior). The data (weight in pounds) and respective categories are entered in the Data/Matrix Editor.

| Table 1: Category vs. Description |  |
| :---: | :--- |
| Category (C2) | Academic Year and Sex |
| 1 | Freshman boys |
| 2 | Freshman girls |
| 3 | Sophomore boys |
| 4 | Sophomore girls |
| 5 | Junior boys |
| 6 | Junior girls |
| 7 | Senior boys |
| 8 | Senior girls |

Table 2: C1 (weight of each student in pounds) vs. C2 (category)

| 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 |

Perform the following steps to compare the weight of high school students to their year in school.

1. Start the Data/Matrix Editor, and create a new Data variable named students.

2. Enter the data and categories from Table 2 into columns c1 and c2, respectively.

3. Open the F2 Plot Setup toolbar menu. Note: Set up several box plots to compare different subsets of the entire data set.

4. Define the plot and filter parameters for Plot 1 as shown in this screen.

5. Copy Plot 1 to Plot 2.

6. Repeat step 5 and copy Plot 1 to Plot 3, Plot 4, and Plot 5.
7. Press F1, and modify the Include Categories item for Plot 2 through Plot 5 to the following:

Plot 2: \{1,2\}
(freshman boys, girls)
Plot 3: $\{7,8\}$
(senior boys, girls)
Plot 4: $\{1,3,5,7\}$
(all boys)
Plot 5: \{2,4,6,8\}
(all girls)
8. In the $\mathbf{Y}=$ Editor, deselect any functions that may be selected from a previous activity.

Note: Only Plot 1 through Plot 5 should be selected.

9. Display the plots by pressing F2 and selecting 9:Zoomdata.

10. Use the Trace tool to compare the median student weights for different subsets.
(1) median, all students
(2) all students
(3) all freshmen


4 all seniors
© all boys
© all girls

## CBL $\mathbf{2}^{\text {TM }}$ Program for the TI-89 Titanium

This activity provides a program that can be used when the TI-89 Titanium is connected to a Calculator-Based Laboratory ${ }^{\text {TM }}$ (CBL $2^{\text {TM }}$ ) unit. This program works with the "Newton's Law of Cooling" experiment, and is similar to the "Coffee To Go" experiment in the CBL System Experiment Workbook. You can use your computer keyboard to type lengthy text and then use TI Connect ${ }^{\text {TM }}$ software to send it to the calculator. More CBL $2{ }^{\text {TM }}$ programs are available from the TI Web site at educaton.ti.com.

| Program Instruction | Description |
| :--- | :--- |
| :cooltemp ( ) | Program name |
| Prgm |  |


| Program Instruction | Description |
| :---: | :---: |
| :Local i | Declare local variable; exists only at run time. |
| :setMode("Graph","FUNCTION") | Set up the TI-89 Titanium for function graphing. |
| :PlotsOff | Turn off any previous plots. |
| :FnOff | Turn off any previous functions. |
| :CIrDraw | Clear any items previously drawn on graph screens. |
| :CIrGraph | Clear any previous graphs. |
| :ClrıO | Clear the TI-89 Titanium Program IO (input/output) screen. |
|  | Set up the Window variables. |
| :-20 $\rightarrow$ ymin: $100 \rightarrow$ ymax $10 \rightarrow$ yscl |  |
| :\{0\} $\rightarrow$ data | Create and/or clear a list named data. |
| : $\{0\} \rightarrow$ time | Create and/or clear a list named time. |
| :Send $\{1,0\}$ | Send a command to clear the CBL $2^{\text {TM }}$ unit. |
| :Send\{1,2,1\} | Set up Chan. 2 of the CBL $2^{\text {TM }}$ to AutolD to record temp. |
| :Disp "Press ENTER to start" | Prompt the user to press ENTER. |
| :Disp "graphingTemperature." |  |
| :Pause | Wait until the user is ready to start. |
| :PtText "TEMP(C)",2,99 | Label the y axis of the graph. |


| Program Instruction | Description |
| :---: | :---: |
| :PtText "T(S)",80,-5 | Label the x axis of the graph. |
| :Send\{3,1,-1,0\} | Send the Trigger command to the CBL $2{ }^{\text {TM }}$; collect data in real-time. |
| :For i, 1,99 | Repeat next two instructions for 99 temperature readings. |
| :Get data[i] | Get a temperature from the CBL $2^{\text {™ }}$ and store it in a list. |
| :PtOn i,data[i] | Plot the temperature data on a graph. |
| :EndFor |  |
| :seq(i, i, 1, 99, 1) $\rightarrow$ time | Create a list to represent time or data sample number. |
| :NewPlot 1,1,time,data,,,,4 | Plot time and data using NewPlot and the Trace tool. |
| :DispG | Display the graph. |
| :PtText "TEMP(C)",2,99 | Re-label the axes. |
| :PtText "T(S)",80,-5 |  |
| :EndPrgm | Stop the program. |

You can also use the Calculator-Based Ranger ${ }^{T M}$ system (CBR ${ }^{\text {TM }}$ ) to explore the mathematical and scientific relationships between distance, velocity, acceleration, and time using data collected from activities you perform.

## Studying the Flight of a Hit Baseball

This activity uses the split screen settings to show a parametric graph and a table at the same time to study the flight of a hit baseball.

## Setting Up a Parametric Graph and Table

Perform the following steps to study the flight of a hit baseball that has an initial velocity of 95 feet per second and an initial angle of 32 degrees.

1. Set the modes for Page $\mathbf{1}$ as shown in this screen.

2. Set the modes for Page $\mathbf{2}$ as shown in this screen.

3. In the $Y=$ Editor on the left side, enter the equation for the distance of the ball at time $\mathbf{t}$ for $\mathbf{x t 1}(\mathbf{t})$.
$\mathbf{x t 1}(\mathrm{t})=95 * \mathbf{t} * \cos \left(32^{\circ}\right)$
Note: Press 2nd [ ${ }^{\circ}$ ] to obtain the degree
 symbol.
4. In the $\mathbf{Y}=$ Editor, enter the equation for the height of the ball at time $t$ for $\mathbf{y t 1}(\mathrm{t})$.
$\mathrm{yt} 1(\mathrm{t})=\mathbf{- 1 6} * \mathrm{t}^{\wedge} \mathbf{2}+95 * \mathbf{t} * \sin \left(32^{\circ}\right)$

5. Set the Window variables to:
$t$ values $=[0,4, .1]$
$x$ values $=[0,300,50]$
$y$ values= $[0,100,10]$

6. Switch to the right side and display the graph.

Note: Press [2nd [ $\boxplus$ ].

7. Display the TABLE SETUP dialog box, and change tblStart to 0 and $\Delta$ tbl to 0.1 .

Note: Press $\rightarrow$ [TBLSET].

8. Display the table in the left side and press $\Theta$ to highlight $\mathbf{t}=\mathbf{2}$.

Note: Press [TABLE].

9. Switch to the right side. Press F3, and trace the graph to show the values of $\mathbf{x c}$ and $\mathbf{y c}$ when $\mathbf{t c}=\mathbf{2}$.
Note: As you move the trace cursor from tc=0.0 to tc=3.1, you will see the position
 of the ball at time $\mathbf{t c}$.

## Optional Exercise

Assuming the same initial velocity of 95 feet per second, find the angle that the ball should be hit to achieve the greatest distance.

## Visualizing Complex Zeros of a Cubic Polynomial

This activity describes graphing the complex zeros of a cubic polynomial.

## Visualizing Complex Roots

Perform the following steps to expand the cubic polynomial $(\mathbf{x}-1)(\mathbf{x}-i)(\mathbf{x}+i)$, find the absolute value of the function, graph the modulus surface, and use the Trace tool to explore the modulus surface.

1. On the Home screen, use the expand( ) function to expand the cubic expression $(\mathbf{x}-1)(\mathbf{x}-i)(\mathbf{x}+i)$ and see the first
 polynomial.
2. Copy and paste the last answer to the entry line and store it in the function $f(\mathbf{x})$.

Note: Move the cursor into the history
(This calculation may take about 2 minutes.)

Note: The absolute value of a function forces any roots to visually just touch rather than cross the $\mathbf{x}$ axis. Likewise, the absolute value of a function of two variables will force any roots to visually just touch the xy plane.
4. Copy and paste the last answer to the entry line and store it in the function z1(x,y).
Note: The graph of $\mathbf{z 1}(\mathbf{x}, \mathbf{y})$ will be the modulus surface.
 press ENTER, to copy it to the entry line.
3. Use the abs() function to find the absolute value of $\mathrm{f}(\mathbf{x}+\mathbf{y} i)$.
5. Set the unit to 3D graph mode, turn on the axes for graph format, and set the Window variables to:
eye= $[20,70,0]$
$x=\quad[-2,2,20]$

$y=\quad[-2,2,20]$
$z=\quad[-1,2]$
ncontour= [5]
6. In the $Y=$ Editor, press:

园 -
and set the Graph Format variables to:
Axes= ON
Labels= ON
Style= HIDDEN SURFACE
Note: Calculating and drawing the graph takes about three minutes.
7. Graph the modulus surface.

The 3D graph is used to visually display a picture of the roots where the surface touches the xy plane.

8. Use the Trace tool to explore the function values at $\mathbf{x}=1$ and $\mathbf{y}=\mathbf{0}$.

9. Use the Trace tool to explore the function values at $\mathbf{x}=\mathbf{0}$ and $\mathbf{y}=1$.

10. Use the Trace tool to explore the function values at $\mathbf{x = 0}$ and $\mathbf{y = - 1}$.


## Summary

Note that $\mathbf{z c}$ is zero for each of the function values in steps $7-9$. Thus, the complex zeros $1,-i, i$ of the polynomial $\mathbf{x}^{3}-\mathbf{x}^{2}+\mathbf{x}-\mathbf{1}$ can be visualized with the three points where the graph of the modulus surface touches the xy plane.

## Solving a Standard Annuity Problem

This activity can be used to find the interest rate, starting principal, number of compounding periods, and future value of an annuity.

## Finding the Interest Rate of an Annuity

Perform the following steps to find the interest rate (i) of an annuity where the starting principal ( $\mathbf{p}$ ) is 1,000, number of compounding periods ( $\mathbf{n}$ ) is 6 , and the future value ( $\mathbf{s}$ ) is 2,000.

1. On the Home screen, enter the equation to solve for $\mathbf{p}$.

| - solve( $\left.s=P \cdot(1+i)^{n}, F\right)$ |  |  |  |
| :---: | :---: | :---: | :---: |
| $P=(i+1)^{-n} E$ |  |  |  |
|  |  |  |  |
| MAIM | Find illta | FUWIC | 1,30 |

2. Enter the equation to solve for $\mathbf{n}$.

$$
\begin{aligned}
& \text { - solve( } \left.s=F \cdot(1+i)^{n}, n\right)
\end{aligned}
$$

3. Enter the equation to solve for $i$ using the with operator.
solve(s=p*(1+i)^n,i)|s=2000 and $\mathbf{p = 1 0 0 0}$ and $\mathrm{n}=6$

Result: The interest rate is $12.246 \%$.
Note:

- To enter the "with" ( | ) operator:

园

- Press ENTER to obtain a floatingpoint result.
- solve $\left(s=p \cdot(1+i)^{n}, i\right) \mid s=-1$




## Finding the Future Value of an Annuity

Find the future value of an annuity using the values from the previous example where the interest rate is $14 \%$.

Enter the equation to solve for $\mathbf{s}$.
solve(s=p*(1+i)^n,s)|i=.14 and $p=1000$ and
 $\mathrm{n}=6$

Result: The future value at $14 \%$ interest is 2,194.97.

## Computing the Time-Value-of-Money

This activity creates a function that can be used to find the cost of financing an item. Detailed information about the steps used in this example can be found in Programming.

## Time-Value-of- Money Function

In the Program Editor, define the following Time-Value-of-Money (tvm) function where temp1 = number of payments, temp2 = annual interest rate, temp3 = present value, temp4 $=$ monthly payment, temp5 = future value, and temp6 = begin- or end-of-payment
period ( $1=$ beginning of month, $0=$ end of month $)$.
:tvm(temp1,temp2,temp3,temp4,temp5,temp6)
:Func
:Local tempi,tempfunc,tempstr1
:-temp3+(1+temp2/1200 temp6) temp4 ((1-(1+temp2/1200)^
(-temp1))/(temp2/1200))-temp5 (1+temp2/1200)^(-temp1)
$\rightarrow$ tempfunc
:For tempi, 1,5,1
:"temp"\&exact(string(tempi)) $\rightarrow$ 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))
:Endlf
:EndFor
:Return "parameter error"
:EndFunc

Note: You can use your computer keyboard to type lengthy text and then use TI Connect ${ }^{\text {TM }}$ software to send it to the TI-89 Titanium.

## Finding the Monthly Payment

Find the monthly payment on 10,000 if you make 48 payments at $10 \%$ interest per year.

On the Home screen, enter the tvm values to find pmt .


Result: The monthly payment is 251.53 .

## Finding the Number of Payments

Find the number of payments it will take to pay off the loan if you could make a 300 payment each month.

On the Home screen, enter the tvm values to find $\mathbf{n}$.


Result: The number of payments is 38.8308 .

## Finding Rational, Real, and Complex Factors

This activity shows how to find rational, real, or complex factors of expressions. Detailed information about the steps used in this example can be found in Symbolic Manipulation.

## Finding Factors

Enter the expressions shown below on the Home screen.

1. factor( $\left.x^{\wedge} 3-5 x\right)$ ENTER displays a rational result.
2. factor( $\left.x^{\wedge} 3+5 x\right)$ ENTER displays a rational result.
3. factor( $\left.\mathbf{x}^{\wedge} 3-5 \mathrm{x}, \mathrm{x}\right)$ ENTER displays a real result.
4. cfactor( $\left.x^{\wedge} 3+5 x, x\right)$ ENTER displays a complex result.




$$
\text { - } \cos \cot \left(x^{3}+5 \cdot x, x\right)
$$

$$
\begin{aligned}
& \text { - factor }\left(x^{3}-5 \cdot x, x\right)
\end{aligned}
$$

## Simulation of Sampling without Replacement

This activity simulates drawing different colored balls from an urn without replacing them. Detailed information about the steps used in this example can be found in Programming.

## Sampling-without- Replacement Function

In the Program Editor, define drawball( ) as a function that can be called with two parameters. The first parameter is a list where each element is the number of balls of a
certain color. The second parameter is the number of balls to select. This function returns a list where each element is the number of balls of each color that were selected.
:drawball(urnlist,drawnum)
:Func
:Local templist,drawlist,colordim, numballs,i,pick,urncum,j
:If drawnum>sum(urnlist)
:Return "too few balls"
:dim(urnlist) $\rightarrow$ colordim
:urnlist $\rightarrow$ templist
:newlist(colordim) $\rightarrow$ drawlist
:For i,1,drawnum,1
:sum(templist) $\rightarrow$ numballs
:rand(numballs) $\rightarrow$ pick
(continued in next column)
:For j, 1,colordim, 1
:cumSum(templist) $\rightarrow$ urncum
:If pick $\leq$ urncum[j] Then
:drawlist[j]+1 $\rightarrow$ drawlist[j]
:templist[j]-1 $\rightarrow$ templist[j]
:Exit
:Endlf
:EndFor
:EndFor
:Return drawlist
:EndFunc

## Sampling without Replacement

Suppose an urn contains n1 balls of a color, n2 balls of a second color, n3 balls of a third color, etc. Simulate drawing balls without replacing them.

1. Enter a random seed using the RandSeed command.

2. Assuming the urn contains 10 red balls and 25 white balls, simulate picking 5 balls at random from the urn without replacement. Enter drawball(\{10,25\},5).


Result: 2 red balls and 3 white balls.

## Using Vectors to Determine Velocity

A small fishing boat leaves from the south bank of the Allegheny River and heads at a $80^{\circ}$ angle with an engine speed of 20 knots. However, the eastward force of the current carries the boat along so it actually travels at a $60^{\circ}$ angle with the shore.

How fast is the current, and how fast does the boat actually travel?

river bank

1. Set the modes for Page $\mathbf{1}$ as shown in this screen. (Show angles in degrees instead of radians and display all digits with a floating decimal point.)


Press: $\operatorname{MODE} \odot \ominus \ominus$. On the Angle option, select 2:DEGREE. On the Display Digits option, select E:FLOAT.
2. Enter vectors describing the initial path of the boat, water current, and resultant path of the boat.

Store these vectors as $\mathbf{i}, \mathbf{c}$, and $\mathbf{r}$. Use the
 value a for the unknown speed of the current. Use the value $\mathbf{b}$ for the speed of the boat.

Enter:
$\left[20,80^{\circ}\right] \rightarrow i$
$\left[\mathrm{a}, \mathrm{O}^{\circ}\right] \rightarrow \mathrm{c}$
$\left[b, 60^{\circ}\right] \rightarrow r$
Vectors are commonly written in either polar or rectangular form, so it is useful to convert polar vectors into rectangular form.
3. Define function $\mathbf{p} 2 \mathrm{r}$.


Enter: Define $\mathrm{p} 2 \mathrm{r}(\mathrm{x})=\left[\mathrm{x}[1,1]^{*} \cos (\mathrm{x}[1,2])\right.$, $\left.x[1,1]^{*} \sin (x[1,2])\right]$

When converted to rectangular form, the sum of vectors $\mathbf{i}$ and $\mathbf{c}$ equals the resultant vector $\mathbf{r}$.
4. Using function $\mathbf{p} 2 \mathbf{r}$, convert vectors $\mathbf{i}, \mathbf{c}$, and $r$ to rectangular form.
Enter:

$$
p 2 r(i) \rightarrow i
$$

$$
\mathrm{p} 2 \mathrm{r}(\mathrm{c}) \rightarrow \mathrm{c}
$$

$$
\mathrm{p} 2 \mathrm{r}(\mathrm{r}) \rightarrow \mathrm{r}
$$

Because the vectors are equal, the $x$ coordinate of $\mathbf{i}+\mathbf{c}$ must equal the x -coordinate of the resultant vector $r$. Likewise, the $y$ coordinate of $\mathbf{i}+\mathrm{c}$ must equal the y -coordinate of resultant vector $\mathbf{r}$.
5. Set up two equations involving vectors ifc and $\mathbf{r}$.

- Equation 1 sets the $x$-coordinates equal to each other.
- Equation 2 sets the y-coordinates equal.

Store these equations into eq and eq, respectively. Enter:
$i[1,1]+c[1,1]=r[1,1] \rightarrow e q 1$
$i[1,2]+c[1,2]=r[1,2] \rightarrow e q 2$
$[1,2]+c[1,2]=r[1,2]>$ eq 2
,

6. Solve eq2 for $\mathbf{b}$ to calculate the actual speed of the boat.
solve(eq2,b)
7. Substitute the known value of $\mathbf{b}$ into eq1, and solve eq1 for a to determine a, the speed of the eastward traveling current. solve(eq1,a) | b

The boat travels at a speed of 22.7 knots, and the water current is approximately 7.9 knots.

## Appendix A: Functions and Instructions

Quick-Find Locator ..... 760
Alphabetical Listing of Operations ..... 764

This section describes the syntax and action of each TI-89 Titanium/ Voyage ${ }^{\text {TM }} 200$ function and instruction that is included in the operating system (OS). See modules relating to calculator software applications (Apps) for functions and instructions specific to those Apps.


## Quick-Find Locator

This section lists the TI-89 Titanium / Voyage ${ }^{\text {TM }} 200$ functions and instructions in functional groups along with the page numbers where they are described.

| Algebra | I ("with") <br> cSolve() <br> factor() <br> nSolve() <br> solve() <br> zeros() | 885 <br> 776 <br> 795 <br> 822 <br> 851 <br> 868 | cFactor() cZeros getDenom() propFrac() tCollect() | 768 <br> 780 <br> 801 <br> 829 <br> 861 | comDenom() <br> expand() <br> getNum() <br> randPoly() <br> tExpand() | 771 <br> 793 <br> 802 <br> 834 <br> 861 |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Calculus | ```f()(integrate) arcLen() deSolve() limit() ' (prime)``` | 880 <br> 766 <br> 784 <br> 809 <br> 883 | П() (product) avgRC() fMax() nDeriv() seq() | 881 <br> 767 <br> 796 <br> 818 <br> 841 | $\begin{aligned} & \Sigma() \text { (sum) } \\ & d() \\ & \text { fMin() } \\ & \text { nInt() } \\ & \text { taylor() } \\ & \hline \end{aligned}$ | 881 <br> 782 <br> 797 <br> 820 <br> 860 |
| Graphics | AndPic <br> ClrDraw <br> DrawFunc <br> DrawPol <br> FnOff <br> Line <br> LineVert <br> PtOff <br> PtText <br> PxiHorz <br> PxIOn <br> PxiVert <br> RplcPic <br> StoPic <br> XorPic <br> ZoomDec <br> ZoomInt <br> ZoomRcl <br> ZoomSto | 765 <br> 770 <br> 788 <br> 789 <br> 797 <br> 810 <br> 810 <br> 829 <br> 830 <br> 830 <br> 831 <br> 831 <br> 839 <br> 856 <br> 867 <br> 870 <br> 871 <br> 872 <br> 873 | BIdData <br> CIrGraph <br> Drawlnv <br> DrawSIp <br> FnOn <br> LineHorz <br> NewPic <br> PtOn <br> PxiChg <br> PxILine <br> pxITest() <br> RcIGDB <br> Shade <br> Style <br> ZoomBox <br> ZoomFit <br> ZoomOut <br> ZoomSqr <br> ZoomTrig | 768 <br> 770 <br> 788 <br> 789 <br> 797 <br> 810 <br> 819 <br> 829 <br> 830 <br> 830 <br> 831 <br> 835 <br> 846 <br> 856 <br> 869 <br> 871 <br> 872 <br> 872 <br> 873 | Circle <br> CyclePic <br> DrawParm <br> DrwCtour <br> Graph <br> LineTan <br> PtChg <br> ptTest() <br> PxiCrcl <br> PxIOff <br> PxIText <br> RcIPic <br> StoGDB <br> Trace <br> ZoomData <br> ZoomIn <br> ZoomPrev <br> ZoomStd | 769 <br> 780 <br> 789 <br> 790 <br> 804 <br> 810 <br> 829 <br> 829 <br> 830 <br> 831 <br> 831 <br> 835 <br> 855 <br> 863 <br> 870 <br> 871 <br> 872 <br> 873 |
| Lists | + (add) <br> / (divide) <br> augment() <br> $\operatorname{dim}()$ <br> left() <br> mat list() <br> $\min ()$ <br> product() <br> shift() <br> sum() | 873 <br> 875 <br> 766 <br> 787 <br> 809 <br> 815 <br> 816 <br> 828 <br> 847 <br> 857 | - (subtract) <br> - (negate) <br> crossP() <br> $\operatorname{dotP}()$ <br> $\Delta$ list() <br> $\max ()$ <br> newList() <br> right() <br> SortA | 874 <br> 877 <br> 775 <br> 788 <br> 811 <br> 815 <br> 819 <br> 837 <br> 854 | * (multiply) <br> $\wedge$ (power) cumSum() exprlist() listrmat() mid() polyEval() rotate() SortD | 874 <br> 876 <br> 778 <br> 793 <br> 811 <br> 816 <br> 827 <br> 837 <br> 854 |


| Math | + (add) | 873 | - (subtract) | 874 | * (multiply) | 874 |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | / (divide) | 875 | - (negate) | 877 | \% (percent) | 877 |
|  | ! (factorial) | 879 | $\sqrt{ }($ ) (sqr. root) | 881 | $\wedge$ (power) | 876 |
|  | ${ }^{\circ}$ (degree) | 882 | $\angle$ (angle) | 882 | ${ }^{\circ}$, ', " | 883 |
|  | _ (underscore) | 883 | - (convert) | 884 | 10^() | 884 |
|  | Ob, Oh | 886 | -Bin | 767 | Cylind | 780 |
|  | DD | 782 | Dec | 782 | DMS | 788 |
|  | -Hex | 804 | PPolar | 827 | -Rect | 835 |
|  | Sphere | 854 | abs() | 764 | and | 764 |
|  | angle() | 765 | approx() | 766 | ceiling() | 768 |
|  | conj() | 772 | cos | 773 | $\cos ^{-1}()$ | 773 |
|  | $\cosh ()$ | 774 | $\cosh ^{-1}()$ | 774 | $\cot ()$ | 774 |
|  | $\cot ^{-1}()$ | 775 | coth() | 775 | coth ${ }^{-1}$ () | 775 |
|  | csc() | 775 | $\csc ^{-1}()$ | 776 | $\boldsymbol{\operatorname { c s c h }}()$ | 776 |
|  | $\cosh ^{-1}()$ | 776 | E | 790 | $e^{\wedge}()$ | 790 |
|  | exact() | 792 | floor() | 796 | fPart() | 798 |
|  | gcd() | 799 | imag() | 806 | int() | 807 |
|  | intDiv() | 807 | iPart() | 807 | isPrime() | 808 |
|  | lcm() | 808 | $\ln ()$ | 811 | $\log ()$ | 813 |
|  | $\max ()$ | 815 | $\min ()$ | 816 | $\bmod ()$ | 817 |
|  | nCr() | 818 | $n \mathrm{Pr}()$ | 821 | PrRx() | 824 |
|  | PrRy() | 824 | $\mathbf{r}$ (radian) | 882 | RPP $\boldsymbol{\theta}$ () | 833 |
|  | $\mathrm{R} P \mathrm{Pr}()$ | 834 | real() | 835 | remain() | 836 |
|  | rotate() | 837 | round() | 838 | $\mathbf{s e c}()$ | 840 |
|  | $\sec ^{-1}()$ | 840 | sech() | 840 | $\operatorname{sech}^{-1}()$ | 840 |
|  | shift() | 847 | sign() | 848 | $\boldsymbol{s i n}()$ | 849 |
|  | $\boldsymbol{s i n}^{-1}()$ | 849 | $\boldsymbol{\operatorname { s i n h }}()$ | 850 | $\sinh ^{-1}()$ | 850 |
|  | $\boldsymbol{t a n}()$ | 859 | $\tan ^{-1}()$ | 859 | $\boldsymbol{t a n h}()$ | 859 |
|  | tanh ${ }^{-1}$ () | 860 | tmpCnv() | 862 | $\Delta \operatorname{tmpCnv}()$ | 863 |
|  | $\mathbf{x}^{-1}$ | 884 |  |  |  |  |
| Matrices | + (add) | 873 | - (subtract) | 874 | * (multiply) | 874 |
|  | / (divide) | 875 | - (negate) | 877 | .+ (dot add) | 876 |
|  | .- (dot subt.) | 876 | . (dot mult.) | 877 | . $/$ (dot divide) | 877 |
|  | .$^{\wedge}$ (dot power) | 877 | $\wedge$ (power) | 876 | augment() | 766 |
|  | colDim() | 771 | colNorm() | 771 | crossP() | 775 |
|  | cumSum() | 778 | det() | 786 | diag() | 786 |
|  | dim() | 787 | dotP() | 788 | eigVc() | 791 |
|  | eigVI() | 791 | Fill | 796 | identity() | 805 |
|  | listrmat() | 811 | LU | 814 | matrlist() | 815 |
|  | $\max ()$ | 815 | mean() | 815 | median() | 815 |
|  | $\min ()$ | 816 | mRow() | 817 | mRowAdd() | 817 |
|  | newMat() | 819 | norm() | 821 | product() | 828 |
|  | QR | 832 | randMat() | 834 | ref() | 836 |
|  | rowAdd() | 838 | rowDim() | 838 | rowNorm() | 839 |
|  | rowSwap() | 839 | rref() | 839 | simult() | 848 |
|  | stdDev() | 855 | subMat() | 856 | sum() | 857 |
|  | T | 858 | unitV() | 865 | variance() | 865 |
|  | $\mathbf{x}^{-1}$ | 884 |  |  |  |  |

## Programming

| = | 878 | \# | 878 | < | 878 |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\leq$ | 879 | > | 879 | $\geq$ | 879 |
| \# (indirection) | 881 | $\rightarrow$ (store) | 885 | © (comment) | 886 |
| and | 764 | ans() | 766 | Archive | 766 |
| checkTmr() | 769 | ClockOff | 769 | ClockOn | 770 |
| ClrErr | 770 | CIrGraph | 770 | ClrHome | 771 |
| Clrio | 771 | CIrTable | 771 | Copy Var | 772 |
| CustmOff | 779 | CustmOn | 779 | Custom | 779 |
| Cycle | 779 | dayOfWk() | 782 | Define | 783 |
| DelFold | 784 | DelVar | 784 | Dialog | 786 |
| Disp | 787 | DispG | 787 | DispHome | 787 |
| DispTbl | 788 | DropDown | 789 | Else | 791 |
| Elself | 791 | EndCustm | 791 | EndDlog | 791 |
| EndFor | 791 | EndFunc | 791 | Endlf | 791 |
| EndLoop | 792 | EndPrgm | 792 | EndTBar | 792 |
| EndTry | 792 | EndWhile | 792 | entry() | 792 |
| Exec | 793 | Exit | 793 | For | 798 |
| format() | 798 | Func | 799 | Get | 799 |
| GetCalc | 799 | getConfg() | 800 | getDate() | 800 |
| getDtFmt() | 801 | getDtStr() | 801 | getFold() | 801 |
| getKey() | 801 | getMode() | 802 | getTime() | 802 |
| getTmFmt() | 802 | getTmStr() | 802 | getTmZn() | 803 |
| getType() | 803 | getUnits() | 803 | Goto | 804 |
| If | 805 | Input | 806 | InputStr | 807 |
| isClkOn() | 808 | Item | 808 | Lbl | 808 |
| left() | 809 | Local | 812 | Lock | 812 |
| Loop | 814 | MoveVar | 817 | NewFold | 819 |
| NewProb | 820 | not | 821 | or | 823 |
| Output | 823 | part() | 824 | PassErr | 826 |
| Pause | 826 | PopUp | 827 | Prgm | 828 |
| Prompt | 828 | Rename | 836 | Request | 836 |
| Return | 837 | right() | 837 | Send | 840 |
| SendCalc | 840 | SendChat | 841 | setDate() | 842 |
| setDtFmt() | 842 | setFold() | 842 | setGraph() | 842 |
| setMode() | 843 | setTable() | 844 | setTime() | 844 |
| setTmFmt() | 844 | setTmZn() | 845 | startTmr() | 854 |
| setUnits() | 845 | Stop | 855 | Style | 856 |
| switch() | 857 | Table | 858 | Text | 861 |
| Then | 861 | timeCnv() | 861 | Title | 862 |
| Toolbar | 863 | Try | 864 | Unarchiv | 864 |
| Unlock | 865 | when() | 865 | While | 866 |
| xor | 867 |  |  |  |  |


| Statistics | ! (factorial) cumSum() LnReg median() NewData OneVar PowerReg rand() ShowStat SortD variance() | 879 <br> 778 <br> 812 <br> 815 <br> 818 <br> 822 <br> 828 <br> 834 <br> 848 <br> 854 <br> 865 | BIdData <br> ExpReg <br> Logistic <br> MedMed <br> NewPlot <br> PlotsOff <br> QuadReg <br> randNorm() <br> SinReg <br> stdDev() | 768 <br> 795 <br> 813 <br> 816 <br> 820 <br> 827 <br> 832 <br> 834 <br> 851 <br> 855 | CubicReg <br> LinReg <br> mean() <br> nCr() <br> nPr() <br> PlotsOn <br> QuartReg <br> RandSeed <br> SortA <br> TwoVar | 778 <br> 811 <br> 815 <br> 818 <br> 821 <br> 827 <br> 833 <br> 834 <br> 854 <br> 864 |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Strings | \& (append) $\operatorname{dim}()$ inString() ord() shift() | 880 <br> 787 <br> 807 <br> 823 <br> 847 | \# (indirection) <br> expr() <br> left() <br> right() <br> string() | 881 <br> 794 <br> 809 <br> 837 <br> 856 | char() format() mid() rotate() | 769 <br> 798 <br> 816 <br> 837 |

## Alphabetical Listing of Operations

Operations whose names are not alphabetic (such as,$+!$, and $>$ ) are listed at the end of this appendix, starting on page 873. Unless otherwise specified, all examples in this section were performed in the default reset mode, and all variables are assumed to be undefined. Additionally, due to formatting restraints, approximate results are truncated at three decimal places (3.14159265359 is shown as 3.141...).

| abs() | MATH/Number menu |  |
| :---: | :---: | :---: |
|  | abs(expression1) $\Rightarrow$ expression <br> abs(list1) $\Rightarrow$ list <br> abs(matrix 1 ) $\Rightarrow$ matrix <br> Returns the absolute value of the argument. <br> If the argument is a complex number, returns the number's modulus. <br> Note: All undefined variables are treated as real variables. | abs $(\{\pi / 2,-\pi / 3\})$ ENTER $\left\{\frac{\pi}{2}\right.$ <br> $\frac{\pi}{3}$  <br> abs $(2-3 i)$ ENTER $\sqrt{13}$ <br> abs $(z)$ ENTER $\|z\|$ <br> abs $(x+y i)$ ENTER $\sqrt{x^{2}+y^{2}}$ |
| and | MATH/Test and MATH/Base menus |  |
|  | Boolean expression1 and expression2 $\Rightarrow$ Boolean expression <br> Boolean list1 and list2 $\Rightarrow$ Boolean list <br> Boolean matrix1 and matrix2 $\Rightarrow$ Boolean matrix <br> Returns true or false or a simplified form of the original entry. | $\begin{aligned} & x \geq 3 \text { and } x \geq 4 \text { ENTER } x \geq 4 \\ & \{x \geq 3, x \leq 0\} \text { and }\{x \geq 4, x \leq-2\} \text { ENTER } \\ & \{x \geq 4 \quad x \leq-2\} \end{aligned}$ |
|  | integer1 and integer2 $\Rightarrow$ integer | In Hex base mode: |
|  | Compares two real integers bit-by-bit using an and operation. Internally, both integers are converted to signed, 32-bit binary numbers. When corresponding bits are compared, the result is 1 if both bits are 1 ; otherwise, the result is 0 . The returned value represents the bit results, and is displayed according to the Base mode. | 0h7AC36 and 0h3D5F ENTER 0h2C16 $\qquad$ Important: Zero, not the letter O . <br> In Bin base mode: <br> 0b100101 and 0b100 EENTER Ob100 |
|  | You can enter the integers in any number base. For a binary or hexadecimal entry, you must use the 0b or Oh prefix, respectively. Without a prefix, integers are treated as decimal (base 10). | In Dec base mode: <br> 37 and 0b100 ENTER |
|  | If you enter a decimal integer that is too large for a signed, 32 -bit binary form, a symmetric modulo operation is used to bring the value into the appropriate range. | Note: A binary entry can have up to 32 digits (not counting the Ob prefix). A hexadecimal entry can have up to 8 digits. |

AndPic picVarf, row, column]
Displays the Graph screen and logically "ANDS" the picture stored in picVar and the current graph screen at pixel coordinates (row, column).
picVar must be a picture type.
Default coordinates are $(0,0)$, which is the upper left corner of the screen.

In function graphing mode and $Y=$ Editor:
$y 1(x)=\cos (x) \oplus$
[2nd[F6] Style = 3:Square
F6 Style = 3:Square
F2 Zoom = 7:ZoomTrig
F1 = 2:Save Copy As...
Type $=$ Picture, Variable = PIC1
$y 2(x)=\sin (x)$
[2nd [F6] Style = 3:Square

y1 = no checkmark (F4 to
deselect)
F2 Zoom = 7:ZoomTrig


HOME HOME
AndPic PIC1 ENTER


## angle() MATH/Complex menu

angle(expression1) $\Rightarrow$ expression
Returns the angle of expression1, interpreting expression1 as a complex number.
Note: All undefined variables are treated as real variables.

In Degree angle mode:
angle( $0+2 i)$ ENTER 90

In Radian angle mode:
angle (1+i) ENTER $\frac{\pi}{4}$
angle(z) ENTER
angle(x+iy) ENTER
$-2 \operatorname{argle}(z) \frac{-\pi \cdot(\operatorname{sign}(z)-1)}{2}$

- argle (x+icy)

$$
\frac{\pi \cdot \operatorname{sigh(y)}}{2}-\tan ^{-1}\left(\frac{x}{y}\right)
$$

In Radian angle mode:
angle( $\{1+2 \boldsymbol{i}, 3+0 \boldsymbol{i}, 0-4 \boldsymbol{i}\})$ ENTER


| ans（）［2nd［ANS］key |  |
| :---: | :---: |
| ans（）$\Rightarrow$ value <br> ans（integen）$\Rightarrow$ value <br> Returns a previous answer from the Home screen history area． <br> integer，if included，specifies which previous answer to recall．Valid range for integer is from 1 to 99 and cannot be an expression．Default is 1 ，the most recent answer． | To use ans（）to generate the Fibonacci sequence on the Home screen，press： <br> 1 ENTER <br> 1 ENTER <br> ［2nd［ANS］$⿴ 囗 十$ 2nd［ANS］© © $\square 2$ ENTER ENTER <br> ENTER |
| approx（）MATH／Algebra menu |  |
| approx（expression）$\Rightarrow$ value <br> Returns the evaluation of expression as a decimal value，when possible，regardless of the current Exact／Approx mode． <br> This is equivalent to entering expression and pressing ENTER on the Home screen． | $\operatorname{approx}(\pi)$ ENTER 3．141．．． |
| approx（list1）$\Rightarrow$ list <br> approx（matrix1）$\Rightarrow$ matrix <br> Returns a list or matrix where each element has been evaluated to a decimal value，when possible． |  |
| Archive CATALOG |  |
| Archive var1［，var2］［，var3］．．． <br> Moves the specified variables from RAM to the user data archive memory． <br> You can access an archived variable the same as you would a variable in RAM．However，you cannot delete，rename，or store to an archived variable because it is locked automatically． <br> To unarchive variables，use Unarchiv． |  |
| arcLen（）MATH／Calculus menu |  |
| arcLen（expression1，var，start，enc）$\Rightarrow$ expression <br> Returns the arc length of expression from start to end with respect to variable var． <br> Regardless of the graphing mode，arc length is calculated as an integral assuming a function mode definition． | $\begin{aligned} & \operatorname{arcLen}(\cos (x), x, 0, \pi) \text { ENTER } 3.820 \ldots \\ & \operatorname{arcLen}(f(x), x, a, b) \text { ENTER } \\ & \qquad \int_{a}^{b} \sqrt{\left(\frac{d}{d x}(f(x))\right)^{2}+1} d x \end{aligned}$ |
| $\operatorname{arcLen}(\text { list1, ,var,start,enc) } \Rightarrow \text { list }$ <br> Returns a list of the arc lengths of each element of list1 from start to end with respect to var． | $\begin{array}{r} \operatorname{arcLen}(\{\sin (x), \cos (x)\}, x, 0, \pi) \\ \{3.820 \ldots \\ 3.820 \ldots\} \end{array}$ |
| augment（）MATH／Matrix menu |  |
| $\text { augment }(\text { listt }, \text { list2) }) \Rightarrow \text { list }$ <br> Returns a new list that is list2 appended to the end of list1． | $\begin{gathered} \operatorname{augment}(\{1,-3,2\},\{5,4\}) \text { ENTER } \\ \{1-3254\} \end{gathered}$ |

> of list1.
$\begin{array}{ll}\text { augment(matrix1, matrix2) } & \Rightarrow \text { matrix } \\ \text { augment(matrix1; matrix2) }\end{array} \Rightarrow$ matrix
Returns a new matrix that is matrix2 appended to matrix1. When the "," character is used, the matrices must have equal row dimensions, and matrix2 is appended to matrix 1 as new columns. When the ";" character is used, the matrices must have equal column dimensions, and matrix2 is appended to matrix1 as new rows. Does not alter matrix1 or matrix2.
$[1,2 ; 3,4] \rightarrow$ M1 ENTER $\quad\left[\begin{array}{ll}1 & 2 \\ 3 & 4\end{array}\right]$
$[5 ; 6] \rightarrow$ M2 ENTER
$\left[\begin{array}{l}5 \\ 6\end{array}\right]$
augment (M1,M2) ENTER $\quad\left[\begin{array}{lll}1 & 2 & 5 \\ 3 & 4 & 6\end{array}\right]$
$[5,6] \rightarrow$ M2 ENTER $\left[\begin{array}{ll}5 & 6\end{array}\right]$
augment(M1;M2) ENTER

## avgRC() CATALOG

$\operatorname{avgRC}($ expression $1, \operatorname{var}[, h]) \Rightarrow$ expression
Returns the forward-difference quotient (average rate of change).
expression1 can be a user-defined function name (see Func).
$h$ is the step value. If $h$ is omitted, it defaults to 0.001 .

Note that the similar function nDeriv() uses the central-difference quotient.
$\operatorname{avgRC}(f(x), x, h)$ ENTER

$$
\frac{f(x+h)-f(x)}{h}
$$

$\operatorname{avgRC}(\sin (x), x, h) \mid x=2$ ENTER

$$
\frac{\sin (h+2)-\sin (2)}{h}
$$

$\operatorname{avgRC}\left(x^{\wedge} 2-x+2, x\right)$ ENTER
2. $\cdot(x-.4995)$
$\operatorname{avgRC}\left(x^{\wedge} 2-x+2, x, .1\right)$ ENTER
2. $\cdot(x-.45)$
$\operatorname{avgRC}\left(x^{\wedge} 2-x+2, x, 3\right)$ ENTER $2 \cdot(x+1)$

## Bin MATH/Base menu

integer1 $>$ Bin $\Rightarrow$ integer
Converts integer1 to a binary number. Binary or hexadecimal numbers always have a 0 b or 0 h prefix, respectively

- Zero, not the letter O, followed by b or h.

Ob binaryNumber
Oh hexadecimalNumber

- A binary number can have up to 32 digits. A hexadecimal number can have up to 8 .

Without a prefix, integer1 is treated as decimal (base 10). The result is displayed in binary, regardless of the Base mode.

If you enter a decimal integer that is too large for a signed, 32-bit binary form, a symmetric modulo operation is used to bring the value into the appropriate range.

256 Bin ENTER
0b100000000
Oh1F Bin ENTER

BIdData [dataVar]
Creates data variable dataVar based on the information used to plot the current graph. BIdData is valid in all graphing modes.

If dataVar is omitted, the data is stored in the system variable sysData.

Note: The first time you start the Data/Matrix Editor after using BIdData, dataVar or sysData (depending on the argument you used with BIdData) is set as the current data variable.

The incremental values used for any independent variables ( $x$ in the example to the right) are calculated according to the Window variable values

For information about the increments used to evaluate a graph, refer to the module that describes that graphing mode.

3D graphing mode has two independent variables. In the sample data to the right, notice that $x$ remains constant as y increments through its range of values.

Then, $x$ increments to its next value and $y$ again increments through its range. This pattern continues until $x$ has incremented through its range.

In function graphing mode and Radian angle mode:
$8 * \sin (x) \rightarrow y 1(x)$ ENTER Done
$2 * \sin (x) \rightarrow y 2(x)$ ENTER Done
ZoomStd ENTER


HOME
BldData ENTER
Done
APPS 6 ENTER


Note: The following sample data is from a 3D graph.


## ceiling() MATH/Number menu

## ceiling(expression1) $\Rightarrow$ integer

ceiling(0.456) ENTER
Returns the nearest integer that is $\geq$ the argument.
The argument can be a real or a complex number.
Note: See also floor().

```
ceiling(list1) }=>\mathrm{ list
ceiling(matrix1) m matrix
```

Returns a list or matrix of the ceiling of each element.

## ceiling( $\{-3.1,1,2.5\})$ ENTER

 $\left\{\begin{array}{lll}-3 . & 1 & 3 .\end{array}\right\}$ceiling([0,-3.2i;1.3,4]) ENTER $\left[\begin{array}{cc}0 & -3 \cdot \cdot \boldsymbol{i} \\ 2 . & 4\end{array}\right]$

## cFactor() MATH/Algebra/Complex menu

cFactor(expression1[, var]) $\Rightarrow$ expression
cFactor(list1[,var]) $\Rightarrow$ list
cFactor(matrix $1[$ var $]$ ) $\Rightarrow$ matrix
cFactor(expression1) returns expression1 factored with respect to all of its variables over a common denominator.
expression1 is factored as much as possible toward linear rational factors even if this introduces new non-real numbers. This alternative is appropriate if you want factorization with respect to more than one variable.
cFactor (a^3*x^2+a* $\left.x^{\wedge} 2+a^{\wedge} 3+a\right)$
ENTER

$$
a \cdot(a+-i) \cdot(a+i) \cdot(x+-i) \cdot(x+i)
$$

cFactor $\left(x^{\wedge} 2+4 / 9\right)$ ENTER

$$
\frac{(3 \cdot x+-2 \cdot \boldsymbol{i}) \cdot(3 \cdot x+2 \cdot \boldsymbol{i})}{9}
$$

cFactor $\left(x^{\wedge} 2+3\right)$ ENTER

$$
x^{2}+3
$$

cFactor $\left(x^{\wedge} 2+a\right)$ ENTER $x^{2}+a$
cFactor(expression1,var) returns expression1 factored with respect to variable var.
expression1 is factored as much as possible toward factors that are linear in var, with perhaps non-real constants, even if it introduces irrational constants or subexpressions that are irrational in other variables.

The factors and their terms are sorted with var as the main variable. Similar powers of var are collected in each factor. Include var if factorization is needed with respect to only that variable and you are willing to accept irrational expressions in any other variables to increase factorization with respect to var. There might be some incidental factoring with respect to other variables.

For the AUTO setting of the Exact/Approx mode, including var also permits approximation with floating-point coefficients where irrational coefficients cannot be explicitly expressed concisely in terms of the built-in functions. Even when there is only one variable, including var might yield more complete factorization.
Note: See also factor().
char() MATH/String menu
char(integer) $\Rightarrow$ character
Returns a character string containing the character numbered integer from the
TI-89 Titanium/Voyage ${ }^{\text {TM }} 200$ character set. See Appendix B for a complete listing of character codes. The valid range for integer is $0-255$.
cFactor $\left(a^{\wedge} 3 * x^{\wedge} 2+a * x^{\wedge} 2+a^{\wedge} 3+a, x\right)$ ENTER

$$
a \cdot\left(a^{2}+1\right) \cdot(x+-i) \cdot(x+i)
$$

cFactor $\left(x^{\wedge} 2+3, x\right)$ ENTER

$$
(x+\sqrt{3} \cdot i) \cdot(x+-\sqrt{3} \cdot i)
$$

cFactor $\left(x^{\wedge} 2+a, x\right)$ ENTER

$$
(x+\sqrt{a} \cdot-i) \cdot(x+\sqrt{a} \cdot i)
$$

cFactor $\left(x^{\wedge} 5+4 x^{\wedge} 4+5 x^{\wedge} 3-6 x-3\right)$
ENTER

$$
x^{5}+4 \cdot x^{4}+5 \cdot x^{3}-6 \cdot x-3
$$

cFactor (ans(1), x) ENTER

$$
\begin{array}{r}
(x-.965) \cdot(x+.612) \cdot(x+2.13) \cdot \\
(x+1.11-1.07 \cdot \boldsymbol{i}) \cdot \\
(x+1.11+1.07 \cdot \boldsymbol{i})
\end{array}
$$

char (38) ENTER "\&"
char $(65)$ ENTER "A"

## checkTmr() CATALOG

checkTmr(starttime) $\Rightarrow$ integer
Returns an integer representing the number of seconds that have elapsed since a timer was started. starttime is an integer returned from the startTmr() function.

You can also use a list or matrix of starttime integers. Valid startime integers must fall between 0 and the current time of the clock. You can run multiple timers simultaneously

Note: See also startTmr() and timeCnv()

## Circle catalog

Circle $x, y, r[$, drawMode]
Draws a circle with its center at window coordinates $(x, y)$ and with a radius of $r$.
$x, y$, and $r$ must be real values.
If drawMode $=1$, draws the circle (default).
If drawMode $=0$, turns off the circle.
If drawMode $=-1$, inverts pixels along the circle
Note: Regraphing erases all drawn items. See also PxiCrcl.
startTmr() ENTER
148083315
checkTmr(148083315)
startTmr() $\rightarrow$ Timer1
:
startTmr() $\rightarrow$ Timer2
$\vdots$
checkTmr(Timer1) $\rightarrow$ Timer1Value :
checkTmr(Timer2) $\rightarrow$ Timer2Value

In a ZoomSqr viewing window:
ZoomSqr:Circle 1,2,3 ENTER


## ClockOff catalog

## ClockOff

Turns the clock OFF.

## ClockOn CATALOG

## ClockOn

Turns the clock ON.

## CIrDraw catalog

## CIrDraw

Clears the Graph screen and resets the Smart Graph feature so that the next time the Graph screen is displayed, the graph will be redrawn.

While viewing the Graph screen, you can clear all drawn items (such as lines and points) by pressing F4 (ReGraph) or pressing:
[0] 2nd [F6]
1i F6 and selecting 1:CIrDraw.

## CIrErr CATALOG

## CIrErr

Clears the error status. It sets errornum to zero and clears the internal error context variables.

The Else clause of the Try...EndTry in the program should use CIrErr or PassErr. If the error is to be processed or ignored, use CIrErr. If what to do with the error is not known, use PassErr to send it to the next error handler. If there are no more pending Try...EndTry error handlers, the error dialog box will be displayed as normal.

Note: See also PassErr and Try.

Program listing:
:clearerr()
: Prgm
:Plots0ff:Fn0ff:ZoomStd
:For i, 0,238
: $\Delta x * i+x m i n \rightarrow x \operatorname{cord}$
: Try
: Pt0n xcord, $1 n(x c o r d)$
: Else
: If errornum=800 or
errornum=260 Then
ClrErr oclear the error
: Else
: PassErr opass on any other
error
: EndIf
: EndTry
: EndFor
: EndPrgm

## CIrGraph CATALOG

## CIrGraph

Clears any functions or expressions that were graphed with the Graph command or were created with the Table command. (See Graph or Table.)
Any previously selected $Y=$ functions will be graphed the next time that the graph is displayed.

## CIrHome catalog

## ClrHome

Clears all items stored in the entry() and ans() Home screen history area. Does not clear the current entry line.

While viewing the Home screen, you can clear the history area by pressing $\mathbb{F}$ and selecting 8:Clear Home.

For functions such as solve() that return arbitrary constants or integers (@1, @2, etc.), ClrHome resets the suffix to 1 .

## CIrIO CATALOG <br> ClriO

Clears the Program I/O screen.

## CIrTable CATALOG

## CIrTable

Clears all table values. Applies only to the ASK setting on the Table Setup dialog box.
While viewing the Table screen in Ask mode, you can clear the values by pressing F1 and selecting 8:Clear Table.

## colDim() MATH/Matrix/Dimensions menu

colDim(mattix) $\Rightarrow$ expression
colDim([0,1,2;3,4,5]) ENTER
Returns the number of columns contained in matrix.
Note: See also rowDim().

## colNorm() MATH/Matrix/Norms menu

colNorm(matrix) $\Rightarrow$ expression
Returns the maximum of the sums of the absolute values of the elements in the columns in matrix.
$[1,-2,3 ; 4,5,-6] \rightarrow$ mat ENTER

$$
\left[\begin{array}{rrr}
1 & -2 & 3 \\
4 & 5 & -6
\end{array}\right]
$$

colNorm(mat) ENTER

Note: Undefined matrix elements are not allowed. See also rowNorm().

## comDenom() MATH/Algebra menu

```
comDenom(expression1[,var]) }=>\mathrm{ expression
comDenom \(\left(\left(y^{\wedge} 2+y\right) /(x+1)^{\wedge} 2+y^{\wedge} 2+y\right)\)
```

comDenom(list1[,var]) $\Rightarrow$ list
comDenom(matrix1[,var]) $\Rightarrow$ matrix
comDenom(expression1) returns a reduced ratio of a fully expanded numerator over a fully expanded denominator.

ENTER

| ComDenom $\left(\frac{y^{2}+y}{(x+1)^{2}}+y^{2}+y\right)$ |
| :---: |
| $\frac{x^{2} \cdot y^{2}+x^{2} \cdot y+2 \cdot x \cdot y^{2}+2}{}$ |
| $x^{2}+2 \cdot x+1$ |

comDenom(expression1,var) returns a reduced ratio of numerator and denominator expanded with respect to var. The terms and their factors are sorted with var as the main variable. Similar powers of var are collected. There might be some incidental factoring of the collected coefficients. Compared to omitting var, this often saves time, memory, and screen space, while making the expression more comprehensible. It also makes subsequent operations on the result faster and less likely to exhaust memory.

If var does not occur in expression1, comDenom(expression1, var) returns a reduced ratio of an unexpanded numerator over an unexpanded denominator. Such results usually save even more time, memory, and screen space. Such partially factored results also make subsequent operations on the result much faster and much less likely to exhaust memory.

Even when there is no denominator, the comden function is often a fast way to achieve partial factorization if factor() is too slow or if it exhausts memory.

Hint: Enter this comden() function definition and routinely try it as an alternative to comDenom() and factor().
conj() MATH/Complex menu
conj(expression1) $\Rightarrow$ expression
conj(list1) $\Rightarrow$ list
conj(matrix1) $\Rightarrow$ matrix
Returns the complex conjugate of the argument.
Note: All undefined variables are treated as real variables.
comDenom ( $\left(y^{\wedge} 2+y\right) /(x+1)$
^2 $2+y^{\wedge} 2+y, x$ ) ENTER

| combenom $\left(\frac{y^{2}+y}{(x+1)^{2}}+y^{2}+y, v\right.$ |
| :---: |
| $\frac{x^{2} \cdot y \cdot(y+1)+2 \cdot x \cdot y \cdot(y+1)}{x^{2}+2 \cdot x+1}$ |

comDenom ( $\left(y^{\wedge} 2+y\right) /(x+1)$
^2+y^2+y,y) ENTER

| - comDenom $\left(\frac{y^{2}+y}{(x+1)^{2}}+y^{2}+y \cdot v\right.$ |
| :---: |
| $\frac{y^{2} \cdot\left(x^{2}+2 \cdot x+2\right)+y \cdot\left(x^{2}+\right.}{x^{2}+2 \cdot x+1}$ |

comDenom(exprn, abc) $\rightarrow$ comden
(exprn) ENTER Done
comden $\left(\left(y^{\wedge} 2+y\right) /(x+1)^{\wedge} 2+y^{\wedge} 2+y\right)$
ENTER

| - comden $\left(\frac{y^{2}+y}{(x+1)^{2}}+y^{2}+y\right)$ |
| :---: |
| $\frac{\left(x^{2}+2 \cdot x+2\right)}{(x+1)^{2}}(y+1)$ |

comden (1234x^2* ( $\left.\mathrm{y}^{\wedge} 3-\mathrm{y}\right)+2468 \mathrm{x}$

* $\left(y^{\wedge} 2-1\right)$ ) ENTER $1234 \cdot x \cdot(x \cdot y+2) \cdot\left(y^{2}-1\right)$
$\operatorname{conj}(1+2 \boldsymbol{i})$ ENTER $1-2 \cdot \boldsymbol{i}$
$\operatorname{conj}([2,1-3 \mathbf{i} ; \mathbf{- i},-7])$ ENTER

$$
\left[\begin{array}{cc}
2 & 1+3 \cdot i \\
i & -7
\end{array}\right]
$$

conj(z) z
$\operatorname{conj}(x+i y) \quad x+-i \cdot y$

## CopyVar cATALOG

CopyVar var1, var2
Copies the contents of variable var1 to var2. If var2 does not exist, CopyVar creates it.

Note: CopyVar is similar to the store instruction $(\rightarrow)$ when you are copying an expression, list, matrix, or character string except that no simplification takes place when using CopyVar. You must use CopyVar with non-algebraic variable types such as Pic and GDB variables.

| $x+y \rightarrow a$ ENTER | $x+y$ |
| :--- | ---: |
| $10 \rightarrow x$ ENTER | 10 |
| CopyVar a, b ENTER | Done |
| a c c ENTER | $y+10$ |
| DelVar x ENTER | Done |
| $b$ ENTER | $x+y$ |
| c ENTER | $y+10$ |


cos(expression1) $\Rightarrow$ expression
$\cos ($ list 1) $\Rightarrow$ list
$\boldsymbol{\operatorname { c o s }}$ (expression1) returns the cosine of the argument as an expression.
$\boldsymbol{\operatorname { c o s }}$ (list1) returns a list of the cosines of all elements in list1.

Note: The argument is interpreted as either a degree or radian angle, according to the current angle mode setting. You can use ${ }^{\circ}$ or ${ }^{r}$ to override the angle mode temporarily.

In Degree angle mode:
$\cos \left((\pi / 4)^{r}\right)$ ENTER $\frac{\sqrt{2}}{2}$
$\cos (45)$ ENTER $\frac{\sqrt{2}}{2}$
$\cos (\{0,60,90\})$ ENTER $\{1 \quad 1 / 2 \quad 0\}$
In Radian angle mode:
$\cos (\pi / 4)$ ENTER
$\frac{\sqrt{2}}{2}$
$\cos \left(45^{\circ}\right)$ ENTER
$\frac{\sqrt{2}}{2}$
$\boldsymbol{\operatorname { c o s }}($ squareMatrix1) $\Rightarrow$ squareMatrix
Returns the matrix cosine of squareMatrix1. This is not the same as calculating the cosine of each element.

When a scalar function $f(A)$ operates on squareMatrix1 (A), the result is calculated by the algorithm:

1. Compute the eigenvalues $\left(\lambda_{i}\right)$ and eigenvectors $\left(V_{i}\right)$ of $A$.
squareMatrix1 must be diagonalizable. Also, it cannot have symbolic variables that have not been assigned a value.
2. Form the matrices:

$$
\mathrm{B}=\left[\begin{array}{cccc}
\lambda_{1} & 0 & \ldots & 0 \\
0 & \lambda_{2} & \ldots & 0 \\
0 & 0 & \ldots & 0 \\
0 & 0 & \ldots & \lambda_{n}
\end{array}\right] \text { and } \mathrm{X}=\left[\mathrm{V}_{1}, \mathrm{~V}_{2}, \ldots, \mathrm{~V}_{n}\right]
$$

3. Then $A=X B X^{-1}$ and $f(A)=X f(B) X^{-1}$. For example, $\cos (A)=X \cos (B) X^{-1}$ where:
$\cos (B)=\left[\begin{array}{cccc}\cos \left(\lambda_{1}\right) & 0 & \ldots & 0 \\ 0 & \cos \left(\lambda_{2}\right) & \ldots & 0 \\ 0 & 0 & \ldots & 0 \\ 0 & 0 & \ldots & \cos \left(\lambda_{n}\right)\end{array}\right]$
All computations are performed using floating-point arithmetic.

## $\cos ^{-1}()$ 园 $\left[\mathrm{cos}^{-1}\right]$ key [nd $\left[\mathrm{COS}^{-1}\right]$ key

cos $^{-1}$ (expression1) $\Rightarrow$ expression $\cos ^{-1}$ (list1) $\Rightarrow$ list
$\cos ^{-1}$ (expression1) returns the angle whose cosine is expression1 as an expression.
$\cos ^{-1}$ (list1) returns a list of the inverse cosines of each element of list1.

Note: The result is returned as either a degree or radian angle, according to the current angle mode setting.

In Radian angle mode:
$\cos ([1,5,3 ; 4,2,1 ; 6,-2,1])$ ENTER
$\left[\begin{array}{lll}.212 \ldots . & .205 \ldots & .121 \ldots \\ .160 \ldots . . & .259 \ldots & .037 \ldots \\ .248 . . . & -.090 \ldots & .218 \ldots\end{array}\right]$
$\boldsymbol{c o s}^{-1}$ (squareMatrix1) $\Rightarrow$ squareMatrix
Returns the matrix inverse cosine of squareMatrix1. This is not the same as calculating the inverse cosine of each element. For information about the calculation method, refer to $\boldsymbol{\operatorname { c o s } ( ) .}$
squareMatrix1 must be diagonalizable. The result always contains floating-point numbers.

In Radian angle mode and Rectangular complex format mode:
$\cos ^{-1}([1,5,3 ; 4,2,1 ; 6,-2,1])$
ENTER


## $\cosh () \quad$ MATH/Hyperbolic menu

$\cosh$ (expression1) $\Rightarrow$ expression $\cosh ($ list1) $\Rightarrow$ list
cosh (expression1) returns the hyperbolic cosine of the argument as an expression.
cosh (list1) returns a list of the hyperbolic cosines of each element of list1.
$\cosh ($ squareMatrix1) $\Rightarrow$ squareMatrix
Returns the matrix hyperbolic cosine of squareMatrix1. This is not the same as calculating the hyperbolic cosine of each element. For information about the calculation method, refer to $\boldsymbol{\operatorname { c o s } ( ) . ~}$
squareMatrix1 must be diagonalizable. The result always contains floating-point numbers.
$\cosh (1.2)$ ENTER $1.810 \ldots$
$\cosh (\{0,1.2\})$ ENTER $\{1$ 1.810... $\}$

In Radian angle mode
$\cosh ([1,5,3 ; 4,2,1 ; 6,-2,1])$
ENTER
$\left[\begin{array}{lll}421.255 & 253.909 & 216.905 \\ 327.635 & 255.301 & 202.958 \\ 226.297 & 216.623 & 167.628\end{array}\right]$
$\cosh ^{-1}$ () MATH/Hyperbolic menu
$\cosh ^{-1}$ (expression 1$) \Rightarrow$ expression
$\cosh ^{-1}$ (list1) $\Rightarrow$ list
$\cosh ^{-1}$ (expression1) returns the inverse hyperbolic cosine of the argument as an expression.
$\cosh ^{-1}$ (list1) returns a list of the inverse hyperbolic cosines of each element of list1.
$\cosh ^{-1}$ (squareMatrix1) $\Rightarrow$ squareMatrix
Returns the matrix inverse hyperbolic cosine of squareMatrix1. This is not the same as calculating the inverse hyperbolic cosine of each element. For information about the calculation method, refer to $\cos ()$.
squareMatrix1 must be diagonalizable. The result always contains floating-point numbers.
cot() MATH/Trig menu
$\cot ($ expression 1$) \Rightarrow$ expression $\cot ($ list1) $\Rightarrow$ list

Returns the cotangent of expression1 or returns a list of the cotangents of all elements in list1.

Note: The argument is interpreted as either a degree or radian angle, according to the current angle mode.
$\cosh ^{-1}(1)$ ENTER

$\cosh ^{-1}(\{1,2.1,3\})$ ENTER
$\left\{0 \quad 1.372 \ldots \cosh ^{-1}(3)\right\}$

In Radian angle mode and Rectangular complex format mode:
$\cosh ^{-1}([1,5,3 ; 4,2,1 ; 6,-2,1])$
ENTER
$\left[\begin{array}{lll}2.525 \ldots+1.734 \ldots \cdot . \boldsymbol{i} & -.009 \ldots-. .-1.490 \ldots \cdot \boldsymbol{i} & \ldots \\ .486 \ldots-.725 \ldots \cdot \boldsymbol{i} & 1.662 \ldots+.623 \ldots \cdot \boldsymbol{i} & \ldots \\ -.322 \ldots-2 . . .083 \ldots \cdot \boldsymbol{i} & 1.267 \ldots+1.790 \ldots \cdot \boldsymbol{i} & \ldots .\end{array}\right]$

In Degree angle mode: $\cot (45)$ ENTER

In Radian angle mode:
$\cot (\{1,2.1,3\})$ ENTER

$$
\frac{1}{\tan (1)}-.584 \ldots \frac{1}{\tan (3)}
$$

| $\cot ^{-1}()$ | MATH/Trig menu |  |
| :---: | :---: | :---: |
|  | $\begin{aligned} & \text { xpression1) } \Rightarrow \text { expression } \\ & \text { ist1) } \Rightarrow \text { list } \end{aligned}$ <br> Returns the angle whose cotangent is expression1 or returns a list containing the inverse cotangents of each element of list1. <br> Note: The result is returned as either a degree or radian angle, according to the current angle mode. | In Degree angle mode: <br> $\cot ^{-1}(1)$ ENTER <br> In Radian angle mode: <br> $\cot ^{-1}(1)$ ENTER |
| coth() | MATH/Hyperbolic menu |  |
|  | $\text { expression1) } \Rightarrow \text { expression }$ $t 1) \Rightarrow \text { list }$ <br> Returns the hyperbolic cotangent of expression1 or returns a list of the hyperbolic cotangents of all elements of list1. | $\begin{aligned} & \operatorname{coth}(1.2) \text { ENTER } \\ & \operatorname{coth}(\{1,3.2\}) \frac{1.199 \ldots}{\frac{\text { ENTER }}{\tanh (1)}} 1.003 \ldots \end{aligned}$ |
| $\operatorname{coth}^{-1}($ | MATH/Hyperbolic menu |  |
|  | $\begin{aligned} & (\text { expression1) } \Rightarrow \text { expression } \\ & (\text { list1 }) \Rightarrow \text { list } \end{aligned}$ <br> Returns the inverse hyperbolic cotangent of expression1 or returns a list containing the inverse hyperbolic cotangents of each element of list1. | $\begin{aligned} & \operatorname{coth}^{-1}(3.5) \text { ENTER } \\ & \operatorname{coth}^{-1}(\{-2,2.1,6\}) \text { ENTER } \\ & \frac{-\ln (3)}{2} \cdot 518 \ldots \frac{\ln (7 / 5)}{2} \end{aligned}$ |
| crossP() MATH/Matrix/Vector ops menu |  |  |
| cross P (list1, list2) $\Rightarrow$ list <br> Returns the cross product of list1 and list2 as a list. list1 and list2 must have equal dimension, and the dimension must be either 2 or 3 . |  |  |
|  | $\mathbf{P}$ (vector1, vector2) $\Rightarrow$ vector <br> Returns a row or column vector (depending on the arguments) that is the cross product of vector1 and vector2. <br> Both vector1 and vector2 must be row vectors, or both must be column vectors. Both vectors must have equal dimension, and the dimension must be either 2 or 3. |  |
| csc() | MATH/Trig menu |  |
|  | $\begin{aligned} & \text { ression1) } \Rightarrow \text { expression } \\ & ) \Rightarrow \text { list } \end{aligned}$ | In Degree angle mode: |
|  | Returns the cosecant of expression1 or returns a list containing the cosecants of all elements in list1. | $\csc (\pi / 4) \quad \frac{1}{\sin \left(\frac{\pi}{4}\right)}$ |
|  |  | In Radian angle mode: $\csc (\{1, \pi / 2, \pi / 3\})$ ENTER $\frac{1}{\sin (1)} 1 \frac{2 \cdot \sqrt{3}}{3}$ |


| $\csc ^{-1}()$ | MATH/Trig menu |  |
| :---: | :---: | :---: |
|  | $\begin{aligned} & \text { xpression1) } \Rightarrow \text { expression } \\ & \text { ist1) } \Rightarrow \text { list } \end{aligned}$ <br> Returns the angle whose cosecant is expression1or returns a list containing the inverse cosecants of each element of list1. <br> Note: The result is returned as either a degree or radian angle, according to the current angle mode. | In Degree angle mode: $\csc ^{-1}(1)$ ENTER <br> In Radian angle mode: $\csc ^{-1}(\{1,4,6\})$ ENTER $\frac{\pi}{2} \sin ^{-1}(1 / 4) \sin ^{-1}(1 / 6)$ |
| csch() | MATH/Hyperbolic menu |  |
|  | $\begin{aligned} & \text { xpression1) } \Rightarrow \text { expression } \\ & \text { ist1) } \Rightarrow \text { list } \end{aligned}$ | $\operatorname{csch}(3) \text { ENTER } \quad \frac{1}{\sinh (3)}$ |
|  | Returns the hyperbolic cosecant of expression1 or returns a list of the hyperbolic cosecants of all elements of list1. | $\operatorname{csch}(\{1,2.1,4\})$ ENTER $\frac{1}{\sinh (1)} \cdot 248 \ldots \frac{1}{\sinh (4)}$ |
| $\operatorname{csch}^{-1}()$ | MATH/Hyperbolic menu |  |
|  | $\begin{aligned} & (\text { expression1 }) \Rightarrow \text { expression } \\ & (\text { list1 }) \Rightarrow \text { list } \end{aligned}$ <br> Returns the inverse hyperbolic cosecant of expression1 or returns a list containing the inverse hyperbolic cosecants of each element of list1. | $\begin{aligned} & \operatorname{csch}^{-1}(1) \text { ENTER } \sinh ^{-1}(1) \\ & \operatorname{csch}^{-1}(\{1,2.1,3\}) \text { ENTER } \\ & \sinh ^{-1}(1) .459 \ldots \sinh ^{-1}(1 / 3) \end{aligned}$ |
| cSolve() | MATH/Algebra/Complex menu |  |
|  | (equation, var) $\Rightarrow$ Boolean expression <br> Returns candidate complex solutions of an equation for var. The goal is to produce candidates for all real and non-real solutions. Even if equation is real, cSolve() allows non-real results in real mode. <br> Although the TI-89 Titanium/Voyage ${ }^{\text {TM }} 200$ processes all undefined variables that do not end with an underscore (_) as if they were real, cSolve() can solve polynomial equations for complex solutions. | $\begin{aligned} & \text { cSolve }\left(x^{\wedge} 3=-1, x\right) \text { ENTER } \\ & \text { solve }\left(x^{\wedge} 3=-1, x\right) \text { ENTER } \\ & \begin{array}{l} \text { - cosue( } \left.x^{3}=-1, x\right) \\ 1 / 2+\frac{\sqrt{3}}{2} \cdot i \text { or } x=1 / 2-\frac{\sqrt{2}}{\vdots} \\ \text { solve }\left(x^{3}=-1, x\right) \quad x=-1 \end{array} \end{aligned}$ |
|  | cSolve() temporarily sets the domain to complex during the solution even if the current domain is real. In the complex domain, fractional powers having odd denominators use the principal rather than the real branch. Consequently, solutions from solve() to equations involving such fractional powers are not necessarily a subset of those from cSolve(). | $\begin{array}{ll} \text { cSolve }\left(x^{\wedge}(1 / 3)=-1, x\right) \text { ENTER false } \\ \text { solve }\left(x^{\wedge}(1 / 3)=-1, x\right) \text { ENTER } & x=-1 \end{array}$ |
|  | cSolve() starts with exact symbolic methods. Except in EXACT mode, cSolve() also uses iterative approximate complex polynomial factoring, if necessary. | Display Digits mode in Fix 2: $\begin{aligned} & \operatorname{exact}\left(\operatorname { c S o l v e } \left(x^{\wedge} 5+4 x^{\wedge} 4+5 x\right.\right. \\ & \wedge 3-6 x-3=0, x)) \text { ENTER } \end{aligned}$ |
|  | Note: See also cZeros(), solve(), and zeros(). <br> Note: If equation is non-polynomial with functions such as abs(), angle(), conj(), real(), or imag(), you should place an underscore _ $\square$ <br> 2nd [_]) at the end of var. By default, a variable is treated as a real value. <br> If you use var_, the variable is treated as complex. | cSolve(ans(1), x) ENTER $\begin{array}{r} \text { exact (csolve }\left(x^{5}+4 \cdot x^{4}+5\right. \\ x \cdot\left(x^{4}+4 \cdot x^{3}+5 \cdot x^{2}-6\right)=3 \\ \text { csolve }\left(x \cdot \left(x^{4}+4 \cdot x^{3}+5 \cdot x^{2}\right.\right. \\ x=-1.1138+1.07314 \cdot i \end{array}$ <br> z is treated as real: |

You should also use var_ for any other variables in equation that might have unreal values. Otherwise, you may receive unexpected results.

$$
\text { cSolve(conj(z)=1+i,z) ENTER } z=1+i
$$

$z_{-}$is treated as complex:
cSolve(conj( $\left.\left.z_{-}\right)=1+i, z_{-}\right)$ENTER
cSolve(equation1 and equation2 [and ... ],
\{varOrGuess1, varOrGuess2 [, ... ]\})
$\Rightarrow$ Boolean expression
Returns candidate complex solutions to the simultaneous algebraic equations, where each varOrGuess specifies a variable that you want to solve for.

Optionally, you can specify an initial guess for a variable. Each varOrGuess must have the form:
variable

- or -
variable $=$ real or non-real number
For example, $x$ is valid and so is $x=3+i$.
If all of the equations are polynomials and if you do NOT specify any initial guesses, cSolve() uses the lexical Gröbner/Buchberger elimination method to attempt to determine all complex solutions.

Complex solutions can include both real and nonreal solutions, as in the example to the right.

Simultaneous polynomial equations can have extra variables that have no values, but represent given numeric values that could be substituted later.

You can also include solution variables that do not appear in the equations. These solutions show how families of solutions might contain arbitrary constants of the form @ $k$, where $k$ is an integer suffix from 1 through 255. The suffix resets to 1 when you use CIrHome or F1 8:Clear Home.

For polynomial systems, computation time or memory exhaustion may depend strongly on the order in which you list solution variables. If your initial choice exhausts memory or your patience, try rearranging the variables in the equations and/or varOrGuess list.

Note: The following examples use an underscore
 treated as complex.

$$
\begin{aligned}
& \text { cSolve }\left(u_{-} * v_{-}-u_{-}=v_{-}\right. \text {and } \\
& \left.v_{-} \wedge 2=-u_{-},\left\{u_{-}, v_{-}\right\}\right) \text {ENTER } \\
& u_{-}=1 / 2+\frac{\sqrt{3}}{2} \cdot i \text { and } v_{-}=1 / 2-\frac{\sqrt{3}}{2} \cdot i \\
& \text { or } u_{-}=1 / 2-\frac{\sqrt{3}}{2} \cdot i \text { and } v_{-}=1 / 2+\frac{\sqrt{3}}{2} \cdot i \\
& \quad \text { or } u_{-}=0 \text { and } v_{-}=0
\end{aligned}
$$

cSolve(u_*v_-u_=c_*v_ and $\left.v_{-} \wedge 2=-u_{-},\left\{u_{-}, v_{-}\right\}\right)$ENTER

or

cSolve(u_* v_- $u_{-}=v_{-}$and $\left.v_{-} \wedge 2=-u_{-},\left\{u_{-}, v_{-}, w_{-}\right\}\right)$ENTER
$u_{-}=1 / 2+\frac{\sqrt{3}}{2} \cdot i$ and $v_{-}=1 / 2-\frac{\sqrt{3}}{2} \cdot i$ and $w_{-}=@ 1$
or

$$
\begin{array}{r}
u_{-}=1 / 2 \quad-\frac{\sqrt{3}}{2} \cdot i \text { and } v_{-}=1 / 2+\frac{\sqrt{3}}{2} \cdot i \\
\text { and } w_{-}=@ 1 \\
\\
\text { or } u_{-}=0 \text { and } v_{-}=0 \text { and } w_{-}=@ 1
\end{array}
$$

If you do not include any guesses and if any equation is non-polynomial in any variable but all equations are linear in all solution variables, cSolve() uses Gaussian elimination to attempt to determine all solutions.

If a system is neither polynomial in all of its variables nor linear in its solution variables, cSolve() determines at most one solution using an approximate iterative method. To do so, the number of solution variables must equal the number of equations, and all other variables in the equations must simplify to numbers.
A non-real guess is often necessary to determine a non-real solution. For convergence, a guess might have to be rather close to a solution.
cSolve( $u_{-}+v_{-}=e^{\wedge}\left(w_{-}\right)$and $u_{-}-v_{-}=$ $\left.i,\left\{u_{-}, v_{-}\right\}\right)$ENTER

$$
\begin{aligned}
& u_{-}=\frac{e^{w_{-}}}{2}+1 / 2 \cdot i \text { and } v_{-}=\frac{e^{w_{-}-i}}{2} \\
& \operatorname{cSolve}\left(e^{\wedge}\left(z_{-}\right)=w_{-} \text {and } \mathrm{w}_{-}=z_{-}{ }^{\wedge} 2,\right. \\
& \left.\left\{\mathrm{w}_{-}, \mathrm{z}_{-}\right\}\right) \frac{\text { ENTER }}{\mathrm{w}_{-}=.494 \ldots \text { and } z_{-}=-.703 . . .}
\end{aligned}
$$

cSolve( $e^{\wedge}\left(z_{-}\right)=w_{-}$and $w_{-}=z_{-}{ }^{\wedge} 2$, $\left\{w_{-}, z_{-}=1+i\right\}$ ) ENTER

$$
\begin{array}{r}
w_{-}=.149 \ldots+4.891 \ldots \cdot i \text { and } \\
z_{-}=1.588 \ldots+1.540 \ldots \cdot i
\end{array}
$$

## CubicReg MATH/Statistics/Regressions menu

CubicReg list1, list2[, [list3] [, list4, list5]]
Calculates the cubic polynomial regression and updates all the statistics variables.
All the lists must have equal dimensions except for list5.
list1 represents xlist.
list2 represents ylist.
list3 represents frequency.
list4 represents category codes.
list5 represents category include list.
Note: list1 through list4 must be a variable name or c1-c99 (columns in the last data variable shown in the Data/Matrix Editor). list5 does not have to be a variable name and cannot be c1-c99.

In function graphing mode.
$\left.\begin{array}{lll}\left.\begin{array}{lllll}\{0,1,2,3\} \rightarrow L 1 \text { ENTER } & \{0 & 1 & 2 & 3\end{array}\right\} \\ \{0,2,3,4\} \rightarrow L 2 \text { ENTER } & \{0 & 2 \\ 3 & 3\end{array}\right\}$

CubicReg L1, L2 ENTER
Done
ShowStat ENTER


ENTER
regeq $(x) \rightarrow y 1(x)$ ENTER Done
NewPlot 1,1, L1, L2 ENTER Done

- [GRAPH]



## cumSum() MATH/List menu

cumSum(list1) $\Rightarrow$ list
cumSum $(\{1,2,3,4\})$ ENTER
$\left\{\begin{array}{llll}1 & 3 & 6 & 10\end{array}\right\}$
Returns a list of the cumulative sums of the elements in list1, starting at element 1.

```
cumSum(matrix1) = matrix
```

Returns a matrix of the cumulative sums of the elements in matrix1. Each element is the cumulative sum of the column from top to bottom.

| $[1,2 ; 3,4 ; 5,6] \rightarrow m 1$ ENTER | $\left[\begin{array}{ll}1 & 2 \\ 3 & 4 \\ 5 & 6\end{array}\right]$ |
| :--- | :--- |
| cumSum(m1) ENTER | $\left[\begin{array}{ll}1 & 2 \\ 4 & 6 \\ 9 & 12\end{array}\right]$ |

## CustmOff

Removes a custom toolbar.
CustmOn and CustmOff enable a program to control a custom toolbar. Manually, you can press 2nd [CUSTOM] to toggle a custom toolbar on and off. Also, a custom toolbar is removed automatically when you change applications.

## CustmOn CATALOG

## CustmOn

Activates a custom toolbar that has already been set up in a Custom...EndCustm block.

CustmOn and CustmOff enable a program to control a custom toolbar. Manually, you can press 2nd [CUSTOM] to toggle a custom toolbar on and off

## Custom [2nd [custom] key

Custom

## block

EndCustm
Sets up a toolbar that is activated when you press 2nd [CUSTOM]. It is very similar to the ToolBar instruction except that Title and Item statements cannot have labels.
block can be either a single statement or a series of statements separated with the ":" character

Note: 2nd [CUSTOM] acts as a toggle. The first instance invokes the menu, and the second instance removes the menu. The menu is removed also when you change applications.

See Custom program listing example.

## CyclePic CAtALOG

CyclePic picNameString, $n$ [, [wait , [cycles], [direction]]

Displays all the PIC variables specified and at the specified interval. The user has optional control over the time between pictures, the number of times to cycle through the pictures, and the direction to go, circular or forward and backwards.
direction is 1 for circular or -1 for forward and backwards. Default = 1 .

## Cylind MATH/Matrix/Vector ops menu

## vector Cylind

Displays the row or column vector in cylindrical form [ $\mathrm{r} \angle \theta, \mathrm{z}$ ].
vector must have exactly three elements. It can be either a row or a column.

1. Save three pics named pic1, pic2, and pic3.
2. Enter: CyclePic "pic" , 3, . 5, 4, -1
3. The three pictures (3) will be displayed automatically-one-half second (.5) between pictures, for four cycles (4), and forward and backwards ( -1 ).

## cZeros() MATH/Algebra/Complex menu

cZeros(expression, var) $\Rightarrow$ list
Returns a list of candidate real and non-real values of var that make expression=0. cZeros() does this by computing
exp>list(cSolve(expression $=0$, var), var). Otherwise, cZeros() is similar to zeros().

Note: See also cSolve(), solve(), and zeros().
Note: If expression is non-polynomial with functions such as abs(), angle(), conj(), real(), or imag(), you should place an underscore
 default, a variable is treated as a real value. If you use var_, the variable is treated as complex.

You should also use var_ for any other variables in expression that might have unreal values. Otherwise, you may receive unexpected results.

## Display Digits mode in Fix 3 :

## cZeros ( $\left.x^{\wedge} 5+4 x^{\wedge} 4+5 x^{\wedge} 3-6 x-3, x\right)$ ENTER

$$
\begin{array}{rrr}
\{-2.125 & -.612 & .965 \\
-1.114-1.073 \cdot i \\
-1.114+1.073 \cdot i
\end{array}
$$

$z$ is treated as real:
cZeros(conj(z)-1-i,z) ENTER $\{1+i\}$
$z_{-}$is treated as complex:
cZeros(conj(z_)-1-i,z_) ENTER
cZeros(\{expression1, expression2 [, ...]\},
$\{$ varOrGuess1, varOrGuess $2[, \ldots]\}) \Rightarrow$ matrix
Returns candidate positions where the expressions are zero simultaneously. Each varOrGuess specifies an unknown whose value you seek.

Optionally, you can specify an initial guess for a variable. Each varOrGuess must have the form:
variable

- or -
variable $=$ real or non-real number
For example, $x$ is valid and so is $x=3+i$.
If all of the expressions are polynomials and you do NOT specify any initial guesses, cZeros() uses the lexical Gröbner/Buchberger elimination method to attempt to determine all complex zeros.

Note: The following examples use an underscore _ (圈 [-], ․․ [ind [-]) so that the variables will be treated as complex.

Complex zeros can include both real and non-real zeros, as in the example to the right.

Each row of the resulting matrix represents an alternate zero, with the components ordered the same as the varOrGuess list. To extract a row, index the matrix by [row].

Simultaneous polynomials can have extra variables that have no values, but represent given numeric values that could be substituted later.

You can also include unknown variables that do not appear in the expressions. These zeros show how families of zeros might contain arbitrary constants of the form $@ k$, where $k$ is an integer suffix from 1 through 255. The suffix resets to 1 when you use CIrHome or F1 8:Clear Home.

For polynomial systems, computation time or memory exhaustion may depend strongly on the order in which you list unknowns. If your initial choice exhausts memory or your patience, try rearranging the variables in the expressions and/or varOrGuess list.

If you do not include any guesses and if any expression is non-polynomial in any variable but all expressions are linear in all unknowns, cZeros() uses Gaussian elimination to attempt to determine all zeros.
If a system is neither polynomial in all of its variables nor linear in its unknowns, cZeros() determines at most one zero using an approximate iterative method. To do so, the number of unknowns must equal the number of expressions, and all other variables in the expressions must simplify to numbers.

A non-real guess is often necessary to determine a non-real zero. For convergence, a guess might have to be rather close to a zero.
cZeros ( $\left\{u_{-} * v_{-}-u_{-}-v_{-}, v_{-}{ }^{\wedge} 2+u_{-}\right\}$, $\left.\left\{u_{-}, v_{-}\right\}\right)$ENTER

$$
\left[\begin{array}{ll}
1 / 2-\frac{\sqrt{3}}{2} \cdot \boldsymbol{i} & 1 / 2+\frac{\sqrt{3}}{2} \cdot \boldsymbol{i} \\
1 / 2+\frac{\sqrt{3}}{2} \cdot \boldsymbol{i} & 1 / 2-\frac{\sqrt{3}}{2} \cdot \boldsymbol{i} \\
0 & 0
\end{array}\right]
$$

Extract row 2 :
ans(1)[2] ENTER

$$
\left[\begin{array}{lll}
1 / 2+\frac{\sqrt{3}}{2} \cdot \boldsymbol{i} & 1 / 2-\frac{\sqrt{3}}{2} \cdot \boldsymbol{i}
\end{array}\right]
$$

cZeros( $\left\{u_{-} v_{-} \mathbf{v}_{-} u_{-}-\left(c_{-}{ }^{*} v_{-}\right)\right.$,
$\left.\left.v_{-}{ }^{\wedge} 2+u_{-}\right\},\left\{u_{-}, v_{-}\right\}\right)$ENTER
$\left[\begin{array}{cc}\frac{-\left(\sqrt{1-4 \cdot c^{\prime}}+1\right)^{2}}{4} & \frac{\sqrt{1-4 \cdot c_{-}}+1}{2} \\ \frac{-\left(\sqrt{1-4 \cdot c_{2}}-1\right)^{2}}{4} & \frac{-\left(\sqrt{1-4 \cdot c^{2}}-1\right)}{2}\end{array}\right]$
cZeros ( $\left\{u_{-} * v_{-}-u_{-}-v_{-}, v_{-}{ }^{\wedge} 2+u_{-}\right\}$, $\left.\left\{u_{-}, v_{-}, w_{-}\right\}\right)$ENTER

$$
\left[\begin{array}{lll}
1 / 2-\frac{\sqrt{3}}{2} \cdot \boldsymbol{i} & 1 / 2+\frac{\sqrt{3}}{2} \cdot \boldsymbol{i} & @ 1 \\
1 / 2+\frac{\sqrt{3}}{2} \cdot \boldsymbol{i} & 1 / 2-\frac{\sqrt{3}}{2} \cdot \boldsymbol{i} & \text { @1 } \\
0 & 0 & @ 1
\end{array}\right]
$$

cZeros $\left(\left\{u_{-}+v_{-}-e^{\wedge}\left(w_{-}\right), u_{-}-v_{-}-i\right\}\right.$, $\left.\left\{u_{-}, v_{-}\right\}\right)$ENTER

$$
\left[\begin{array}{ll}
\frac{e^{\mathrm{w}_{-}}}{2}+1 / 2 \cdot \boldsymbol{i} & \frac{e^{\mathrm{w}_{-}}-\boldsymbol{i}}{2}
\end{array}\right]
$$

cZeros( $\left\{e^{\wedge}\left(z_{-}\right)-w_{-}, w_{-}-z_{-}{ }^{\wedge} 2\right\}$, $\left.\left\{w_{-}, z_{-}\right\}\right)$ENTER

$$
\left[\begin{array}{ll}
.494 \ldots . . & -.703 \ldots
\end{array}\right.
$$

$\operatorname{cZeros}\left(\left\{e^{\wedge}\left(z_{-}\right)-w_{-}, w_{-}-z_{-}{ }^{\wedge} 2\right\}\right.$, $\left\{w_{-}, z_{-}=1+i\right\}$ ) ENTER
$\left[\begin{array}{ll}.149 \ldots+4.89 \ldots \cdot \boldsymbol{i} & 1.588 \ldots+1.540 \ldots \cdot \boldsymbol{i}]\end{array}\right.$

| $d()$ | 2nd [d] key or MATH/Calculus menu |  |
| :---: | :---: | :---: |
|  | $d$ (expression1, var [,order]) $\Rightarrow$ expression <br> $d$ (list1, var[,order]) $\Rightarrow$ list <br> $d$ (matrix1,var[,order]) $\Rightarrow$ matrix <br> Returns the first derivative of expression1 with respect to variable var. expression1 can be a list or a matrix. <br> order, if included, must be an integer. If the order is less than zero, the result will be an anti-derivative. <br> $d()$ does not follow the normal evaluation mechanism of fully simplifying its arguments and then applying the function definition to these fully simplified arguments. Instead, $\left.\boldsymbol{d}_{( }\right)$performs the following steps: <br> 1. Simplify the second argument only to the extent that it does not lead to a non-variable. <br> 2. Simplify the first argument only to the extent that it does recall any stored value for the variable determined by step 1. <br> 3. Determine the symbolic derivative of the result of step 2 with respect to the variable from step 1. <br> 4. If the variable from step 1 has a stored value or a value specified by a "with" (I) operator, substitute that value into the result from step 3. | $\begin{array}{lr} \begin{array}{ll} d\left(3 x^{\wedge} 3-x+7, x\right) \text { ENTER } & 9 x^{2}-1 \\ d\left(3 x^{\wedge} 3-x+7, x, 2\right) \text { ENTER } & 18 \cdot x \\ d(f(x) * g(x), x) \text { ENTER } & \\ \quad \frac{d}{d x}(f(x)) \cdot g(x)+\frac{d}{d x}(g(x)) \cdot f(x) \\ d(\sin (f(x)), x) \text { ENTER } \\ \cos (f(x)) \frac{d}{d x}(f(x)) \\ d\left(x^{\wedge} 3, x\right) \mid x=5 \text { ENTER } \\ d\left(d\left(x^{\wedge} 2 * y^{\wedge} 3, x\right), y\right) \text { ENTER } & 6 \cdot y^{2} \cdot x \\ d\left(x^{\wedge} 2, x,-1\right) \text { ENTER } \\ d\left(\left\{x^{\wedge} 2, x^{\wedge} 3, x^{\wedge} 4\right\}, x\right) \text { ENTER } \\ \{2 \cdot x & 3 \cdot x^{2} \end{array} \\ \end{array}$ |
| dayOfWk() CATALOG |  |  |
|  | dayOfWk(year,month,day) $\Rightarrow$ integer <br> Returns an integer from 1 to 7, with each integer representing a day of the week. Use dayOfWk() to determine on which day of the week a particular date would occur. <br> Note: May not give accurate results for years prior to 1583 (pre-Gregorian calendar). <br> Enter the year as a four-digit integer. The month and day can be either one- or two-digit integers. | day0fWk(1948, 9, 6) <br> Integer values: <br> 1 = Sunday <br> 2 = Monday <br> 3 = Tuesday <br> 4 = Wednesday <br> 5 = Thursday <br> $6=$ Friday <br> 7 = Saturday |
| DD | MATH/Angle menu |  |
|  | number $\mathbf{D D D} \Rightarrow$ value <br> list1 DDD $\Rightarrow$ list <br> matrix 1 DD $\Rightarrow$ matrix | In Degree angle mode: <br> $1.5^{\circ}$ DD ENTER $1.5^{\circ}$ |
|  | Returns the decimal equivalent of the argument. The argument is a number, list, or matrix that is interpreted by the Mode setting in radians or degrees. | $\begin{array}{r} 45^{\circ} 22^{\prime} 14.3^{\prime \prime} \Rightarrow \text { DD ENTER } \quad 45.370 \ldots{ }^{\circ} \\ \left\{45^{\circ} 22^{\prime} 14.3^{\prime \prime}, 60^{\circ} 0 \cdot 0^{\prime \prime}\right\} \operatorname{DD~ENTER} \\ \{45.370 \ldots \quad 60\}^{\circ} \end{array}$ |
|  | Note: \DD can also accept input in radians. | In Radian angle mode: <br> $1.5>$ DD ENTER $85.9^{\circ}$ |
| Dec | MATH/Base menu |  |
|  | integer1>Dec $\Rightarrow$ integer | Ob10011 Dec ENTER 19 |
|  | Converts integer1 to a decimal (base 10) number. A binary or hexadecimal entry must always have a $0 b$ or Oh prefix, respectively. | Oh1F $\mathrm{Dec}^{\text {ENTER }} 31$ |

— Zero, not the letter O, followed by b or h.
Ob binaryNumber
Oh hexadecimalNumber
L A binary number can have up to 32 digits. A hexadecimal number can have up to 8.

Without a prefix, integer1 is treated as decimal. The result is displayed in decimal, regardless of the Base mode.

## Define CATALOG

Define funcName(arg1Name, arg2Name, ...) = expression

Creates funcName as a user-defined function. You then can use funcName(), just as you use built-in functions. The function evaluates expression using the supplied arguments and returns the result.
funcName cannot be the name of a system variable or built-in function.

The argument names are placeholders; you should not use those same names as arguments when you use the function.

Note: This form of Define is equivalent to executing the expression:
expression $\rightarrow$ funcName ( arg1Name,arg2Name).
This command also can be used to define simple variables; for example, Define a=3.

| Define $g(x, y)=2 x-3 y$ ENTER | Done |
| :--- | ---: |
| $g(1,2)$ ENTER | -4 |
| $1 \rightarrow a: 2 \rightarrow b: g(a, b)$ ENTER | -4 |
| Define $h(x)=$ when $(x<2,2 x-3$, |  |
| $-2 x+3)$ ENTER | Done |

$h(-3)$ ENTER -9
$h(4)$ ENTER -5

$$
\begin{aligned}
& \text { Define eigenv1(a)= } \\
& \text { cZeros(det(identity(dim(a) } \\
& \text { [1])-x*a), x) ENTER Done } \\
& \text { eigenv1([-1,2;4,3]) ENTER } \\
& \qquad\left\{\frac{2 \cdot \sqrt{3}-1}{11} \frac{-(2 \cdot \sqrt{3}+1)}{11}\right\}
\end{aligned}
$$

Define funcName(arg1Name, arg2Name, ...) = Func block

## EndFunc

Is identical to the previous form of Define, except that in this form, the user-defined function funcName() can execute a block of multiple statements.
block can be either a single statement or a series of statements separated with the ":" character. block also can include expressions and instructions (such as If, Then, Else, and For). This allows the function funcName() to use the Return instruction to return a specific result.

Note: It is usually easier to author and edit this form of Function in the program editor rather than on the entry line.

Define progName(arg1Name, arg2Name, ...) = Prgm block
EndPrgm
Creates progName as a program or subprogram, but cannot return a result using Return. Can execute a block of multiple statements.
block can be either a single statement or a series of statements separated with the ":" character. block also can include expressions and instructions (such as If, Then, Else, and For) without restrictions.

Note: It is usually easier to author and edit a program block in the Program Editor rather than on the entry line.

## DelFold catalog

DelFold folderName1[, folderName2] [, folderName3] ...
Deletes user-defined folders with the names folderName1, folderName2, etc. An error message is displayed if the folders contain any variables.

Note: You cannot delete the main folder.

## DelVar catalog

DelVar var1[, var2] [, var3] ...
Deletes the specified variables from memory

## deSolve() MATH/Calculus menu

deSolve(1stOr2ndOrderOde, independentVar, dependentVar) $\Rightarrow$ a general solution

Returns an equation that explicitly or implicitly specifies a general solution to the 1st- or 2nd-order ordinary differential equation (ODE). In the ODE:

- Use a prime symbol ( ', press 2nd [']) to denote the 1st derivative of the dependent variable with respect to the independent variable.
- Use two prime symbols to denote the corresponding second derivative.

The ' symbol is used for derivatives within deSolve() only. In other cases, use $\boldsymbol{d}($ ().

The general solution of a 1st-order equation contains an arbitrary constant of the form @ $k$, where $k$ is an integer suffix from 1 through 255 . The suffix resets to 1 when you use CIrHome or F1 8: Clear Home. The solution of a 2nd-order equation contains two such constants.

```
Define listinpt()=prgm:Local
    n,i,str1,num:InputStr "Enter
    name of list",strl:Input "No.
    of elements",n:For
    i,1,n,1:Input "element
    "&string(i),num:
    num->非str1[i]:EndFor:EndPrgm
    ENTER
```

Done
listinpt() ENTEREnter name of list

NewFold games ENTER Done (creates the folder games)
DelFold games ENTER Done
(deletes the folder games)

| $2 \rightarrow$ a ENTER | 2 |
| :--- | ---: |
| $(a+2)^{\wedge} 2$ ENTER | 16 |
| De1Var a ENTER | Done |
| $(a+2)^{\wedge} 2$ ENTER | $(a+2)^{2}$ |

Note: To type a prime symbol ('), press 2nd ['].

$$
\begin{aligned}
& \text { deSolve }\left(y^{\prime}{ }^{\prime}+2 y^{\prime}+y=x^{\wedge} 2, x, y\right) \text { ENTER } \\
& y=(@ 1 \cdot x+@ 2) \cdot e^{-x}+x^{2}-4 \cdot x+6 \\
& \text { right }(\text { ans }(1)) \rightarrow \text { temp ENTER } \\
& (@ 1 \cdot x+@ 2) \cdot e^{-x}+x^{2}-4 \cdot x+6
\end{aligned} \quad \begin{aligned}
& \boldsymbol{d}(\text { temp }, x, 2)+2 * \boldsymbol{d}(\text { temp }, x)+\text { temp }-x^{\wedge} 2 \\
& \text { ENTER }
\end{aligned}
$$

DelVar temp ENTER Done

Apply solve() to an implicit solution if you want to try to convert it to one or more equivalent explicit solutions.

When comparing your results with textbook or manual solutions, be aware that different methods introduce arbitrary constants at different points in the calculation, which may produce different general solutions.
deSolve( $\left.y^{\prime}=(\cos (y))^{\wedge} 2 * x, x, y\right)$
ENTER

$$
\tan (y)=\frac{x^{2}}{2}+@ 3
$$

solve(ans(1),y) ENTER

$$
y=\tan ^{-1}\left(\frac{x^{2}+2 \cdot @ 3}{2}\right)+@ n 1 \cdot \pi
$$

Note: To type an @ symbol, press:

## (2) STO

ans(1)|@3=c-1 and @n1=0 ENTER

$$
y=\tan ^{-1}\left(\frac{x^{2}+2 \cdot(c-1)}{2}\right)
$$

$\sin (y)=\left(y * e^{\wedge}(x)+\cos (y)\right) y^{\prime} \rightarrow o d e$ ENTER

$$
\sin (y)=\left(e^{x} \cdot y+\cos (y)\right) \cdot y^{\prime}
$$

deSolve(ode and
$y(0)=0, x, y) \rightarrow \operatorname{soln}$ ENTER
$\frac{-\left(2 \cdot \sin (y)+y^{2}\right)}{2}=-\left(e^{x}-1\right) \cdot e^{-x} \cdot \sin (y)$
soln $\mid x=0$ and $y=0$ ENTER true
$\boldsymbol{d}(r i g h t(e q)-1 e f t(e q), x) /$
(d(1eft(eq)-right(eq),y))
$\rightarrow$ impdif(eq,x,y) ENTER
ode|y'=impdif(soln, $x, y$ ) ENTER
true
DelVar ode,soln ENTER Done
deSolve( $y^{\prime \prime}=y^{\wedge}(-1 / 2)$ and
$\mathrm{y}(0)=0$ and $\left.\mathrm{y}^{\prime}(0)=0, \mathrm{t}, \mathrm{y}\right)$ ENTER

$$
\frac{2 \cdot y^{3 / 4}}{3}=t
$$

solve(ans(1),y) ENTER

$$
y=\frac{2^{2 / 3} \cdot(3 \cdot t)^{4 / 3}}{4} \text { and } t \geq 0
$$

For initialCondition1, use the form:
dependentVar (initiallndependentValue) = initialDependentValue

For initialCondition2, use the form:
dependentVar' (initiallndependentValue) $=$ initial1stDerivativeValue
deSolve(2ndOrderOde and boundaryCondition1 and
boundaryCondition2, independentVar, dependentVar) $\Rightarrow$ a particular solution

Returns a particular solution that satisfies 2ndOrderOde and has specified values at two different points.
deSolve(w' $-2 w^{\prime} / x+\left(9+2 / x^{\wedge} 2\right) w=$
$x^{*} e^{\wedge}(x)$ and $w(\pi / 6)=0$ and
$w(\pi / 3)=0, x, w)$ ENTER

$$
\begin{array}{r}
\mathrm{w}=\frac{e^{\frac{\pi}{3}} \cdot \mathrm{x} \cdot \cos (3 \cdot \mathrm{x})}{10} \\
-\frac{e^{\frac{\pi}{6} \cdot x \cdot \sin (3 \cdot x)}}{10}+\frac{\mathrm{x} \cdot e^{x}}{10}
\end{array}
$$

## det() MATH/Matrix menu

$\operatorname{det}($ squareMatrix[, tol]) $\Rightarrow$ expression
Returns the determinant of squareMatrix.
Optionally, any matrix element is treated as zero if its absolute value is less than tol. This tolerance is used only if the matrix has floating-point entries and does not contain any symbolic variables that have not been assigned a value. Otherwise, tol is ignored.

- If you use ENTER or set the mode to Exact/Approx=APPROXIMATE, computations are done using floating-point arithmetic.
- If tol is omitted or not used, the default tolerance is calculated as:

```
    det([a,b;c,d]) ENTER a\cdotd-b}
    det([1,2;3,4]) ENTER -2
```

    det(identity(3)-x*[1,-2,3;
    - \(2,4,1 ;-6,-2,7])\) ENTER
        \(-\left(98 \cdot x^{3}-55 \cdot x^{2}+12 \cdot x-1\right)\)
    $[1 \mathrm{E} 20,1 ; 0,1] \rightarrow \operatorname{mat} 1 \quad\left[\begin{array}{ll}1 . \mathrm{E} 20 & 1 \\ 0 & 1\end{array}\right]$
$\operatorname{det}(m a t 1)$ ENTER 0
det(mat1,.1) ENTER
1.E20
$5 \mathrm{E}-14 * \max (\operatorname{dim}($ squareMatrix) $)$

* rowNorm(squareMatrix)


## diag() MATH/Matrix menu

$\operatorname{diag}($ list $) \Rightarrow$ matrix

| diag(rowMatrix) |
| :--- |
| diag(columnMatrix) |$\Rightarrow$ matrix $\Rightarrow$ matrix \(\quad \operatorname{diag(\{ 2,4,6\} )ENTER} \quad\left[\begin{array}{lll}2 \& 0 \& 0 <br>

0 \& 4 \& 0 <br>
0 \& 0 \& 6\end{array}\right]\)

Returns a matrix with the values in the argument list or matrix in its main diagonal.

```
diag(squareMatrix) }=>\mathrm{ rowMatrix
```

Returns a row matrix containing the elements from the main diagonal of squareMatrix.
squareMatrix must be square.
$[4,6,8 ; 1,2,3 ; 5,7,9]$ ENTER $\left[\begin{array}{ccc}4 & 6 & 8 \\ 1 & 2 & 3 \\ 5 & 7 & 9\end{array}\right]$
diag(ans(1)) ENTER
$\left[\begin{array}{lll}4 & 2 & 9\end{array}\right]$

## Dialog CATALOG

## Dialog

block
EndDlog
Generates a dialog box when the program is executed.
block can be either a single statement or a series of statements separated with the ":" character. Valid block options in the F3 I/O, 1:Dialog menu item in the Program Editor are 1:Text, 2:Request, 4:DropDown, and 7:Title.

The variables in a dialog box can be given values that will be displayed as the default (or initial) value. If ENTER is pressed, the variables are updated from the dialog box and variable ok is set to 1 . If ESC is pressed, its variables are not updated, and system variable ok is set to zero.

Program listing:
: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

$\operatorname{dim}(l i s t) \Rightarrow$ integer
Returns the dimension of list.
$\operatorname{dim}$ (matrix) $\Rightarrow$ list $\operatorname{dim}([1,-1,2 ;-2,3,5])$ ENTER $\{23\}$

Returns the dimensions of matrix as a two-element list \{rows, columns\}.
$\operatorname{dim}$ (string) $\Rightarrow$ integer
Returns the number of characters contained in character string string.

Disp CATALOG
Disp [exprOrString1] [, exprOrString2] ...
Displays the current contents of the Program I/O screen. If one or more exprOrString is specified, each expression or character string is displayed on a separate line of the Program I/O screen.

An expression can include conversion operations such as $\boldsymbol{D D}$ and $\stackrel{\text { Rect. You can also use the }}{ }$ operator to perform unit and number base conversions.

If Pretty Print $=$ ON, expressions are displayed in pretty print.

From the Program I/O screen, you can press F5 to display the Home screen, or a program can use DispHome.

DispG CATALOG

DispG
Displays the current contents of the Graph screen.
dim("Hello") ENTER 5
dim("Hello"\&" there") ENTER 11

Disp "Hello" ENTER Hello
Disp $\cos (2.3)$ ENTER $-.666 \ldots$
$\{1,2,3,4\} \rightarrow$ L1 ENTER
Disp L1 ENTER $\left\{\begin{array}{llll}1 & 2 & 3 & 4\end{array}\right\}$
Disp 180_min _hr ENTER 3.•_hr

Note: To type an underscore ( _ ), press:
$\begin{array}{ll}\text { Fin } & {[-]} \\ \text { 2nd [-] }\end{array}$
To type $\downarrow$, press 2nd [ $\downarrow$ ].

In function graphing mode:
Program segment:

: 5* $\cos (x) \rightarrow y 1(x)$
:-10 $\rightarrow$ xmin
: $10 \rightarrow x$ max
$:-5 \rightarrow y m i n$
: $5 \rightarrow$ ymax
: DispG
$\vdots$


## DispHome CATALOG

## DispHome

Displays the current contents of the Home screen.

Program segment:
:Disp "The result is: ",xx
:Pause "Press Enter to quit"
: DispHome
: EndPrgm

## DispTbl CATALOG

## DispTb

Displays the current contents of the Table screen.
Note: The cursor pad is active for scrolling. Press ESC or ENTER to resume execution if in a program.

5* $\cos (x) \rightarrow y 1(x)$ ENTER
DispTbl ENTER


## DMS MATH/Angle menu

## expression $>$ DMS

listDMS
matrix DMS
Interprets the argument as an angle and displays the equivalent DMS (DDDDDD $\left.{ }^{\circ} M M^{\prime} S S . s s^{\prime \prime}\right)$ number. See ${ }^{\circ}$, ', " on page 883 for DMS (degree, minutes, seconds) format.

Note: „DMS will convert from radians to degrees when used in radian mode. If the input is followed by a degree symbol ( ${ }^{\circ}$ ), no conversion will occur. You can use DMS only at the end of an entry line.

## dotP() MATH/Matrix/Vector ops menu

$\operatorname{dot} \mathbf{P}($ list1, list2) $\Rightarrow$ expression
Returns the "dot" product of two lists.
$\operatorname{dotP}($ vector1, vector2) $\Rightarrow$ expression
Returns the "dot" product of two vectors.
Both must be row vectors, or both must be column vectors.

In Degree angle mode:
45.371 DMS ENTER $45^{\circ} 22^{\prime} 15.6^{\prime \prime}$
$\{45.371,60\}$ DMS ENTER

$$
\left\{45^{\circ} 22^{\prime} 15.6^{\prime \prime} 60^{\circ}\right\}
$$

$\operatorname{dotP}(\{a, b, c\},\{d, e, f\})$ ENTER
$a \cdot d+b \cdot e+c \cdot f$
$\operatorname{dot}(\{1,2\},\{5,6\})$ ENTER 17
$\operatorname{dotP}([a, b, c],[d, e, f])$ ENTER $a \cdot d+b \cdot e+c \cdot f$
$\operatorname{dot}([1,2,3],[4,5,6])$ ENTER 32

## DrawFunc CATALOG

DrawFunc expression
Draws expression as a function, using x as the independent variable.

Note: Regraphing erases all drawn items

In function graphing mode and ZoomStd window:

DrawFunc 1.25x* cos(x) ENTER


## Drawlnv catalog

Drawlnv expression
Draws the inverse of expression by plotting $x$ values on the $y$ axis and $y$ values on the $x$ axis.
$x$ is the independent variable.
Note: Regraphing erases all drawn items.

In function graphing mode and ZoomStd window:

DrawInv 1.25x* cos (x) ENTER


DrawParm expression1, expression2
[, tmin] [, tmax] [, tstep]
Draws the parametric equations expression1 and expression2, using $t$ as the independent variable.

Defaults for tmin, tmax, and tstep are the current settings for the Window variables tmin, tmax, and tstep. Specifying values does not alter the window settings. If the current graphing mode is not parametric, these three arguments are required.

Note: Regraphing erases all drawn items.

## DrawPol CATALOG

DrawPol expression[, $\theta$ min] [, $\theta$ max] [, $\theta$ step]
Draws the polar graph of expression, using $\theta$ as the independent variable.

Defaults for $\theta$ min, $\theta$ max, and $\theta$ step are the current settings for the Window variables $\theta$ min, $\theta$ max, and $\theta$ step. Specifying values does not alter the window settings. If the current graphing mode is not polar, these three arguments are required.

Note: Regraphing erases all drawn items.

## DrawSIp CATALOG

DrawSIp x1, y1, slope
Displays the graph and draws a line using the formula $y-y 1=$ slope $\cdot(x-x 1)$.

Note: Regraphing erases all drawn items.

In function graphing mode and ZoomStd window:

DrawParm
t* $\cos (\mathrm{t}), \mathrm{t} * \sin (\mathrm{t}), 0,10, .1$ ENTER


In function graphing mode and ZoomStd window:

DrawPol 5* $\cos (3 * \theta), 0,3.5, .1$ ENTER


In function graphing mode and ZoomStd window:

DrawS1p 2,3,-2 ENTER


## DropDown catalog

DropDown titleString, \{item1String, item2String, ...\}, varName

Displays a drop-down menu with the name titleString and containing the items 1:item1String, 2: item2String, and so forth. DropDown must be within a Dialog...EndDlog block.

If varName already exists and has a value within the range of items, the referenced item is displayed as the default selection. Otherwise, the menu's first item is the default selection.

When you select an item from the menu, the corresponding number of the item is stored in the variable varName. (If necessary, DropDown creates varName.)

See Dialog program listing example.

## DrwCtour expression

DrwCtour list
Draws contours on the current 3D graph at the $z$ values specified by expression or list. The 3D graphing mode must already be set. DrwCtour automatically sets the graph format style to CONTOUR LEVELS.

By default, the graph automatically contains the number of equally spaced contours specified by the ncontour Window variable. DrwCtour draws contours in addition to the defaults.

To turn off the default contours, set ncontour to zero, either by using the Window screen or by storing 0 to the ncontour system variable.

In 3D graphing mode:
$(1 / 5) x^{\wedge} 2+(1 / 5) y^{\wedge} 2-10 \rightarrow z 1(x, y)$
ENTER

Done
$-10 \rightarrow x \min : 10 \rightarrow x \max$ ENTER 10
$-10 \rightarrow y m i n: 10 \rightarrow y m a x$ ENTER 10
$-10 \rightarrow$ zmin: $10 \rightarrow$ zmax ENTER 10
$0 \rightarrow$ ncontour ENTER
DrwCtour $\{-9,-4.5,-3,0,4.5,9\}$
ENTER


- Use the cursor to change the viewing angle. Press 0 (zero) to return to the original view.

To toggle between different graph format styles, press:


- Press X, Y, or Z to look down the corresponding axis

| E |  |  |  |
| :---: | :---: | :---: | :---: |
|  | mantissaEexponent | 2.3 E 4 ENTER | 23000. |
|  | Enters a number in scientific notation. The number is interpreted as mantissa $\times 10$ exponent. | $2.3 \mathrm{E} 9+4.1 \mathrm{E} 15$ ENTER | 4.1 e 15 |
|  | Hint: If you want to enter a power of 10 without causing a decimal value result, use $10^{\wedge}$ integer. | $3 * 10 \wedge 4$ ENTER | 30000 |
| $e^{\wedge}()$ |  |  |  |
|  | $e^{\wedge}$ (expression1) $\Rightarrow$ expression | $e^{\wedge}(1)$ ENTER | $e$ |
|  | Returns e raised to the expression1 power. | $e^{\wedge}$ (1.) ENTER | 2.718... |
|  | Note: On the TI-89 Titanium, pressing $\left[\mathrm{e}^{x}\right]$ to display $\mathrm{e}^{\wedge}$ ( is different from pressing alphal [E]. On the Voyage 200, pressing 2nd [e ${ }^{x}$ ] to display $\mathrm{e}^{\wedge}$ is different from accessing the character e from the QWERTY keyboard. | $e^{\wedge}(3) \wedge 2$ ENTER | $e^{9}$ |
|  | You can enter a complex number in re ${ }^{i \theta}$ polar form. However, use this form in Radian angle mode only; it causes a Domain error in Degree angle mode. |  |  |

$e^{\wedge}($ list 1$) \Rightarrow$ list $\quad e^{\wedge}(\{1,1 ., 0, .5\})$ ENTER
Returns e raised to the power of each element in list1.
$e^{\wedge}$ (squareMatrix1) $\Rightarrow$ squareMatrix
Returns the matrix exponential of squareMatrix1. This is not the same as calculating e raised to the power of each element. For information about the calculation method, refer to $\boldsymbol{\operatorname { c o s } ( )}$ ).
squareMatrix1 must be diagonalizable. The result always contains floating-point numbers.
\{e 2.718... 1 1.648... $\}$
$e^{\wedge}([1,5,3 ; 4,2,1 ; 6,-2,1])$ ENTER
$\left[\begin{array}{lll}782.209 & 559.617 & 456.509 \\ 680.546 & 488.795 & 396.521 \\ 524.929 & 371.222 & 307.879\end{array}\right]$

| eigVc() | MATH/Matrix menu |  |
| :---: | :---: | :---: |
| eigV | squareMatrix) $\Rightarrow$ matrix <br> Returns a matrix containing the eigenvectors for a real or complex squareMatrix, where each column in the result corresponds to an eigenvalue. Note that an eigenvector is not unique; it may be scaled by any constant factor. The eigenvectors are normalized, meaning that if $\mathrm{V}=\left[\mathrm{x}_{1}, \mathrm{x}_{2}, \ldots, \mathrm{x}_{\mathrm{n}}\right]$, then: $\sqrt{x_{1}^{2}+x_{2}^{2}+\ldots+x_{n}^{2}}=1$ <br> squareMatrix is first balanced with similarity transformations until the row and column norms are as close to the same value as possible. The squareMatrix is then reduced to upper Hessenberg form and the eigenvectors are computed via a Schur factorization. | In Rectangular complex format mode: |
| eigVI() | MATH/Matrix menu |  |
| eigVI | quareMatrix) $\Rightarrow$ list <br> Returns a list of the eigenvalues of a real or complex squareMatrix. <br> squareMatrix is first balanced with similarity transformations until the row and column norms are as close to the same value as possible. The squareMatrix is then reduced to upper Hessenberg form and the eigenvalues are computed from the upper Hessenberg matrix. | In Rectangular complex format mode: |
| Else | See If, page 805. |  |
| Elself | CATALOG See also If, page 805. |  |
|  | an expression1 Then <br> k1 <br> Boolean expression2 Then <br> k2 <br> Boolean expressionN Then <br> kN <br> Elself can be used as a program instruction for program branching. | ```Program segment: :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 | See Custom, page 779. |  |
| EndDlog | See Dialog, page 786. |  |
| EndFor | See For, page 798. |  |
| EndFunc | See Func, page 799. |  |
| Endlf | See If, page 805. |  |

EndLoop See Loop, page 814.

## EndPrgm See Prgm, page 828.

## EndTBar See ToolBar, page 863.

## EndTry See Try, page 864.

## EndWhile See While, page 866.

## entry() CATALOG

entry() $\Rightarrow$ expression
entry(integer) $\Rightarrow$ expression
Returns a previous entry-line entry from the Home screen history area.
integer, if included, specifies which entry expression in the history area. The default is 1 , the most recently evaluated entry. Valid range is from 1 to 99 and cannot be an expression.

Note: If the last entry is still highlighted on the Home screen, pressing ENTER is equivalent to executing entry(1)

On the Home screen:
$1+1 / x$ 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 menu

exact(expression1 [, tol]) $\Rightarrow$ expression
exact(list 1 [, tol]) $\Rightarrow$ list
exact(matrix $1[$ tof $]) \Rightarrow$ matrix
Uses Exact mode arithmetic regardless of the Exact/Approx mode setting to return, when possible, the rational-number equivalent of the argument.
tol specifies the tolerance for the conversion; the default is 0 (zero).
exact(.25) ENTER
1/4
exact(.333333) ENTER $\quad \frac{333333}{1000000}$
exact(.33333,.001)
$1 / 3$
exact $(3.5 x+y)$ ENTER $\quad \frac{7 \cdot x}{2}+y$
exact(\{.2,.33,4.125\}) ENTER
$\left\{1 / 5 \frac{33}{100} 33 / 8\right\}$

Exec string [, expression1] [, expression2] ...
Executes a string consisting of a series of Motorola 68000 op-codes. These codes act as a form of an assembly-language program. If needed, the optional expressions let you pass one or more arguments to the program.

For more information, check the TI Web site: http://www.ti.com/calc

Warning: Exec gives you access to the full power of the microprocessor. Please be aware that you can easily make a mistake that locks up the calculator and causes you to lose your data. We suggest you make a backup of the calculator contents before attempting to use the Exec command.

## Exit CATALOG

## Exit

Exits the current For, While, or Loop block.
Exit is not allowed outside the three looping structures (For, While, or Loop).

Program listing:
$: 0 \rightarrow$ temp
$:$ For i,1,100,1
: temp $i \rightarrow$ temp
$:$ If temp $>20$
$:$ Exit
: EndFor
:Disp temp
Contents of temp after execution:
:For i,1,100,1
: temp+i $\rightarrow$ temp
: If temp>20
: Exit
: EndFor
Contents of temp after execution:

## exp>list() CATALOG

exprlist(expression,var) $\Rightarrow$ list
Examines expression for equations that are separated by the word "or," and returns a list containing the right-hand sides of the equations of the form var=expression. This gives you an easy way to extract some solution values embedded in the results of the solve(), cSolve(), $\mathbf{f M i n}()$, and $\mathbf{f M a x}()$ functions.

Note: exprlist() is not necessary with the zeros and cZeros() functions because they return a list of solution values directly.

## expand() MATH/Algebra menu

expand(expression1[,var]) $\Rightarrow$ expression
expand(list1 [,var]) $\Rightarrow$ list a sum and/or difference of simple terms. In contrast, the goal of factor() is to transform expressiont into a product and/or quotient of simple factors.
expand(matrix $[, v a r]) \Rightarrow$ matrix
expand(expression1) returns expression1 expanded with respect to all its variables. The expansion is polynomial expansion for polynomials and partial fraction expansion for rational expressions.

The goal of expand() is to transform expression 1 into

```
solve(x^2-x-2=0,x) ENTER }x=2\mathrm{ or
x=-1
exp\1ist(solve(x^2-x-2=0,x),x)
ENTER
```

expand ( $\left.(x+y+1)^{\wedge} 2\right)$ ENTER

$$
x^{2}+2 \cdot x \cdot y+2 \cdot x+y^{2}+2 \cdot y+1
$$

expand ( ( $\left.x^{\wedge} 2-x+y^{\wedge} 2-y\right) /\left(x^{\wedge} 2 * y^{\wedge} 2\right.$
$\left.\left.-x^{\wedge} 2 * y-x * y^{\wedge} 2+x^{*} y\right)\right)$ ENTER
$-\exp \operatorname{sind}\left[\frac{x^{2}-x+y^{2}-y}{x^{2} \cdot y^{2}-x^{2} \cdot y-x \cdot y}\right]$
expand(expression1, var) returns expression expanded with respect to var. Similar powers of var are collected. The terms and their factors are sorted with var as the main variable. There might be some incidental factoring or expansion of the collected coefficients. Compared to omitting var, this often saves time, memory, and screen space, while making the expression more comprehensible.

Even when there is only one variable, using var might make the denominator factorization used for partial fraction expansion more complete.

Hint: For rational expressions, propFrac() is a faster but less extreme alternative to expand().
Note: See also comDenom() for an expanded numerator over an expanded denominator.
expand(expression1,[var]) also distributes logarithms and fractional powers regardless of var. For increased distribution of logarithms and fractional powers, inequality constraints might be necessary to guarantee that some factors are nonnegative.
expand(expression1, [var]) also distributes absolute values, sign(), and exponentials, regardless of var.
Note: See also tExpand() for trigonometric anglesum and multiple-angle expansion.
expand $\left((x+y+1)^{\wedge} 2, y\right)$ ENTER

$$
y^{2}+2 \cdot y \cdot(x+1)+(x+1)^{2}
$$

expand $\left((x+y+1)^{\wedge} 2, x\right)$ ENTER

$$
x^{2}+2 \cdot x \cdot(y+1)+(y+1)^{2}
$$

expand ( ( $\left.x^{\wedge} 2-x+y^{\wedge} 2-y\right) /\left(x^{\wedge} 2 * y^{\wedge} 2\right.$
$\left.\left.-x^{\wedge} 2 * y-x * y^{\wedge} 2+x * y\right), y\right)$ ENTER

| - Exp:ands |
| ---: |
| $\frac{x^{2}-x+y^{2}-y}{x^{2} \cdot y^{2}-x^{2} \cdot y-x \cdot y}$ |
| $\frac{1}{y-1}-\frac{1}{y}+\frac{1}{x \cdot(x-1)}$ |

expand(ans(1),x) ENTER

| - expand $\left(\frac{1}{y-1}-\frac{1}{y}+\frac{1}{x \cdot(x-1}\right.$ |
| ---: |
| $\frac{1}{x-1}-\frac{1}{x}+\frac{1}{y \cdot(y-1)}$ |

$\operatorname{expand}\left(\left(x^{\wedge} 3+x^{\wedge} 2-2\right) /\left(x^{\wedge} 2-2\right)\right)$
ENTER

$$
\frac{2 \cdot x}{x^{2}-2}+x+1
$$

expand(ans(1),x) ENTER

$$
\frac{1}{x-\sqrt{2}}+\frac{1}{x+\sqrt{2}}+x+1
$$

$\ln (2 x * y)+\sqrt{ }(2 x * y)$ 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)) $|y\rangle=0$ ENTER

$$
\ln (x)+\sqrt{2} \cdot \sqrt{x} \cdot \sqrt{y}+\ln (y)+\ln (2)
$$

$\operatorname{sign}(x * y)+a b s(x * y)+e^{\wedge}(2 x+y)$
ENTER

$$
e^{2 \cdot x+y}+\operatorname{sign}(x \cdot y)+|x \cdot y|
$$

expand(ans(1)) ENTER

$$
\operatorname{sign}(x) \cdot \operatorname{sign}(y)+|x| \cdot|y|+\left(e^{x}\right)^{2} \cdot e^{y}
$$

## expr() MATH/String menu

expr(string) $\Rightarrow$ expression
Returns the character string contained in string as an expression and immediately executes it.
$\operatorname{expr}\left(" 1+2+x^{\wedge} 2+x\right.$ ") ENTER $x^{2}+x+3$
expr("expand((1+x)^2)") ENTER $x^{2}+2 \cdot x+1$
"Define cube(x)=x^3" $\rightarrow$ funcstr ENTER
"Define cube(x)=x^3"
expr(funcstr) ENTER Done
cube (2) ENTER
8

## ExpReg list1, list2 [, [list3] [, list4, list5]]

Calculates the exponential regression and updates all the system statistics variables.

All the lists must have equal dimensions except for list5.
list1 represents xlist.
list2 represents ylist. list3 represents frequency.
list4 represents category codes.
list5 represents category include list.
Note: list1 through list4 must be a variable name or c1-c99 (columns in the last data variable shown in the Data/Matrix Editor). list5 does not have to be a variable name and cannot be c1-c99.

In function graphing mode:
$\{1,2,3,4,5,6,7,8\} \rightarrow$ L1 ENTER
$\{1,2,2,2,3,4,5,7\} \rightarrow$ L2 ENTER
$\{12 \ldots\}$
Done
ExpReg L1,L2 ENTER
ShowStat ENTER


ENTER
Regeq $(x) \rightarrow y 1(x)$ ENTER Done
NewPlot 1,1,L1,L2 ENTER Done
[GRAPH]


## factor() MATH/Algebra menu

factor(expression1[, var]) $\Rightarrow$ expression
factor(list1[,var]) $\Rightarrow$ list
factor(matrix $1[$,var]) $\Rightarrow$ matrix
factor(expression1) returns expression1 factored with respect to all of its variables over a common denominator.
expression 1 is factored as much as possible toward linear rational factors without introducing new nonreal subexpressions. This alternative is appropriate if you want factorization with respect to more than one variable.
factor $\left(a^{\wedge} 3 * x^{\wedge} 2-a * x^{\wedge} 2-a^{\wedge} 3+a\right)$
ENTER

| $a \cdot(a-1) \cdot(a+1) \cdot(x-1) \cdot(x+1)$ |  |
| :--- | ---: |
| factor $\left(x^{\wedge} 2+1\right)$ ENTER | $x^{2}+1$ |
| factor $\left(x^{\wedge} 2-4\right)$ ENTER $(x-2) \cdot(x+2)$ |  |
| factor $\left(x^{\wedge} 2-3\right)$ ENTER | $x^{2}-3$ |
| factor $\left(x^{\wedge} 2-a\right)$ ENTER | $x^{2}-a$ |

factor(expression1,var) returns expression1 factored with respect to variable var.
expression 1 is factored as much as possible toward real factors that are linear in var, even if it introduces irrational constants or subexpressions that are irrational in other variables.

The factors and their terms are sorted with var as the main variable. Similar powers of var are collected in each factor. Include var if factorization is needed with respect to only that variable and you are willing to accept irrational expressions in any other variables to increase factorization with respect to var. There might be some incidental factoring with respect to other variables.
factor $\left(a^{\wedge} 3 * x^{\wedge} 2-a * x^{\wedge} 2-a^{\wedge} 3+a, x\right)$
ENTER

$$
a \cdot\left(a^{2}-1\right) \cdot(x-1) \cdot(x+1)
$$

factor ( $\left.x^{\wedge} 2-3, x\right)$ ENTER

$$
(x+\sqrt{3}) \cdot(x-\sqrt{3})
$$

factor( $\left.x^{\wedge} 2-a, x\right)$ ENTER

$$
(x+\sqrt{a}) \cdot(x-\sqrt{a})
$$

For the AUTO setting of the Exact/Approx mode, including var permits approximation with floatingpoint coefficients where irrational coefficients cannot be explicitly expressed concisely in terms of the built-in functions. Even when there is only one variable, including var might yield more complete factorization

Note: See also comDenom() for a fast way to achieve partial factoring when factor() is not fast enough or if it exhausts memory.

Note: See also cFactor() for factoring all the way to complex coefficients in pursuit of linear factors.
factor(rationalNumber) returns the rational number factored into primes. For composite numbers, the computing time grows exponentially with the number of digits in the second-largest factor. For example, factoring a 30-digit integer could take more than a day, and factoring a 100-digit number could take more than a century.

Note: To stop (break) a computation, press 0 N .
If you merely want to determine if a number is prime, use isPrime() instead. It is much faster, particularly if rationalNumber is not prime and if the second-largest factor has more than five digits.

Fill MATH/Matrix menu
Fill expression, matrixVar $\Rightarrow$ matrix
Replaces each element in variable matrixVar with expression.
matrixVar must already exist
factor(152417172689) ENTER
$123457 \cdot 1234577$
isPrime(152417172689) EENTERfalse
$[1,2 ; 3,4] \rightarrow \operatorname{amatrx}$ ENTER $\quad\left[\begin{array}{ll}1 & 2 \\ 3 & 4\end{array}\right]$
Fil1 1.01,amatrx ENTER Done amatrx ENTER $\left[\begin{array}{ll}1.01 & 1.01 \\ 1.01 & 1.01\end{array}\right]$

Fill expression, listVar $\Rightarrow$ list
Replaces each element in variable listVar with expression.
listVar must already exist.

## $\{1,2,3,4,5\} \rightarrow$ alist ENTER

$\left\{\begin{array}{lllll}1 & 2 & 3 & 4 & 5\end{array}\right\}$
Fil1 1.01,alist ENTER Done alist ENTER
$\{1.011 .011 .011 .011 .01\}$

## floor() MATH/Number menu

floor(expression) $\Rightarrow$ integer
floor(-2.14) ENTER

- 3. 

Returns the greatest integer that is $\leq$ the argument. This function is identical to int().

The argument can be a real or a complex number.

```
floor(list1) \(\Rightarrow\) list
floor (\{3/2,0,-5.3\}) ENTER
floor(matrix1) \(\Rightarrow\) matrix
```

Returns a list or matrix of the floor of each element.
Note: See also ceiling() and int().
floor([1.2,3.4;2.5,4.8]) ENTER
$\left[\begin{array}{ll}1.3 . \\ 2 . & 4 .\end{array}\right]$

## fMax() MATH/Calculus menu

fMax(expression, var) $\Rightarrow$ Boolean expression
Returns a Boolean expression specifying candidate values of var that maximize expression or locate its least upper bound.
factor $\left(x^{\wedge} 5+4 x^{\wedge} 4+5 x^{\wedge} 3-6 x-3\right)$
ENTER

$$
\begin{array}{r}
x^{5}+4 \cdot x^{4}+5 \cdot x^{3}-6 \cdot x-3 \\
\text { factor }(\operatorname{ans}(1), x) \text { ENTER } \\
(x-.964 \ldots) \cdot(x+.611 \ldots) \cdot \\
(x+2.125 \ldots) \cdot\left(x^{2}+2.227 \ldots \cdot\right. \\
x+2.392 \ldots)
\end{array}
$$

Use the "l" operator to restrict the solution interval and/or specify the sign of other undefined variables.

For the APPROX setting of the Exact/Approx mode, fMax() iteratively searches for one approximate local maximum. This is often faster, particularly if you use the "I" operator to constrain the search to a relatively small interval that contains exactly one local maximum.

Note: See also fMin() and $\max ()$.

## fMin() MATH/Calculus menu

$\mathbf{f M i n}$ (expression, var) $\Rightarrow$ Boolean expression
Returns a Boolean expression specifying candidate values of var that minimize expression or locate its greatest lower bound.

Use the "" operator to restrict the solution interval and/or specify the sign of other undefined variables.
For the APPROX setting of the Exact/Approx mode, fMin() iteratively searches for one approximate local minimum. This is often faster, particularly if you use the "|" operator to constrain the search to a relatively small interval that contains exactly one local minimum.

Note: See also $\mathbf{f M a x}()$ and $\boldsymbol{\operatorname { m i n } ( ) .}$

## FnOff CATALOG

## FnOff

Deselects all $\mathrm{Y}=$ functions for the current graphing mode.

In split-screen, two-graph mode, FnOff only applies to the active graph.

## FnOff [1] [, 2] ... [,99]

Deselects the specified $Y=$ functions for the current graphing mode.
fMax(.5x^3-x-2,x)|x<1 ENTER $x=-.816 \ldots$
fMax (a* $\left.x^{\wedge} 2, x\right)$ ENTER
$x=\infty$ or $x=-\infty$ or $x=0$ or $a=0$
fMax $\left(a * x^{\wedge} 2, x\right) \mid a<0$ ENTER $\quad x=0$

$$
\begin{aligned}
& \text { fMin }\left(1-(x-a)^{\wedge} 2-(x-b)^{\wedge} 2, x\right) \text { ENTER } \\
& x=\infty \text { or } x=-\infty
\end{aligned} \quad \begin{array}{r}
\text { fMin(. } \left.5 x^{\wedge} 3-x-2, x\right) \mid x \geq 1 \text { ENTER } x=1 \\
\text { fMin }\left(a * x^{\wedge} 2, x\right) \text { ENTER } \\
\quad x=\infty \text { or } x=-\infty \text { or } x=0 \text { or } a=0 \\
\text { fMin }\left(a * x^{\wedge} 2, x\right) \mid a>0 \text { and } x>1 \begin{array}{r}
\text { ENTER } \\
x=1 .
\end{array} \\
\text { fMin }\left(a * x^{\wedge} 2, x\right) \mid a>0 \text { ENTER }
\end{array}
$$

In function graphing mode:
FnOff 1,3 ENTER deselects $y 1(x)$ and y3(x).
In parametric graphing mode:
FnOff 1,3 ENTER deselects $x t 1(t)$, $\mathrm{yt1}(\mathrm{t})$, $x t 3(t)$, and $y t 3(t)$.

## FnOn catalog

FnOn
Selects all $Y=$ functions that are defined for the current graphing mode.

In split-screen, two-graph mode, FnOn only applies to the active graph.

## FnOn [1] [, 2] ... [,99]

Selects the specified $Y=$ functions for the current graphing mode.
Note: In 3D graphing mode, only one function at a time can be selected. FnOn 2 selects $\mathrm{z} 2(\mathrm{x}, \mathrm{y})$ and deselects any previously selected function. In the other graph modes, previously selected functions are not affected.

| For | CATALOG |  |
| :---: | :---: | :---: |
| For var, low, high [, step] block <br> EndFor |  | ```Program segment: :0->tempsum : 1 }->\mathrm{ step :For i,1,100,step : tempsum+i->tempsum :EndFor :Disp tempsum``` |
|  | Executes the statements in block iteratively for each value of var, from low to high, in increments of step. var must not be a system variable. step can be positive or negative. The default value is 1. |  |
|  | block can be either a single statement or a series of statements separated with the ":" character. | Contents of tempsum after execution: $5050$ |
|  |  | Contents of tempsum when step is changed to 2 : |
| format() MATH/String menu |  |  |
|  | (expression[, formatString]) $\Rightarrow$ string <br> Returns expression as a character string based on the format template. <br> expression must simplify to a number. formatString is a string and must be in the form: "F[n]", "S $[n]$ ", "E[n]", "G[n][C]", where [ ] indicate optional portions. <br> $F[n]$ : Fixed format. $n$ is the number of digits to display after the decimal point. <br> $S[n]$ : Scientific format. $n$ is the number of digits to display after the decimal point. <br> $\mathrm{E}[n]$ : Engineering format. $n$ is the number of digits after the first significant digit. The exponent is adjusted to a multiple of three, and the decimal point is moved to the right by zero, one, or two digits. <br> $\mathrm{G}[n][\mathrm{C}]$ : Same as fixed format but also separates digits to the left of the radix into groups of three. $c$ specifies the group separator character and defaults to a comma. If $c$ is a period, the radix will be shown as a comma. <br> [RC]: Any of the above specifiers may be suffixed with the Rc radix flag, where $c$ is a single character that specifies what to substitute for the radix point. |  |
| fPart() | MATH/Number menu |  |
| $\begin{aligned} & \text { fPart(expression1) } \Rightarrow \text { expression } \\ & \text { fPart(list1) } \Rightarrow \text { list } \\ & \mathrm{fPart}(\text { matrix1 }) \Rightarrow \text { matrix } \end{aligned}$ |  | $\begin{aligned} & \text { fPart }(-1.234) \text { ENTER }-.234 \\ & \text { fPart }(\{1,-2.3,7.003\}) \text { ENTER } \end{aligned}$ |
|  | Returns the fractional part of the argument. | fPart (\{1, -2.3, 7.003\}) ENTER |
|  | For a list or matrix, returns the fractional parts of the elements. |  |
|  | The argument can be a real or a complex number. |  |



GetCalc var $[$ ，port $]$
Retrieves a value from the link port and stores it in variable var on the receiving TI－89 Titanium．

If the port is not specified，or port $=0$ is specified，the Tl－89 Titanium waits for data from either port．

If port $=1$ ，the $\mathrm{T}-89$ Titanium waits for data from the USB port．

If port $=2$ ，the T －-89 Titanium waits for data from the I／O port．

## getConfg（）CATALOG

```
getConfg() = ListPairs
```

Returns a list of calculator attributes．The attribute name is listed first，followed by its value．

圈：
getConfg（）ENTER
\｛＂Product Name＂＂Advanced
Mathematics Software＂
＂Version＂＂2．00，09／25／1999＂
＂Product ID＂＂03－1－4－68＂
＂ID 非＂＂01012 34567 ABCD＂
＂Cert．Rev．非＂ 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 非＂＂01012 34567 ABCD＂
＂Cert．Rev．非＂ 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\}
Note：Your screen may display different attribute values．The Cert．Rev．\＃attribute appears only if you have purchased and installed additional software into the calculator．

## getDate（）CATALOG

getDate（）$\Rightarrow$ list
getDate（）EnTER
$\{2002222\}$
Returns a list giving the date according to the current value of the clock．The list is in $\{$ year，month，day\} format.
getDenom(expression1) $\Rightarrow$ expression
Transforms expression1 into one having a reduced common denominator, and then returns its denominator.
getDenom $((x+2) /(y-3))$ ENTER $y-3$
getDenom(2/7) ENTER 7
getDenom(1/x+( $\left.\left.y^{\wedge} 2+y\right) / y^{\wedge} 2\right)$ ENTER $x \cdot y$

## getDtFmt() CATALOG

## getDtFmt() $\Rightarrow$ integer

Returns an integer representing the date format that is currently set on the device.

Integer values:
$1=M M / D D / Y Y$
$2=D D / M M / Y Y$
$3=M M . D D . Y Y$
$4=D D . M M . Y Y$
5 = YY.MM.DD
$6=M M-D D-Y Y$
7 = DD-MM-YY
$8=Y Y-M M-D D$

## getDtStr() catalog

## getDtStr([integer]) $\Rightarrow$ string

Returns a string of the current date in the current date format. For example, a returned string of 28/09/02 represents the 28th day of September, 2002 (when the date format is set to DD/MM/YY).

If you enter the optional integer that corresponds to a date format, the string returns the current date in the specified format.

Optional integer values:
$1=M M / D D / Y Y$
2 = DD/MM/YY
$3=M M . D D . Y Y$
4 = DD.MM.YY
5 = YY.MM.DD
$6=M M-D D-Y Y$
$7=D D-M M-Y Y$
$8=Y Y-M M-D D$

## getFold() CATALOG

getFold() $\Rightarrow$ nameString
Returns the name of the current folder as a string.
getFold() ENTER
getFold() $\rightarrow 01 \mathrm{dfoldr}$ ENTER oldfoldr ENTER
"main"
"main"
"main"

## getKey() CATALOG

```
    getKey() = integer
```

Returns the key code of the key pressed. Returns 0 if no key is pressed.

The prefix keys (shift $\uparrow$, second function 2nd,
 recognized by themselves; however, they modify the keycodes of the key that follows them. For example: $\triangle \triangle \neq \boxed{\triangle} \neq 2 \mathrm{nd}$.

For a listing of key codes, see Appendix B.

Program listing:
: Disp
: Loop
: getKey() $\rightarrow$ key
: while key=0 getKey () $\rightarrow$ key
: EndWhile
: Disp key
If key = ord("a")
: Stop
: EndLoop
getMode(modeNameString) $\Rightarrow$ string
getMode("ALL") $\Rightarrow$ ListStringPairs
If the argument is a specific mode name, returns a string containing the current setting for that mode.

If the argument is "ALL", returns a list of string pairs containing the settings of all the modes. If you want to restore the mode settings later, you must store the getMode("ALL") result in a variable, and then use setMode() to restore the modes.

For a listing of mode names and possible settings, see setMode().
Note: To set or return information about the Unit System mode, use setUnits() or getUnits() instead of setMode() or getMode().

```
getMode("angle") ENTER "RADIAN"
getMode("graph") ENTER "FUNCTION"
getMode("a11") 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"}
```

Note: Your screen may display different mode settings.

## getNum() MATH/Algebra/Extract menu

getNum(expression1) $\Rightarrow$ expression
Transforms expression 1 into one having a reduced common denominator, and then returns its numerator.
$\operatorname{getNum}((x+2) /(y-3))$ ENTER $x+2$
getNum(2/7) ENTER 2
getNum(1/x+1/y) ENTER $x+y$

## getTime() CATALOG

getTime() $\Rightarrow$ list
Returns a list giving the time according to the current value of the clock. The list is in \{hour,minute,second\} format. The time is returned in the 24 hour format.

## getTmFmt() CATALOG

getTmFmt() $\Rightarrow$ integer
Returns an integer representing the clock time format that is currently set on the device.

Integer values:
$12=12$ hour clock
$24=24$ hour clock

## getTmStr() CATALOG

getTmStr([integer]) $\Rightarrow$ string
Returns a string of the current clock time in the current time format.

If you enter the optional integer that corresponds to a clock time format, the string returns the current time in the specified format.

Optional integer values:

$$
\begin{aligned}
& 12=12 \text { hour clock } \\
& 24=24 \text { hour clock }
\end{aligned}
$$

## getTmZn() $\Rightarrow$ integer

Returns an integer representing the time zone that is currently set on the device.

The returned integer represents the number of minutes the time zone is offset from Greenwich Mean Time (GMT), as established in Greenwich, England. For example, if the time zone is offset from GMT by two hours, the device would return 120 (minutes).

Integers for time zones west of GMT are negative.
Integers for time zones east of GMT are positive.

## getType()

CATALOG

$$
\begin{aligned}
& \text { getType(var) } \Rightarrow \text { string } \\
& \text { Returns a string indicating the data type of variable } \\
& \text { var. } \\
& \text { If var has not been defined, returns the string } \\
& \text { "NONE". }
\end{aligned}
$$

If Greenwich Mean Time is 14:07:07, it is:
8:07:07 a.m. in Denver, Colorado (Mountain Daylight Time)
(-360 minutes from GMT)
16:07:07 p.m. in Brussels, Belgium (Central
European Standard Time)
(+120 minutes from GMT)

| $\{1,2,3\} \rightarrow$ temp ENTER | $\{123\}$ |
| :---: | :---: |
| getType(temp) ENTER | "LIST" |
| $2+3 i \rightarrow$ temp ENTER | $2+3 i$ |
| getType(temp) ENTER | "EXPR" |
| DelVar temp ENTER | Done |
| getType(temp) ENTER | "NONE" |


| Data Type | Variable Contents |
| :--- | :--- |
| "ASM" | Assembly-language program |
| "DATA" | Data type |
| "EXPR" | Expression (includes complex/arbitrary/undefined, $\infty,-\infty$, TRUE, FALSE, pi, e) |
| "FUNC" | Function |
| "GDB" | Graph data base |
| "LIST" | List |
| "MAT" | Matrix |
| "NONE" | Variable does not exist |
| "NUM" | Real number |
| "OTHER" | Miscellaneous data type for future use by software applications |
| "PIC" | Picture |
| "PRGM" | Program |
| "STR" | String |
| "TEXT" | Text type |
| "VAR" | Name of another variable |

## getUnits() CATALOG

## getUnits() $\Rightarrow$ list

Returns a list of strings that contain the current default units for all categories except constants, temperature, amount of substance, luminous intensity, and acceleration. list has the form:
\{"system" "cat1" "unit1" "cat2" "unit2" ...\}
The first string gives the system (SI, ENG/US, or CUSTOM). Subsequent pairs of strings give a category (such as Length) and its default unit (such as _m for meters).

To set the default units, use setUnits().
getUnits() ENTER
\{"SI" "Area" "NONE"
"Capacitance"
"Charge"

Note: Your screen may display different default units.

Goto labelName
Transfers program control to the label labelName.
labelName must be defined in the same program
using a Lbl instruction.

Transfers program control to the label labelName. using a Lbl instruction.

Program segment:

```
\vdots
0-> temp
l i
:Lb1 TOP
: temp+i->temp
: If i<10 Then
        i+1-> i
: Goto TOP
: EndIf
:Disp temp
    \vdots
```


## Graph catalog

Graph expression1[, expression2] [, var1] [, var2]
The Smart Graph feature graphs the requested expressions/ functions using the current graphing mode.

Expressions entered using the Graph or Table commands are assigned increasing function numbers starting with 1 . They can be modified or individually deleted using the edit functions available when the table is displayed by pressing F44 Header. The currently selected $Y=$ functions are ignored.

If you omit an optional var argument, Graph uses the independent variable of the current graphing mode.

Note: Not all optional arguments are valid in all modes because you can never have all four arguments at the same time.

Some valid variations of this instruction are:

| Function graphing | Graph expr, $x$ |
| :--- | :--- |
| Parametric graphing | Graph $x$ Expr, yExpr, $t$ |
| Polar graphing | Graph expr, $\theta$ |
| Sequence graphing | Not allowed. |
| 3D graphing | Graph expr, $x, y$ |
| Diff Equations graphing | Not allowed. |

In function graphing mode and ZoomStd window:

Graph 1.25a* $\cos (a)$, a ENTER


In parametric graphing mode and ZoomStd window:

Graph
time,2cos(time)/time,time ENTER


In 3D graphing mode:
Graph $\left(v^{\wedge} 2-w^{\wedge} 2\right) / 4, v, w$ ENTER


Note: Use CIrGraph to clear these functions, or go to the $Y=$ Editor to re-enable the system $Y=$ functions.

## Hex MATH/Base menu

integer1 H Hex $\Rightarrow$ integer
Converts integer1 to a hexadecimal number. Binary or hexadecimal numbers always have a Ob or Oh prefix, respectively.

Ob111100001111 Hex ENTER OhFOF

- Zero, not the letter O, followed by b or h.

Ob binaryNumber
Oh hexadecimalNumber
L A binary number can have up to 32 digits. A hexadecimal number can have up to 8.

Without a prefix, integer1 is treated as decimal (base 10). The result is displayed in hexadecimal, regardless of the Base mode.

If you enter a decimal integer that is too large for a signed, 32-bit binary form, a symmetric modulo operation is used to bring the value into the appropriate range.

## identity() MATH/Matrix menu

identity(expression) $\Rightarrow$ matrix identity (4) ENTER
Returns the identity matrix with a dimension of expression. expression must evaluate to a positive integer.

## If CATALOG

If Boolean expression statement
If Boolean expression Then
$\quad$ block
Endlf

If Boolean expression evaluates to true, executes the single statement statement or the block of statements block before continuing execution.

If Boolean expression evaluates to false, continues execution without executing the statement or block of statements.
block can be either a single statement or a sequence of statements separated with the ":" character.

Program segment:

```
: If}x<
:Disp "x is negative"
-or-
:
:If x<0 Then
: Disp "x is negative"
: abs(x)->x
:EndIf
    \vdots
```

Program segment:
:If $x<0$ Then
: Disp "x is negative"
Else
: Disp "x is positive or zero"
: EndIf

If Boolean expression evaluates to true, executes block1 and then skips block2.

If Boolean expression evaluates to false, skips block1 but executes block2.
block1 and block2 can be a single statement.


InputStr [promptString,] var
Pauses the program, displays promptString on the Program I/O screen, waits for you to enter a response, and stores your response as a string in variable var.

If you omit promptString, "?" is displayed as a prompt.
Note: The difference between Input and InputStr is that InputStr always stores the result as a string so that "" are not required.

Program segment:
引
:InputStr "Enter Your Name",str1
inString() MATH/String menu
inString(srcString, subString[, starf]) $\Rightarrow$ integer inString("Hello there","the")
Returns the character position in string srcString at which the first occurrence of string subString begins.
start, if included, specifies the character position within srcString where the search begins. Default $=1$ (the first character of srcString).

If srcString does not contain subString or start is $>$ the length of srcString, returns zero.

| int() | CATALOG |  |
| :---: | :---: | :---: |
|  | $\begin{aligned} & \text { oression } \underset{\text { onteger }}{\Rightarrow} \Rightarrow \text { int } \Rightarrow \text { list } \\ & \text { atrix1 }) \Rightarrow \text { matrix } \end{aligned}$ <br> Returns the greatest integer that is less than or equal to the argument. This function is identical to floor(). <br> The argument can be a real or a complex number. <br> For a list or matrix, returns the greatest integer of each of the elements. | int (-2.5) ENTER -3. <br> int([-1.234, 0, 0.37]) ENTER $[-2.00$. |
| intDiv() | CATALOG |  |
| intDiv (number1, number2) $\Rightarrow$ integer intDiv(list1, list2) $\Rightarrow$ list <br> intDiv(matrix1, matrix2) $\Rightarrow$ matrix <br> Returns the signed integer part of argument 1 divided by argument 2. <br> For lists and matrices returns the signed integer part of argument 1 divided by argument 2 for each element pair. |  | intDiv( $\{12,-14,-16\},\{5,4,-3\})$ ENTER $\left\{\begin{array}{lll} 2 & -3 & 5 \end{array}\right\}$ |
| integrate See $\int()$, page 880. |  |  |
| iPart() MATH/Number menu |  |  |
| iPart(number) $\Rightarrow$ integer <br> iPart(list1) $\Rightarrow$ list <br> iPart(matrixi) $\Rightarrow$ matrix |  | iPart(-1.234) ENTER <br> iPart(\{3/2,-2.3,7.003\}) ENTER |
| Returns the integer part of the argument. |  |  |
| For lists and matrices, returns the integer part of each element. |  |  |
|  | The argument can be a real or a complex number. |  |

isCIkOn() $\Rightarrow$ true,false
Determines if the clock is ON or OFF. Returns true if the clock is ON. Returns false if the clock is OFF.

## isPrime() MATH/Test menu

isPrime(number) $\Rightarrow$ Boolean constant expression
Returns true or false to indicate if number is a whole number $\geq 2$ that is evenly divisible only by itself and 1.

If number exceeds about 306 digits and has no factors $\leq 1021$, isPrime(number) displays an error message.

If you merely want to determine if number is prime, use isPrime() instead of factor(). It is much faster, particularly if number is not prime and has a secondlargest factor that exceeds about five digits.

## IsPrime(5) ENTER <br> true <br> IsPrime(6) ENTER false

Function to find the next prime after a specified number:

Define nextPrim(n)=Func:Loop: $n+1 \rightarrow n: i f$ isPrime(n):return $n$ : EndLoop:EndFunc ENTER Done
nextPrim(7) ENTER 11

Item itemNameString
Item itemNameString, label
Valid only within a Custom...EndCustm or ToolBar...EndTBar block. Sets up a drop-down menu element to let you paste text to the cursor position (Custom) or branch to a label (ToolBar).

Note: Branching to a label is not allowed within a Custom block.

## Lbl CATALOG

Lbl labelName
Defines a label with the name labelName in the program.

You can use a Goto labelName instruction to transfer program control to the instruction immediately following the label.
labelName must meet the same naming requirements as a variable name.

Icm() MATH/Number menu
Icm(number1, number2) $\Rightarrow$ expression
Icm(list1, list2) $\Rightarrow$ list
$\mathbf{I c m}$ (matrix1, matrix2) $\Rightarrow$ matrix
Returns the least common multiple of the two arguments. The Icm of two fractions is the Icm of their numerators divided by the gcd of their denominators. The Icm of fractional floating-point numbers is their product.

For two lists or matrices, returns the least common multiples of the corresponding elements.

## Program segment:

```
-Lb1 1bl1
```

:InputStr "Enter password",
str1
:If strl $1 \neq$ password
: Goto 1bl1
:Disp "Welcome to ..."
:
$1 \mathrm{~cm}(6,9)$ ENTER
$1 \mathrm{~cm}(\{1 / 3,-14,16\},\{2 / 15,7,5\})$
ENTER
$\{2 / 31480\}$

| left() | MATH/String menu |  |
| :---: | :---: | :---: |
|  | left(sourceString[, num]) $\Rightarrow$ string <br> Returns the leftmost num characters contained in character string sourceString. <br> If you omit num, returns all of sourceString. | left("Hello",2) ENTER "He" |
|  | $\operatorname{left}(\text { list1[, num] }) \Rightarrow \text { list }$ <br> Returns the leftmost num elements contained in list1. If you omit num, returns all of list1. | left(\{1,3,-2,4\},3) ENTER $\left\{\begin{array}{lll} \{1 & 3 & -2 \end{array}\right\}$ |
|  | left(comparison) $\Rightarrow$ expression <br> Returns the left-hand side of an equation or inequality. | 1eft $(x<3)$ ENTER $x$ |
| limit() | MATH/Calculus menu |  |
|  | limit(expression1, var, point[, direction]) $\Rightarrow$ expression <br> $\operatorname{limit}($ list1, var, point[, direction]) $\Rightarrow$ list <br> $\operatorname{limit}$ (matrix1, var, point[, direction]) $\Rightarrow$ matrix <br> Returns the limit requested. <br> direction: negative=from left, positive=from right, otherwise=both. (If omitted, direction defaults to both.) <br> Limits at positive $\infty$ and at negative $\infty$ are always converted to one-sided limits from the finite side. <br> Depending on the circumstances, limit() returns itself or undef when it cannot determine a unique limit. This does not necessarily mean that a unique limit does not exist. undef means that the result is either an unknown number with finite or infinite magnitude, or it is the entire set of such numbers. |  |
|  | limit() uses methods such as L'Hopital's rule, so there are unique limits that it cannot determine. If expression 1 contains undefined variables other than var, you might have to constrain them to obtain a more concise result. <br> Limits can be very sensitive to rounding error. When possible, avoid the APPROX setting of the Exact/Approx mode and approximate numbers when computing limits. Otherwise, limits that should be zero or have infinite magnitude probably will not, and limits that should have finite non-zero magnitude might not. |  |

Line xStart, yStart, xEnd, yEndl, drawMode]
Displays the Graph screen and draws, erases, or inverts a line segment between the window coordinates ( $x$ Start, yStart) and (xEnd, yEnd), including both endpoints.

If drawMode $=1$, draws the line (default).
If drawMode $=0$, turns off the line.
If drawMode $=-1$, turns a line that is on to off or off to on (inverts pixels along the line).

Note: Regraphing erases all drawn items. See also Pxiline.

## LineHorz catalog

## LineHorz y[, drawMode]

Displays the Graph screen and draws, erases, or inverts a horizontal line at window position $y$.
If drawMode = 1 , draws the line (default).
If drawMode $=0$, turns off the line.
If drawMode $=-1$, turns a line that is on to off or off to on (inverts pixels along the line).
Note: Regraphing erases all drawn items. See also PxIHorz.

## LineTan catalog

LineTan expression1, expression2
Displays the Graph screen and draws a line tangent to expression1 at the point specified.
expression 1 is an expression or the name of a function, where $x$ is assumed to be the independent variable, and expression2 is the $x$ value of the point that is tangent.

Note: In the example shown, expression 1 is graphed separately. LineTan does not graph expression1.

In function graphing mode and a ZoomTrig window:

Graph $\cos (x)$
[ HOME
됴 [CALC HOME]
LineTan $\cos (x), \pi / 4$ ENTER


In a ZoomStd window:
LineVert - 2.5 ENTER


Note: Regraphing erases all drawn items. See also PxiVert.

LinReg list1, list2[, [list3] [, list4, list5]]
Calculates the linear regression and updates all the system statistics variables.
All the lists must have equal dimensions except for list5.
list1 represents xlist.
list2 represents ylist. list3 represents frequency. list4 represents category codes. list5 represents category include list.

Note: list1 through list4 must be a variable name or c1-c99 (columns in the last data variable shown in the Data/Matrix Editor). list5 does not have to be a variable name and cannot be c1-c99.

In function graphing mode:
$\{0,1,2,3,4,5,6\} \rightarrow$ L1 ENTER
$\left\{\begin{array}{llll}0 & 1 & 2 & \ldots\end{array}\right\}$
$\{0,2,3,4,3,4,6\} \rightarrow$ L2 ENTER
$\left\{\begin{array}{llll}0 & 2 & 3 & \ldots\end{array}\right\}$
LinReg L1, L2 ENTER
Done
ShowStat ENTER


ENTER

| Regeq $(x) \rightarrow y 1(x)$ ENTER | Done |
| :--- | :--- |
| NewPlot 1,1, L1, L2 ENTER | Done |

- [GRAPH]



## $\Delta$ list() MATH/List menu

list $($ list1 $) \Rightarrow$ list
Returns a list containing the differences between consecutive elements in list1. Each element of list1 is subtracted from the next element of list1. The resulting list is always one element shorter than the original list1.

## list>mat() MATH/List menu

Returns a matrix filled row-by-row with the elements from list.
elementsPerRow, if included, specifies the number of elements per row. Default is the number of elements in list (one row).

If list does not fill the resulting matrix, zeros are added.

| $\ln ()$ | [ 2 2nd [LN] key | ㅍin Lix key |
| :---: | :---: | :---: |

$\ln$ (expression1) $\Rightarrow$ expression
$\ln ($ list1) $\Rightarrow$ list
Returns the natural logarithm of the argument.
For a list, returns the natural logarithms of the elements.
$\Delta 1 \mathrm{ist}(\{20,30,45,70\})$ ENTER
$\{10,15,25\}$
listmat (\{1,2,3\}) ENTER [1 2 3 $]$
1istmat ( $\{1,2,3,4,5\}, 2$ ) ENTER
$\left[\begin{array}{ll}1 & 2 \\ 3 & 4 \\ 5 & 0\end{array}\right]$ $1 \mathrm{n}(2.0)$ ENTER .693...

If complex format mode is REAL:
$\ln (\{-3,1.2,5\})$ ENTER
Error: Non-real result
If complex format mode is RECTANGULAR:
$\ln (\{-3,1.2,5\})$ ENTER
$\{\ln (3)+\pi \cdot \boldsymbol{i} .182 \ldots \quad \ln (5)\}$
$\operatorname{In}$ (squareMatrix1) $\Rightarrow$ squareMatrix
Returns the matrix natural logarithm of squareMatrix1. This is not the same as calculating the natural logarithm of each element. For information about the calculation method, refer to $\boldsymbol{\operatorname { c o s } ( )}$ ) on.
squareMatrix 1 must be diagonalizable. The result always contains floating-point numbers.

In Radian angle mode and Rectangular complex format mode:
$\ln ([1,5,3 ; 4,2,1 ; 6,-2,1])$ ENTER
$\left[\begin{array}{lll}1.831 \ldots+1.734 \ldots \cdot \boldsymbol{i} & .009 \ldots-1.490 \ldots \cdot \boldsymbol{i} & \ldots \\ .448 \ldots-. .725 \ldots \cdot \boldsymbol{i} & 1.064 \ldots+. .623 \cdot \boldsymbol{i} & \ldots \\ -.266 \ldots-2.083 \ldots \cdot . \boldsymbol{i} & 1.124 \ldots+1.790 \ldots \cdot \boldsymbol{i} & \ldots .\end{array}\right]$

## LnReg MATH/Statistics/Regressions menu

LnReg list1, list2[, [list3] [, list4, list5]]
Calculates the logarithmic regression and updates all the system statistics variables.
All the lists must have equal dimensions except for list5.
list1 represents xlist. list2 represents ylist. list3 represents frequency. list4 represents category codes. list5 represents category include list.

Note: list1 through list4 must be a variable name or c1-c99 (columns in the last data variable shown in the Data/Matrix Editor). list5 does not have to be a variable name and cannot be c1-c99.

In function graphing mode:


## ENTER

$\operatorname{Regeq}(x) \rightarrow y 1(x)$ ENTER Done NewPlot 1,1,L1,L2 ENTER Done

- [GRAPH]



## Program listing:

:prgmname()
: Prgm
:Local $x, y$
:Input "Enter x",x
:Input "Enter $y$ ",y
:Disp x*y
: EndPrgm
Note: $x$ and $y$ do not exist after the program executes.

## Lock CATALOG

Lock var1[, var2] ...
Locks the specified variables. This prevents you from accidentally deleting or changing the variable without first using the unlock instruction on that variable.

In the example to the right, the variable L1 is locked and cannot be deleted or modified.

Note: The variables can be unlocked using the Unlock command.
$\{1,2,3,4\} \rightarrow L 1$ ENTER
Lock L1 ENTER
$\{1,2,3,4\}$

DelVar L1 ENTER
Error: Variable is locked or protected
$\log$ (expression1) $\Rightarrow$ expression
$\log ($ list1) $\Rightarrow$ list
Returns the base-10 logarithm of the argument.
For a list, returns the base-10 logs of the elements.
$\log (2.0)$ ENTER . 301...
If complex format mode is REAL:

```
log({-3,1.2,5}) ENTER
    Error: Non-real result
```

If complex format mode is RECTANGULAR:
$\log (\{-3,1.2,5\})$ ENTER
$\left\{\frac{\ln (3)}{\ln (10)}+\frac{\pi}{\ln (10)} \cdot \boldsymbol{i} .079 \ldots \frac{\ln (5)}{\ln (10)}\right\}$

Returns the matrix base-10 logarithm of squareMatrix1. This is not the same as calculating the base-10 logarithm of each element. For information about the calculation method, refer to $\boldsymbol{\operatorname { c o s }}(\mathbf{)}$.
squareMatrix1 must be diagonalizable. The result always contains floating-point numbers.

In Radian angle mode and Rectangular complex format mode:
$\log ([1,5,3 ; 4,2,1 ; 6,-2,1])$ ENTER
$\left[\begin{array}{lll}.795 \ldots+.753 \ldots \cdot . \cdot i & .003 \ldots-. .647 \ldots \cdot i & \ldots \\ .194 \ldots-.315 \ldots \cdot i & .462 \ldots 8+.270 \cdot \boldsymbol{i} & \ldots \\ -.115 \ldots-. .904 \ldots \cdot \boldsymbol{i} & .488 \ldots+.777 \ldots \cdot \boldsymbol{i} & \ldots .\end{array}\right]$

## Logistic MATH/Statistics/Regressions menu

Logistic list1, list2 [, [iterations], [list3] [, list4, list5]]
Calculates the logistic regression and updates all the system statistics variables.

All the lists must have equal dimensions except for list5.
list1 represents xlist. list2 represents ylist. list3 represents frequency. list4 represents category codes. list5 represents category include list.
iterations specifies the maximum number of times a solution will be attempted. If omitted, 64 is used. Typically, larger values result in better accuracy but longer execution times, and vice versa.
Note: list1 through list4 must be a variable name or c1-c99 (columns in the last data variable shown in the Data/Matrix Editor). list5 does not have to be a variable name and cannot be c1-c99.

In function graphing mode:
$\{1,2,3,4,5,6\} \rightarrow$ L1 ENTER $\{122 \ldots \ldots$
$\{1,1.3,2.5,3.5,4.5,4.8\} \rightarrow L 2$
ENTER
$\left\{\begin{array}{lllll}1 & 1.3 & 2.5 & \ldots\end{array}\right\}$
Logistic L1,L2 ENTER Done
ShowStat ENTER


ENTER
$\begin{array}{ll}\text { regeq }(x) \rightarrow y 1(x) \text { ENTER } & \text { Done } \\ \text { NewPlot } 1,1, \text { L1, L2 ENTER } & \text { Done }\end{array}$ - [GRAPH]

F2 9


mat>list(matrix) $\Rightarrow$ list
Returns a list filled with the elements in matrix. The elements are copied from matrix row by row.
matlilist([1,2,3]) ENTER
$\left\{\begin{array}{lll}1 & 2 & 3\end{array}\right\}$
$[1,2,3 ; 4,5,6] \rightarrow$ M1 ENTER
$\left[\begin{array}{lll}1 & 2 & 3 \\ 4 & 5 & 6\end{array}\right]$
matlist(M1) ENTER $\left\{\begin{array}{llllll}1 & 2 & 3 & 4 & 5 & 6\end{array}\right\}$
$\max () \quad$ MATH/List menu
$\max ($ expression1, expression2) $\Rightarrow$ expression
$\max (2.3,1.4)$ ENTER
$\max ($ list1, list2) $\Rightarrow$ list
$\max ($ matrix1, matrix2) $\Rightarrow$ matrix
$\max (\{1,2\},\{-4,3\})$ ENTER

Returns the maximum of the two arguments. If the arguments are two lists or matrices, returns a list or matrix containing the maximum value of each pair of corresponding elements.

```
max(lisf) m expression max({0,1,-7,1.3,.5}) ENTER 1.3
```

Returns the maximum element in list.
$\overline{\max (\text { matrix } 1)} \Rightarrow$ matrix $\max ([1,-3,7 ;-4,0, .3])$ ENTER
Returns a row vector containing the maximum element of each column in matrix1.

Note: See also $\mathbf{f M a x}()$ and $\boldsymbol{\operatorname { m i n }}()$.
mean() MATH/Statistics menu
mean(listf, freqlisf]) $\Rightarrow$ expression
mean $(\{.2,0,1,-.3, .4\})$ ENTER . 26
Returns the mean of the elements in list.
Each freqlist element counts the number of consecutive occurrences of the corresponding element in list.
mean(matrix1[, freqmatrix]) $\Rightarrow$ matrix
Returns a row vector of the means of all the columns in matrix1.

Each freqmatrix element counts the number of consecutive occurrences of the corresponding element in matrix1.

In vector format rectangular mode:
mean([.2,0;-1,3;.4,-.5]) ENTER
[-. 133... . 833...]
mean([1/5,0;-1,3;2/5,-1/2])
ENTER
$\left[\begin{array}{cc}-2 / 15 & 5 / 6]\end{array}\right.$
mean([1,2;3,4;5,6],[5,3;4,1;
6,2]) ENTER
$[47 / 15,11 / 3]$
mean $(\{1,2,3\},\{3,2,1\})$ ENTER $5 / 3$

## median() MATH/Statistics menu

median(list) $\Rightarrow$ expression
median(\{.2,0,1,-.3,.4\}) ENTER .2
Returns the median of the elements in list1.

```
median(matrix1) }=>\mathrm{ matrix
median([.2,0;1,-.3;.4,-.5])
```

Returns a row vector containing the medians of the columns in matrix1.

Note: All entries in the list or matrix must simplify to numbers.

MedMed list1, list2[, [list3] [, list4, list5]]
Calculates the median-median line and updates all the system statistics variables.
All the lists must have equal dimensions except for list5.
list1 represents xlist.
list2 represents ylist. list3 represents frequency.
list4 represents category codes.
list5 represents category include list.
Note: list1 through list4 must be a variable name or c1-c99 (columns in the last data variable shown in the Data/Matrix Editor). list5 does not have to be a variable name and cannot be c1-c99.

In function graphing mode:
$\{0,1,2,3,4,5,6\} \rightarrow \operatorname{L1}$ ENTER $\{012 \ldots\}$ $\{0,2,3,4,3,4,6\} \rightarrow$ L2 ENTER $\left\{0 \begin{array}{lll}0 & 3\end{array} ..\right\}$ MedMed L1,L2 ENTER Done ShowStat ENTER


ENTER
Regeq $(x) \rightarrow y 1(x)$ ENTER Done
NewPlot 1,1,L1,L2 ENTER Done

- [GRAPH]



## mid() MATH/String menu

mid(sourceString, start[, counf]) $\Rightarrow$ string
Returns count characters from character string sourceString, beginning with character number start.

If count is omitted or is greater than the dimension of sourceString, returns all characters from sourceString, beginning with character number start.
count must be $\geq 0$. If count $=0$, returns an empty string.

```
mid("Hello there",2) ENTER
                                    "ello there"
mid("Hello there",7,3) ENTER
    "the"
mid("Hel1o there",1,5) ENTER
"Hel1o"
mid("Hello there",1,0) ENTER
```

$\operatorname{mid}($ sourceList, start [, count]) $\Rightarrow$ list
$\operatorname{mid}(\{9,8,7,6\}, 3)$ ENTER
Returns count elements from sourceList, beginning with element number start.

If count is omitted or is greater than the dimension of sourceList, returns all elements from sourceList, beginning with element number start.
count must be $\geq 0$. If count $=0$, returns an empty list.
$\operatorname{mid}($ sourceStringList, start[, counf]) $\Rightarrow$ list
Returns count strings from the list of strings sourceStringList, beginning with element number start.

## $\min () \quad$ MATH/List menu

$\min$ (expression1, expression2) $\Rightarrow$ expression $\boldsymbol{\operatorname { m i n }}($ list1, list2) $\Rightarrow$ list $\min$ (matrix1, matrix2) $\Rightarrow$ matrix

Returns the minimum of the two arguments. If the arguments are two lists or matrices, returns a list or matrix containing the minimum value of each pair of corresponding elements.
$\min (2.3,1.4)$ ENTER 1.4
$\min (\{1,2\},\{-4,3\})$ ENTER $\{-42\}$

Returns the minimum element of list.

nCr (expression1, expression2) $\Rightarrow$ expression
For integer expression1 and expression2 with expression $1 \geq$ expression $2 \geq 0, \mathrm{nCr}()$ is the number of combinations of expression1 things taken expression2 at a time. (This is also known as a binomial coefficient.) Both arguments can be integers or symbolic expressions.

```
nCr(expression, 0) = 1
nCr(expression, negInteger) => 0
nCr(expression, posinteger) }
        expression • (expression- 1)... (expression-posinteger+1)/
        posInteger!
nCr(expression, nonInteger) => expression!/
    ((expression-nonInteger)! • nonInteger!)
```

nCr (list1, list2) $\Rightarrow$ list

Returns a list of combinations based on the corresponding element pairs in the two lists. The arguments must be the same size list.
$\operatorname{nCr}(z, 3)$
$\frac{z \cdot(z-2) \cdot(z-1)}{6}$
$\operatorname{ans}(1) \mid z=5$ 10
$n C r(z, c)$ $\frac{z!}{c!(z-c)!}$
ans(1)/nPr(z,c)
nCr (matrix1, matrix2) $\Rightarrow$ matrix
Returns a matrix of combinations based on the corresponding element pairs in the two matrices. The arguments must be the same size matrix.

## nDeriv() MATH/Calculus menu

nDeriv(expression1, var[, $h]$ ) $\Rightarrow$ expression
nDeriv(expression1, var, list) $\Rightarrow$ list
nDeriv (list, var $[, h]) \Rightarrow$ list
nDeriv(matrix, var $[, h]) \Rightarrow$ matrix
Returns the numerical derivative as an expression. Uses the central difference quotient formula.
$h$ is the step value. If $h$ is omitted, it defaults to 0.001 .

When using list or matrix, the operation gets mapped across the values in the list or across the matrix elements.

Note: See also avgRC() and $d()$.
nDeriv(cos $(x), x, h)$ ENTER

$$
\frac{-(\cos (x-h)-\cos (x+h))}{2 \cdot h}
$$

1imit(nDeriv(cos(x), $x, h), h, 0)$
ENTER

$$
-\sin (x)
$$

nDeriv $\left(x^{\wedge} 3, x, 0.01\right)$ ENTER

$$
3 . \cdot\left(x^{2}+.000033\right)
$$

nDeriv $(\cos (x), x) \mid x=\pi / 2$ ENTER
nDeriv( $\left.x^{\wedge} 2, x,\{.01, .1\}\right)$ ENTER

$$
\{2 . \cdot x 2 . \cdot x\}
$$

## NewData catalog

NewData dataVar, list1[, list2] [, list3]...
Creates data variable dataVar, where the columns are the lists in order.

Must have at least one list.
list1, list2, ..., listn can be lists as shown, expressions that resolve to lists, or list variable names.

NewData makes the new variable current in the Data/Matrix Editor.

NewData mydata, $\{1,2,3\},\{4,5,6\}$ ENTER

Done
(Go to the Data/Matrix Editor and open the var mydata to display the data variable below.)


NewData dataVar, matrix
Creates data variable dataVar based on matrix.

NewData sysData, matrix
Loads the contents of matrix into the system data variable sysData.

## NewFold CATALOG

NewFold folderName
NewFold games ENTER
Done
Creates a user-defined folder with the name folderName, and then sets the current folder to that folder. After you execute this instruction, you are in the new folder.

## newList() CATALOG

newList(numElements) $\Rightarrow$ list
newList(4) ENTER
$\left\{\begin{array}{llll}0 & 0 & 0 & 0\end{array}\right\}$
Returns a list with a dimension of numElements.
Each element is zero.

## newMat() CATALOG also Math/Matrix menu

newMat(numRows, numColumns) $\Rightarrow$ matrix
Returns a matrix of zeros with the dimension numRows by numColumns.

## NewPic CATALOG

NewPic matrix, picVar [, maxRow][, maxCol]
Creates a pic variable picVar based on matrix. matrix must be an $n \times 2$ matrix in which each row represents a pixel. Pixel coordinates start at 0,0. If picVar already exists, NewPic replaces it.

The default for picVar is the minimum area required for the matrix values. The optional arguments, maxRow and maxCol, determine the maximum boundary limits for picVar.

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 $n$, type, xList [,[yLisf], [frqLisf], [catLisf], [includeCatList], [mark] [, bucketSize]] <br> Creates a new plot definition for plot number $n$. <br> type specifies the type of the graph plot. <br> 1 = scatter plot <br> 2 = xyline plot <br> 3 = box plot <br> 4 = histogram <br> 5 = modified box plot <br> mark specifies the display type of the mark. <br> $1=\square$ (box) <br> $2=x$ (cross) <br> $3=+$ (plus ) <br> $4=-$ (square) <br> 5 = •(dot) <br> bucketSize is the width of each histogram "bucket" (type $=4$ ), and will vary based on the window variables xmin and xmax. bucketSize must be $>0$. Default $=1$. <br> Note: $n$ can be 1-9. Lists must be variable names or c1-c99 (columns in the last data variable shown in the Data/Matrix Editor), except for includeCatList, which does not have to be a variable name and cannot be c1-c99. | $\left.\begin{array}{lrr}\text { Fn0ff ENTER } & & \text { Done } \\ \text { Plots0ff ENTER } & & \text { Done } \\ \{1,2,3,4\} \rightarrow \text { L1 ENTER } & \{1 & 2 \\ 3 & 3 & 4\end{array}\right\}$ <br> Press $\rightarrow$ [GRAPH] to display: |
| NewProb CATALOG |  |
| NewProb <br> Performs a variety of operations that let you begin a new problem from a cleared state without resetting the memory. <br> - Clears all single-character variable names (Clear $\mathrm{a}-\mathrm{z}$ ) in the current folder, unless the variables are locked or archived. <br> - Turns off all functions and stat plots (FnOff and PlotsOff) in the current graphing mode. <br> - Perfoms CIrDraw, CIrErr, CIrGraph, CIrHome, CIrIO, and CIrTable. | NewProb ENTER Done |
| nlnt() MATH/Calculus menu |  |
| nInt(expression1, var, lower, upper) $\Rightarrow$ expression <br> If the integrand expression 1 contains no variable other than var, and if lower and upper are constants, positive $\infty$, or negative $\infty$, then $\operatorname{nInt}()$ returns an approximation of $\int$ (expression1, var, lower, upper). This approximation is a weighted average of some sample values of the integrand in the interval lower<var<upper. | $\begin{aligned} n \operatorname{Int}\left(e^{\wedge}\left(-x^{\wedge} 2\right), x,-1,1\right) \text { ENTER } \\ 1.493 \ldots \end{aligned}$ |
| The goal is six significant digits. The adaptive algorithm terminates when it seems likely that the goal has been achieved, or when it seems unlikely that additional samples will yield a worthwhile improvement. <br> A warning is displayed ("Questionable accuracy") when it seems that the goal has not been achieved. | $\begin{aligned} & n \operatorname{Int}(\cos (x), x,-\pi, \pi+1 \mathrm{E}-12) \text { ENTER } \\ & -1.041 \ldots \mathrm{E}-12 \end{aligned} \begin{array}{r} \int\left(\cos (\mathrm{x}), \mathrm{x},-\pi, \pi+10^{\wedge}(-12)\right) \text { ENTER } \\ -\sin \left(\frac{1}{1000000000000}\right) \end{array}$ |

Nest nint() to do multiple numeric integration. Integration limits can depend on integration variables outside them.

```
nInt(nInt(e^(-x*y)/\sqrt{}{( (x^2- y^2),}
    y,-x,x),x,0,1) ENTER 3.304...
```

Note: See also $\int()$.


## not integer1 $\Rightarrow$ integer

Returns the one's complement of a real integer. Internally, integer1 is converted to a signed, 32-bit binary number. The value of each bit is flipped ( 0 becomes 1, and vice versa) for the one's complement. Results are displayed according to the Base mode.

You can enter the integer in any number base. For a binary or hexadecimal entry, you must use the Ob or Oh prefix, respectively. Without a prefix, the integer is treated as decimal (base 10).

If you enter a decimal integer that is too large for a signed, 32-bit binary form, a symmetric modulo operation is used to bring the value into the appropriate range.

In Hex base mode:
not $0 h 7 A C 36$ ENTER OhFFF853C9
$\square$ Important: Zero, not the letter O.
In Bin base mode:
0b100101 dec ENTER
37
not 0b100101 ENTER
Ob11111111111111111111111111011010
ans(1) dec ENTER -38
Note: A binary entry can have up to 32 digits (not counting the 0b prefix). A hexadecimal entry can have up to 8 digits.

Note: To type the conversion operator, press 2nd [■]. You can also select base conversions from the MATH/Base menu.
$\mathbf{n P r}$ (expression1, expression2) $\Rightarrow$ expression
For integer expression1 and expression2 with expression $1 \geq$ expression $2 \geq 0, \mathbf{n P r}()$ is the number of permutations of expression1 things taken expression2 at a time. Both arguments can be integers or symbolic expressions.
$\mathbf{n P r}$ (expression, 0 ) $\Rightarrow 1$
nPr(expression, negInteger) $\Rightarrow$
1/((expression +1$) \cdot($ expression +2$) .. .($ expression-negInteger))
$\mathbf{n P r}($ expression, posinteger) $\Rightarrow$ expression • (expression-1)... (expression-posinteger +1 )
nPr (expression, nonInteger) $\Rightarrow$ expression!/ (expression-noninteger)!
$n \operatorname{Pr}(z, 3)$ ENTER $z \cdot(z-2) \cdot(z-1)$
ans (1)|z=5 ENTER
60
$n \operatorname{Pr}(z,-3) \operatorname{ENTER} \frac{1}{(z+1) \cdot(z+2) \cdot(z+3)}$
$n \operatorname{Pr}(z, c)$ ENTER $\frac{z!}{(z-c)!}$
$\operatorname{ans}(1) * n P r(z-c,-c)$ ENTER
1
nPr (list1, list2) $\Rightarrow$ list
Returns a list of permutations based on the corresponding element pairs in the two lists. The arguments must be the same size list.
nPr (matrix1, matrix2) $\Rightarrow$ matrix
Returns a matrix of permutations based on the corresponding element pairs in the two matrices. The arguments must be the same size matrix.

## nSolve() MATH/Algebra menu Specify varOrGuess as: <br> variable <br> - or - <br> variable = real number

nSolve(equation, varOrGuess) $\Rightarrow$ number or error_string
Iteratively searches for one approximate real numeric solution to equation for its one variable.

For example, x is valid and so is $\mathrm{x}=3$.
nSolve() is often much faster than solve() or zeros(), particularly if the "" operator is used to constrain the search to a small interval containing exactly one simple solution.
nSolve() attempts to determine either one point where the residual is zero or two relatively close points where the residual has opposite signs and the magnitude of the residual is not excessive. If it cannot achieve this using a modest number of sample points, it returns the string "no solution found."

If you use nSolve() in a program, you can use getType() to check for a numeric result before using it in an algebraic expression.

Note: See also cSolve(), cZeros(), solve(), and zeros().
nSolve( $\left.x^{\wedge} 2+5 x-25=9, x\right)$ ENTER
3.844...
nSolve $\left(x^{\wedge} 2=4, x=-1\right)$ ENTER -2 .
nSolve $\left(x^{\wedge} 2=4, x=1\right)$ ENTER 2 .
Note: If there are multiple solutions, you can use a guess to help find a particular solution.

$$
\begin{aligned}
& \text { nSolve }\left(x^{\wedge} 2+5 x-25=9, x\right) \mid x<0 \text { ENTER } \\
& \\
& \text { nSolve }\left(\left((1+r)^{\wedge} 24-1\right) / r=26, r\right) \mid r> \\
& 0 \text { and } r<.25 \text { ENTER } \\
& \text { nSolve }\left(x^{\wedge} 2=-1, x\right) \text { ENTER } \\
& \text { "no solution found" }
\end{aligned}
$$

## OneVar MATH/Statistics menu

OneVar list1 [[, list2] [, list3] [, list4]]
Calculates 1 -variable statistics and updates all the system statistics variables.

All the lists must have equal dimensions except for list4.
list1 represents xlist. list2 represents frequency. list3 represents category codes. list4 represents category include list.

Note: list1 through list3 must be a variable name or c1-c99 (columns in the last data variable shown in the Data/Matrix Editor). list4 does not have to be a variable name and cannot be c1-c99.
$\{0,2,3,4,3,4,6\} \rightarrow$ L1 ENTER OneVar L1 [ENTER

Done ShowStat ENTER


P)Rx() MATH/Angle menu

PVRx(rExpression, $\theta$ Expression) $\Rightarrow$ expression
P>Rx(rList, $\theta$ List) $\Rightarrow$ list
PVRx(IMatrix, $\theta$ Matrix) $\Rightarrow$ matrix
Returns the equivalent $x$-coordinate of the (r, $\theta$ ) pair.

Note: The $\theta$ argument is interpreted as either a degree or radian angle, according to the current angle mode. If the argument is an expression, you can use ${ }^{\circ}$ or ${ }^{r}$ to override the angle mode setting temporarily.

In Radian angle mode:
$P>\operatorname{Rx}(r, \theta) \cos (\theta) \cdot r$
$\operatorname{P>P}\left(4,60^{\circ}\right)$ ENTER 2
$\operatorname{P} \operatorname{Rx}(\{-3,10,1.3\},\{\pi / 3,-\pi / 4,0\})$
ENTER

$$
\left\{\begin{array}{lll}
-3 / 2 & 5 \cdot \sqrt{2} & 1.3
\end{array}\right\}
$$

## P>Ry() MATH/Angle menu

PrRy(rExpression, $\theta$ Expression) $\Rightarrow$ expression
P>Ry(rList, $\theta$ List) $\Rightarrow$ list
P>Ry(rMatrix, $\theta$ Matrix) $\Rightarrow$ matrix
Returns the equivalent $y$-coordinate of the (r, $\theta$ ) pair.
Note: The $\theta$ argument is interpreted as either a degree or radian angle, according to the current angle mode. If the argument is an expression, you can use ${ }^{\circ}$ or ${ }^{r}$ to override the angle mode setting temporarily.

In Radian angle mode:
$P \operatorname{Ry}(r, \theta) \sin \sin (\theta) \cdot r$
PVRy $\left(4,60^{\circ}\right)$ ENTER
$2 \cdot \sqrt{3}$
$P>R y(\{-3,10,1.3\},\{\pi / 3,-\pi / 4,0\})$
ENTER

$$
\left\{\begin{array}{lll}
\frac{-3 \cdot \sqrt{3}}{2} & -5 \cdot \sqrt{2} & 0 .
\end{array}\right\}
$$

## part() CATALOG

part(expression1[,nonNegativelnteger])
This advanced programming function lets you identify and extract all of the sub-expressions in the simplified result of expression1.
For example, if expression1 simplifies to $\cos (\pi * x+3)$ :

- The $\boldsymbol{\operatorname { c o s }}()$ function has one argument: $(\pi * x+3)$.
- The sum of $\left(\pi^{*} \mathrm{x}+3\right)$ has two operands: $\pi^{*} \mathrm{x}$ and 3.
- The number 3 has no arguments or operands.
- The product $\pi *$ x has two operands: $\pi$ and x .
- The variable $x$ and the symbolic constant $\pi$ have no arguments or operands.
If $x$ has a numeric value and you press $\rightarrow$ ENTER, the numeric value of $\pi^{*} \mathrm{x}$ is calculated, the result is added to 3 , and then the cosine is calculated. $\cos ($ ) is the top-level operator because it is applied last.
part(expression1) $\Rightarrow$ number
Simplifies expression 1 and returns the number of toplevel arguments or operands. This returns 0 if expression1 is a number, variable, or symbolic constant such as $\pi, e, i$, or $\infty$.
part(expression1,0) m string part(cos( }\mp@subsup{\pi}{*}{*}x+3),0) ENTER "cos"
part(expression1,0) m string part(cos( }\mp@subsup{\pi}{*}{*}x+3),0) ENTER "cos"
Simplifies expression1 and returns a string that contains the top-level function name or operator. This returns string(expression1) if expression1 is a number, variable, or symbolic constant such as $\pi, e$, $i$, or $\infty$.
part(expression1, $n$ ) $\Rightarrow$ expression
Simplifies expression 1 and returns the $n^{\text {th }}$ argument or operand, where $n$ is $>0$ and $\leq$ the number of toplevel arguments or operands returned by part(expression1). Otherwise, an error is returned.

By combining the variations of part(), you can extract all of the sub-expressions in the simplified result of expression1. As shown in the example to the right, you can store an argument or operand and then use part() to extract further sub-expressions.

Note: When using part(), do not rely on any particular order in sums and products.

Expressions such as $(x+y+z)$ and $(x-y-z)$ are represented internally as $(x+y)+z$ and $(x-y)-z$. This affects the values returned for the first and second argument. There are technical reasons why part $(x+y+z, 1)$ returns $y+x$ instead of $x+y$.

Similarly, $x^{*} y^{*} z$ is represented internally as $(x * y) * z$. Again, there are technical reasons why the first argument is returned as $y \cdot x$ instead of $x \cdot y$.

When you extract sub-expressions from a matrix, remember that matrices are stored as lists of lists, as illustrated in the example to the right.
$\operatorname{part}(\cos (\pi * x+3), 1)$ ENTER $3+\pi \cdot x$
Note: Simplification changed the order of the argument.

| part $\left(\cos \left(\pi^{*} x+3\right)\right)$ ENTER | 1 |
| :---: | :---: |
| $\operatorname{part}(\cos (\pi * x+3), 0)$ ENTER | "cos" |
| $\operatorname{part}\left(\cos \left(\pi^{*} x+3\right), 1\right) \rightarrow$ temp | ENTER |
|  | $3+\pi \cdot x$ |
| temp ENTER | $\pi \cdot x+3$ |
| part (temp,0) ENTER | "+" |
| part(temp) ENTER | 2 |
| part(temp,2) ENTER | 3 |
| part(temp,1) $\rightarrow$ temp ENTER | $\pi \cdot \mathrm{x}$ |
| part(temp,0) ENTER | "*" |
| part(temp) ENTER | 2 |
| part(temp,1) [ENTER | $\pi$ |
| part(temp,2) ENTER | X |
| part ( $x+y+z$ ) ENTER | 2 |
| part ( $x+y+z, 2$ ) ENTER | Z |
| part ( $x+y+z, 1$ ) ENTER | $y+x$ |

part (x*y*z) ENTER 2
part $(x * y * z, 2)$ ENTER $z$
$\operatorname{part}(x * y * z, 1)$ ENTER $y \cdot x$
$\operatorname{part}([a, b, c ; x, y, z], 0)$ ENTER $"\{"$
$\operatorname{part}([a, b, c ; x, y, z])$ ENTER 2
$\operatorname{part}([a, b, c ; x, y, z], 2) \rightarrow t e m p$
ENTER
$\left\{\begin{array}{lll}x & y & z\end{array}\right\}$
part(temp,0) ENTER "\{"
part(temp) ENTER 3
part(temp,3) ENTER z
delVar temp ENTER Done

The example Program Editor function to the right uses getType() and part() to partially implement symbolic differentiation. Studying and completing this function can help teach you how to differentiate manually. You could even include functions that the cannot differentiate, such as Bessel functions.
: $d(y, x)$
: Func
:Local f
:If getType(y)="VAR"
: Return when $(y=x, 1,0,0)$
: If part $(y)=0$
: Return 0 © $y=\pi, \infty, i, n u m b e r s$
: part $(y, 0) \rightarrow f$
:If f="-" oif negate
: Return - d(part(y,1),x)
:If f="-" oif minus
: Return d(part(y,1),x)

$$
-d(\operatorname{part}(y, 2), x)
$$

:If f="+"
: Return d(part(y,1),x) $+d(\operatorname{part}(y, 2), x)$
:If f="*"
: Return
part(y,1)*d(part(y,2),x) $+\operatorname{part}(\mathrm{y}, 2) * d(\operatorname{part}(\mathrm{y}, 1), x)$
:If f="\{"
: Return seq(d(part(y,k),x), k,1,part(y))
: Return undef
: EndFunc

## PassErr CATALOG

## PassErr

Passes an error to the next level.
If "errornum" is zero, PassErr does not do anything.
The Else clause in the program should use CIrErr or PassErr. If the error is to be processed or ignored, use CIrErr. If what to do with the error is not known, use PassErr to send it to the next error handler. (See also CIrErr.)

## Pause catalog

Pause [expression]
Suspends program execution. If you include expression, displays expression on the Program I/O screen.
expression can include conversion operations such as $>D D$ and $>$ Rect. You can also use the operator to perform unit and number base conversions.

If the result of expression is too big to fit on a single screen, you can use the cursor pad to scroll the display.

Program execution resumes when you press ENTER.

Program segment:
: Cirio
: DelVar temp
: $1 \rightarrow$ temp [1]
: $1 \rightarrow$ temp[2]
:Disp temp[2]
: © Guess the Pattern
:For i,3,20
: temp[i-2]+temp[i-1] $\rightarrow$ temp[i]
: Disp temp[i]
: Disp temp,"Can you guess the next","number?"
: Pause
: EndFor

PlotsOff [1] [, 2] [, 3] ... [, 9]
Turns off the specified plots for graphing. When in 2-graph mode, only affects the active graph.

Plots0ff 1,2,5 ENTER
Plots0ff ENTER
Done

If no parameters, then turns off all plots.

## PlotsOn catalog

PlotsOn [1] [, 2] [, 3] ... [, 9]
Turns on the specified plots for graphing. When in 2-graph mode, only affects the active graph.

If you do not include any arguments, turns on all plots.

Plots0n 2,4,5 ENTER Done
Plots0n ENTER Done

POolar MATH/Matrix/Vector ops menu vector Polar

Displays vector in polar form [r $\angle \theta$ ]. The vector must be of dimension 2 and can be a row or a column.

Note: $>$ Polar is a display-format instruction, not a conversion function. You can use it only at the end of an entry line, and it does not update ans.

Note: See also Rect.

## complexValue •Polar

Displays complexVector in polar form.

- Degree angle mode returns ( $\mathrm{r} \angle \theta$ ).
- Radian angle mode returns re ${ }^{i \theta}$.
complexValue can have any complex form. However, an re ${ }^{i \theta}$ entry causes an error in Degree angle mode.
Note: You must use the parentheses for an $(r \angle \theta)$ polar entry.
$[1,3$.$] Polar ENTER$
$[\mathrm{x}, \mathrm{y}]>$ Polar ENTER
- $\left.\begin{array}{ll}1 & 3 .\end{array}\right]$ Polar
[ $3.16228 \angle 1.24905]$
- [ $\left.\begin{array}{ll}x & 4\end{array}\right]$ PFol.ar
$\left[\sqrt{x^{2}+y^{2}}<\frac{\pi \cdot \operatorname{sign}(y)}{2}-\operatorname{tar} t\right]$

In Radian angle mode:
$3+4 i$ Polar ENTER $\quad e^{i} \cdot\left(\frac{\pi}{2}-\tan ^{-1}(3 / 4)\right) \cdot 5$
$(4 \angle \pi / 3)$ PPolar ENTER $\quad e^{\frac{i \cdot \pi}{3}} \cdot 4$

In Degree angle mode:
$3+4$ irPolar ENTER (5 $\angle 90-\tan ^{-1}(3 / 4)$ )

## polyEval() MATH/List menu

```
polyEval(list1, expression1) = expression
    polyEval(list1, list2) = expression
```

Interprets the first argument as the coefficient of a descending-degree polynomial, and returns the polynomial evaluated for the value of the second argument.
polyEval ( $\{a, b, c\}, x)$ ENTER $a \cdot x^{2}+b \cdot x+c$
polyEval (\{1,2,3,4\},2) ENTER 26
polyEval (\{1, 2, 3, 4\}, \{2, -7\})
ENTER
\{26-262\}

## PopUp CATALOG

PopUp itemList, var
Displays a pop-up menu containing the character strings from itemList, waits for you to select an item, and stores the number of your selection in var.
The elements of itemList must be character strings: \{item1String, item2String, item3String, ...\}

If var already exists and has a valid item number, that item is displayed as the default choice.
itemList must contain at least one choice.

PopUp
\{"1990", "1991", "1992"\}, var1 ENTER


PowerReg list1, list2[, [list3] [, list4, list5]]
Calculates the power regression and updates all the system statistics variables.

All the lists must have equal dimensions except for list5.
list1 represents xlist. list2 represents ylist. list3 represents frequency. list4 represents category codes. list5 represents category include list.

Note: list1 through list4 must be a variable name or c1-c99 (columns in the last data variable shown in the Data/Matrix Editor). list5 does not have to be a variable name and cannot be c1-c99.

## Prgm CATALOG

Prgm
EndPrgm
Required instruction that identifies the beginning of a program. Last line of program must be EndPrgm.

In function graphing mode:
$\{1,2,3,4,5,6,7\} \rightarrow \operatorname{L1}$ ENTER $\left\{\begin{array}{llll}1 & 2 & 3 & \ldots\end{array}\right\}$
$\{1,2,3,4,3,4,6\} \rightarrow$ L2 ENTER
$\left\{\begin{array}{llll}1 & 2 & 3 & \ldots\end{array}\right\}$
Done
PowerReg L1,L2 ENTER
ShowStat ENTER


ENTER
$\operatorname{Regeq}(x) \rightarrow y 1(x)$ ENTER Done

NewP1ot 1,1,L1,L2 ENTER Done

- [GRAPH]


Program segment:

```
:prgmname()
:Prgm
:
:EndPrgm
```


## Product (PI) See $\Pi()$, page 881.

## product() MATH/List menu

product(list[l, start[, end]]) $\Rightarrow$ expression
Returns the product of the elements contained in list. Start and end are optional. They specify a range of elements.
product(matrix $1[$, start[, end]]) $\Rightarrow$ matrix
Returns a row vector containing the products of the elements in the columns of matrix1. Start and end are optional. They specify a range of rows.
product(\{1,2,3,4\}) ENTER
24
product $(\{2, x, y\})$ ENTER $2 \cdot x \cdot y$
product (\{4,5,8,9\},2,3) ENTER 40
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
$[4,10,18]$

## Prompt CATALOG

Prompt var1[, var2] [, var3] ...
Displays a prompt on the Program I/O screen for each variable in the argument list, using the prompt var1?. Stores the entered expression in the corresponding variable.

Prompt must have at least one argument.

Program segment:

Prompt A,B,C
$\vdots$
EndPrgm
propFrac(4/3) ENTER
$1+1 / 3$
propFrac(rational_number) returns rational_number as the sum of an integer and a fraction having the same sign and a greater denominator magnitude than numerator magnitude.
propFrac(rational_expression, var) returns the sum of proper ratios and a polynomial with respect to var. The degree of var in the denominator exceeds the degree of var in the numerator in each proper ratio. Similar powers of var are collected. The terms and their factors are sorted with var as the main variable.

If var is omitted, a proper fraction expansion is done with respect to the most main variable. The coefficients of the polynomial part are then made proper with respect to their most main variable first and so on.

For rational expressions, propFrac() is a faster but less extreme alternative to expand().

## PtChg catalog

PtChg $x, y$
PtChg xList, yList
Displays the Graph screen and reverses the screen pixel nearest to window coordinates $(x, y)$.

Note: PtChg through PtText show continuing similar examples.
PtChg 2,4 ENTER


PtOff CATALOG
Ptoff $x, y$
Ptoff $x$ List, $y$ List
Displays the Graph screen and turns off the screen pixel nearest to window coordinates $(x, y)$.

## PtOn CATALOG

PtOn $x, y$
PtOn xList, yList
Displays the Graph screen and turns on the screen pixel nearest to window coordinates $(x, y)$.
ptTest() CATALOG
ptTest $(x, y) \Rightarrow$ Boolean constant expression ptTest (xList, yList) $\Rightarrow$ Boolean constant expression

Pt0ff 2,4 ENTER


PtOn 3,5 ENTER


Returns true or false. Returns true only if the screen pixel nearest to window coordinates $(x, y)$ is on.

## PtText CATALOG

## PtText string, $x, y$

Displays the Graph screen and places the character string string on the screen at the pixel nearest the specified $(x, y)$ window coordinates.
string is positioned with the upper-left corner of its

PtText "sample", 3,5 ENTER
 first character at the coordinates.

## PxIChg CATALOG

PxiChg row, col
PxIChg rowList, colList
Displays the Graph screen and reverses the pixel at pixel coordinates (row, col).

Note: Regraphing erases all drawn items.

## PxiCrcl catalog

PxiCrcl row, col, r[, drawMode]
Displays the Graph screen and draws a circle centered at pixel coordinates (row, col) with a radius of $r$ pixels.

If drawMode $=1$, draws the circle (default) If drawMode $=0$, turns off the circle. If drawMode $=-1$, inverts pixels along the
(-) PxiCrcl $40,80,30,1$ ENTER
EP PxlCrcl 50,125,40,1 ENTER circle.

Note: Regraphing erases all drawn items See also Circle.

## PxiHorz catalog

PxIHorz row [, drawMode]
Displays the Graph screen and draws a horizontal line at pixel position row.

If drawMode = 1 , draws the line (default).
If drawMode $=0$, turns off the line.
If drawMode $=-1$, turns a line that is on to off or off to

PxiHorz 25,1 ENTER
 on (inverts pixels along the line).

Note: Regraphing erases all drawn items. See also LineHorz.

Px1Chg 2,4 ENTER


## CATALOG

PxiLine rowStart, colStart, rowEnd, colEnd [, drawMode]
Displays the Graph screen and draws a line between pixel coordinates (rowStart, colStart) and (rowEnd, colEnd), including both endpoints.

If drawMode = 1 , draws the line (default).
If drawMode $=0$, turns off the line.
If drawMode $=-1$, turns a line that is on to off or off to on (inverts pixels along the line).
(-) PxiLine $50,15,20,90,1$ ENTER
표 Pxiline $80,20,30,150,1$
ENTER


Note: Regraphing erases all drawn items. See also Line.

## PxIOff CATALOG

PxiOff row, col
PxIOff rowList, collist
Displays the Graph screen and turns off the pixel at pixel coordinates (row, col).

Note: Regraphing erases all drawn items.

PxiHorz 25,1 ENTER
Pxl0ff 25,50 ENTER


25,50

## PxiOn CATALOG

PxiOn row, col
PxIOn rowList, colList
Displays the Graph screen and turns on the pixel at pixel coordinates (row, col).

Note: Regraphing erases all drawn items.
Px10n 25,50 ENTER


## pxiTest() CATALOG

pxITest (row, col) $\Rightarrow$ Boolean expression pxITest (rowList, colList) $\Rightarrow$ Boolean expression

Returns true if the pixel at pixel coordinates (row, col) is on. Returns false if the pixel is off.

Note: Regraphing erases all drawn items.

Px10n 25,50 ENTER
HOME
PxiTest 25,50 ) ENTER true
Px10ff 25,50 ENTER
[- HOME
$\square$ [CALC HOME]
Px1Test(25,50) ENTER
false

## PxIText CATALOG

PxIText string, row, col
Displays the Graph screen and places character string string on the screen, starting at pixel coordinates (row, col).
string is positioned with the upper-left corner of its first character at the coordinates.

Note: Regraphing erases all drawn items.
(- PxlText "sample
text", 20,10 ENTER
ㅍil PxlText "sample text", 20,50 ENTER


## PxiVert catalog

## PxIVert col [, drawMode]

Draws a vertical line down the screen at pixel position col.

If drawMode = 1 , draws the line (default).
If drawMode $=0$, turns off the line.
If drawMode $=-1$, turns a line that is on to off or off to
PxlVert 50,1 ENTER
 on (inverts pixels along the line).

Note: Regraphing erases all drawn items. See also LineVert.

QR matrix, qMatName, rMatName[, tol]
Calculates the Householder QR factorization of a real or complex matrix. The resulting $Q$ and $R$ matrices are stored to the specified MatNames. The Q matrix is unitary. The $R$ matrix is upper triangular.

Optionally, any matrix element is treated as zero if its absolute value is less than tol. This tolerance is used only if the matrix has floating-point entries and does not contain any symbolic variables that have not been assigned a value. Otherwise, tol is ignored.

- If you use ENTER or set the mode to Exact/Approx=APPROXIMATE, computations are done using floating-point arithmetic.
- If tol is omitted or not used, the default tolerance is calculated as:

5E-14* max(dim(matrix)) * rowNorm(matrix)

The QR factorization is computed numerically using Householder transformations. The symbolic solution is computed using Gram-Schmidt. The columns in qMatName are the orthonormal basis vectors that span the space defined by matrix.

The floating-point number (9.) in m1 causes results to be calculated in floating-point form.

| $[1,2,3 ; 4,5,6 ; 7,8,9]$. | $\rightarrow \mathrm{m} 1$ [ENTER |
| ---: | :--- |
| $\left[\begin{array}{ccc}1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 .\end{array}\right]$ |  |

QR m1, qm, rm ENTER
Done
qm ENTER $\left[\begin{array}{lll}.123 \ldots & .904 \ldots & .408 \ldots \\ .492 \ldots & .301 \ldots & -.816 \ldots \\ .861 \ldots & -.301 \ldots & .408 \ldots\end{array}\right]$
$r m$ ENTER $\left[\begin{array}{lll}8.124 \ldots & 9.601 \ldots & 11.078 \ldots . . \\ 0 . & .904 \ldots & 1.809 \ldots \\ 0 . & 0 . & 0 .\end{array}\right]$

$$
[m, n ; 0, p] \rightarrow m 1 \text { ENTER } \quad\left[\begin{array}{cc}
m & n \\
0 & p
\end{array}\right]
$$

QR m1, qm, rm ENTER
Done
qm ENTER

$$
\left[\begin{array}{ll}
\frac{m}{\sqrt{m^{2}+0^{2}}} & \frac{-\operatorname{sign}(m \cdot p-n \cdot 0) \cdot 0}{\sqrt{m^{2}+0^{2}}} \\
\frac{0}{\sqrt{m^{2}+0^{2}}} & \frac{m \cdot \operatorname{sign}(m \cdot p-n \cdot 0)}{\sqrt{m^{2}+0^{2}}}
\end{array}\right]
$$



## QuadReg MATH/Statistics/Regressions menu

QuadReg list1, list2[, [list3] [, list4, list5]]
Calculates the quadratic polynomial regression and updates the system statistics variables.

All the lists must have equal dimensions except for list5.
list1 represents xlist. list2 represents ylist. list3 represents frequency. list4 represents category codes. list5 represents category include list.

In function graphing mode:



Note: list1 through list4 must be a variable name or c1-c99. (columns in the last data variable shown in the Data/Matrix Editor). list5 does not have to be a variable name and cannot be c1-c99.

ENTER
Regeq $(x) \rightarrow y 1(x)$ ENTER
Done NewP1ot 1,1,L1, L2 ENTER Done

- [GRAPH]



## QuartReg MATH/Statistics/Regressions menu

QuartReg list1, list2[, [list3] [, list4, list5]]
Calculates the quartic polynomial regression and updates the system statistics variables.
All the lists must have equal dimensions except for list5.
list1 represents xlist.
list2 represents ylist. list3 represents frequency. list4 represents category codes. list5 represents category include list.

Note: list1 through list4 must be a variable name or c1-c99 (columns in the last data variable shown in the Data/Matrix Editor). list5 does not have to be a variable name and cannot be c1-c99.

## $\mathbf{R} \mathbf{P} \theta() \quad$ MATH/Angle menu

R>P日 (xExpression, yExpression) $\Rightarrow$ expression $\mathbf{R > P} \boldsymbol{\theta}$ (xList, yList) $\Rightarrow$ list R>P $\boldsymbol{\theta}$ (xMatrix, yMatrix) $\Rightarrow$ matrix

Returns the equivalent $\theta$-coordinate of the $(x, y)$ pair arguments.
Note: The result is returned as either a degree or radian angle, according to the current angle mode.

In function graphing mode:
$\{-2,-1,0,1,2,3,4,5,6\} \rightarrow L 1$ ENTER
$\left\{\begin{array}{llll}-2 & -1 & 0 & \ldots\end{array}\right\}$
$\{4,3,1,2,4,2,1,4,6\} \rightarrow$ L2 ENTER
$\left\{\begin{array}{llll}4 & 3 & 1 & \text {... }\end{array}\right\}$
QuartReg L1,L2 ENTER
Done
ShowStat ENTER


ENTER

| Regeq $(x) \rightarrow y 1(x)$ ENTER | Done |
| :--- | :--- |
| NewPlot 1,1, L1,L2 ENTER | Done |

- [GRAPH]


In Degree angle mode:
$R>P \theta(x, y)$ ENTER

$$
\begin{array}{r}
\text { RrPe(x, } 9) \\
90 \cdot \operatorname{sinn}(9)-\tan ^{-1}\left(\frac{x}{y}\right)
\end{array}
$$

In Radian angle mode:
R $>\operatorname{P\theta }(3,2)$ ENTER
$\mathrm{R} P \operatorname{P\theta }([3,-4,2],[0, \pi / 4,1.5])$ ENTER


R $>\operatorname{Pr}(x E x p r e s s i o n, y E x p r e s s i o n) \Rightarrow$ expression
R>Pr (xList, yList) $\Rightarrow$ list
R>Pr (xMatrix, yMatrix) $\Rightarrow$ matrix
Returns the equivalent $r$-coordinate of the $(x, y)$ pair arguments.

In Radian angle mode:
R $\operatorname{Pr}(3,2)$ ENTER
R $\operatorname{Pr} \operatorname{Pr}(\mathrm{x}, \mathrm{y})$ ENTER
$R \operatorname{Pr}([3,-4,2],[0, \pi / 4,1.5])$ ENTER

rand() MATH/Probability menu
rand $([n]) \Rightarrow$ expression
$n$ is an integer $\neq$ zero.
With no parameter, returns the next random number between 0 and 1 in the sequence. When an argument is positive, returns a random integer in the interval [ $1, n$ ].
When an argument is negative, returns a random integer in the interval $[-n,-1]$.

## randMat() MATH/Probability menu

## randMat(numRows, numColumns) $\Rightarrow$ matrix

Returns a matrix of integers between -9 and 9 of the specified dimension.

Both arguments must simplify to integers.

RandSeed 1147 ENTER
Done
$\uparrow$ (Sets the random-number seed.)
rand () ENTER . 158...
rand (6) ENTER 5
rand (-100) ENTER -49

RandSeed 1147 ENTER
Done
$\left[\begin{array}{rrr}8 & -3 & 6 \\ -2 & 3 & -6 \\ 0 & 4 & -6\end{array}\right]$

Note: The values in this matrix will change each time you press ENTER.

## randNorm() MATH/Probability menu

```
randNorm(mean,sd) => expression
```

Returns a decimal number from the specific normal distribution. It could be any real number but will be heavily concentrated in the interval [mean-3* sd, mean $+3 *$ sd].

RandSeed 1147 ENTER
randNorm ( 0,1 ) ENTER
randNorm(3,4.5) ENTER

Done
. 492... -3.543...

## randPoly() MATH/Probability menu

## randPoly(var, order) $\Rightarrow$ expression

Returns a polynomial in var of the specified order. The coefficients are random integers in the range - 9 through 9 . The leading coefficient will not be zero.

RandSeed 1147 ENTER Done
randPoly $(x, 5)$ ENTER

$$
-2 \cdot x^{5+3} \cdot x^{4}-6 \cdot x^{3}+4 \cdot x-6
$$

order must be 0-99.

## RandSeed MATH/Probability menu

## RandSeed number

If number $=0$, sets the seeds to the factory defaults for the random-number generator. If number $\neq 0$, it is used to generate two seeds, which are stored in system variables seed1 and seed2.

RandSeed 1147 ENTER Done
rand () ENTER .158...

RcIGDB CATALOG
RcIGDB GDBvar
Restores all the settings stored in the Graph database variable GDBvar.

For a listing of the settings, see StoGDB.
Note: It is necessary to have something saved in GDBvar before you can restore it.

## RcIPic CATALOG

RcIPic picVar [, row, column]
Displays the Graph screen and adds the picture stored in picVar at the upper left-hand corner pixel coordinates (row, column) using OR logic.
picVar must be a picture data type.
Default coordinates are $(0,0)$.

## real() MATH/Complex menu

real(expression1) $\Rightarrow$ expression
Returns the real part of the argument.
Note: All undefined variables are treated as real variables. See also imag().
$\operatorname{real}($ list 1$) \Rightarrow$ list $\quad \operatorname{real}(\{a+i * \mathrm{~b}, 3, i\})$ ENTER $\left\{\begin{array}{lll}a & 3 & 0\end{array}\right\}$
Returns the real parts of all elements.


Returns the real parts of all elements.

## Rect MATH/Matrix/Vector ops menu

vector $\mathbf{R e c t}$
Displays vector in rectangular form $[x, y, z]$. The vector must be of dimension 2 or 3 and can be a row or a column.

Note: $>$ Rect is a display-format instruction, not a conversion function. You can use it only at the end of an entry line, and it does not update ans.

Note: See also PPolar.

## complexValue Rect

Displays complexValue in rectangular form a+bi. The complexValue can have any complex form. However, an re ${ }^{i \theta}$ entry causes an error in Degree angle mode.

Note: You must use parentheses for an $(\mathrm{r} \angle \theta)$ polar entry.
$[3, \angle \pi / 4, \angle \pi / 6]$ Rect ENTER

$$
\left[\begin{array}{lll}
\frac{3 \cdot \sqrt{2}}{4} & \frac{3 \cdot \sqrt{2}}{4} & \frac{3 \cdot \sqrt{3}}{2}
\end{array}\right]
$$

[a, $\angle \mathrm{b}, \angle \mathrm{c}]$ ENTER $[\mathrm{a} \cdot \cos (\mathrm{b}) \cdot \sin (\mathrm{c})$ $a \cdot \sin (b) \cdot \sin (c) a \cdot \cos (c)]$

In Radian angle mode:

| $4 e^{\wedge}(\pi / 3) \vee \operatorname{Rect}$ ENTER | $4 \cdot e^{\frac{\pi}{3}}$ |
| :--- | ---: |
| $(4 \angle \pi / 3)$ | $\operatorname{Rect}$ ENTER | $2+2 \cdot \sqrt{3} \cdot i$

In Degree angle mode:
$(4 \angle 60)$ Rect ENTER $2+2 \cdot \sqrt{3} \cdot \boldsymbol{i}$

Note: To type $>$ Rect from the keyboard, press
2nd [ $\downarrow$ ] for the operator. To type $\angle$, press
2nd [ $\angle$ ].

| ref() | MATH/Matrix menu |  |
| :---: | :---: | :---: |
|  | trix $1[$, tol]) $\Rightarrow$ matrix <br> Returns the row echelon form of matrix1. <br> Optionally, any matrix element is treated as zero if its absolute value is less than tol. This tolerance is used only if the matrix has floating-point entries and does not contain any symbolic variables that have not been assigned a value. Otherwise, tol is ignored. <br> - If you use ENTER or set the mode to Exact/Approx=APPROXIMATE, computations are done using floating-point arithmetic. <br> - If tol is omitted or not used, the default tolerance is calculated as: $5 \mathrm{E}-14 * \max (\operatorname{dim}(\text { matrix } 1)) * \operatorname{rowNorm}(\text { matrix } 1)$ <br> Note: See also rref(). | $\begin{aligned} & \operatorname{ref}([-2,-2,0,-6 ; 1,-1,9,-9 ;-5, \\ & 2,4,-4])\left[\begin{array}{cccc} \text { ENTER } \\ {\left[\begin{array}{cccc} 1 & -2 / 5 & -4 / 5 & 4 / 5 \\ 0 & 1 & 4 / 7 & 11 / 7 \\ 0 & 0 & 1 & -62 / 71 \end{array}\right]} \\ {[a, b, c ; e, f, g] \rightarrow m 1 \text { [ENTER }\left[\begin{array}{lll} a & b & c \\ e & f & g \end{array}\right]} \\ \operatorname{ref}(m 1) \text { ENTER }\left[\begin{array}{cccc} 1 & \frac{f}{e} & \frac{g}{e} & \\ 0 & 1 & \frac{a \cdot g-c \cdot e}{a \cdot f-b \cdot e} \end{array}\right] \end{array} .\right. \end{aligned}$ |
| remain() | MATH/Number menu |  |
| ```remain(expression1, expression2) => expression remain(list1, list2) = list remain(matrix1, matrix2) }=>\mathrm{ matrix``` |  | remain(7,0) ENTER <br> remain(7,3) ENTER |
|  | Returns the remainder of the first argument with respect to the second argument as defined by the identities: | remain $(-7,3)$ ENTER -1 <br> remain $(7,-3)$ ENTER 1 |
|  | $\begin{aligned} & \text { remain }(x, 0) \equiv x \\ & \operatorname{remain}(x, y) \equiv x-y * \operatorname{Part}(x / y) \end{aligned}$ | remain $(-7,-3)$ ENTER remain $(\{12,-14,16\},\{9,7,-5\})$ ENTER |
|  | As a consequence, note that remain(- $\mathrm{x}, \mathrm{y}$ ) $\equiv$ - remain $(x, y)$. The result is either zero or it has the same sign as the first argument. <br> Note: See also mod(). | remain([9,-7;6,4],[4,3;4,-3]) <br> ENTER $\left[\begin{array}{ll} 1 & -1 \\ 2 & 1 \end{array}\right]$ |
| Rename | CATALOG |  |
|  | ne oldVarName, newVarName Renames the variable oldVarName as newVarName. | $\{1,2,3,4\} \rightarrow$ L1 ENTER $\{1,2,3,4\}$ <br> Rename L1, list1 ENTER  <br> list1 ENTER $\{1,2,3,4\}$ |
| Request CATALOG |  |  |
| Request promptString, var <br> If Request is inside a Dialog...EndDlog construct, it creates an input box for the user to type in data. If it is a stand-alone instruction, it creates a dialog box for this input. In either case, if var contains a string, it is displayed and highlighted in the input box as a default choice. promptString must be $\leq 20$ characters. |  | Request "Enter Your Name",str1 ENTER |

Return CATALOG

Return [expression]
Returns expression as the result of the function. Use within a Func...EndFunc block, or Prgm...EndPrgm block.

Note: Use Return without an argument to exit a program.

Note: Enter the text as one long line on the Home screen (without line breaks).

Define factoral(nn)=Func
:local answer,count: $1 \rightarrow$ answer
: For count,1,nn
: answer* count $\rightarrow$ answer: EndFor
: Return answer: EndFunc ENTER Done
factoral(3) ENTER 6

## right( ) MATH/List menu

right(list1[, num]) $\Rightarrow$ list
right(\{1,3,-2,4\},3) ENTER
$\left\{\begin{array}{lll} & -2 & 4\end{array}\right\}$

Returns the rightmost num elements contained in list1.

If you omit num, returns all of list1.
right(sourceString[, num]) $\Rightarrow$ string right("He110",2) ENTER "10"

Returns the rightmost num characters contained in character string sourceString.

If you omit num, returns all of sourceString.
right(comparison) $\Rightarrow$ expression $\quad$ right $(x<3)$ ENTER 3
Returns the right side of an equation or inequality.

## rotate() MATH/Base menu

rotate(integer1[,\#ofRotations]) $\Rightarrow$ integer
Rotates the bits in a binary integer. You can enter integer1 in any number base; it is converted automatically to a signed, 32-bit binary form. If the magnitude of integer1 is too large for this form, a symmetric modulo operation brings it within the range.

If \#of Rotations is positive, the rotation is to the left. If \#of Rotations is negative, the rotation is to the right. The default is -1 (rotate right one bit).

For example, in a right rotation:

Each bit rotates right
Ob00000000000001111010110000110101
Rightmost bit rotates to leftmost. produces:

Ob10000000000000111101011000011010
The result is displayed according to the Base mode.

In Bin base mode:
rotate(0b1111010110000110101)
ENTER
Ob10000000000000111101011000011010
rotate(256,1) ENTER 0b1000000000

In Hex base mode:
rotate(0h78E) ENTER
0h3C7
rotate(0h78E,-2) ENTER $0 h 800001 \mathrm{E} 3$
rotate(0h78E,2) ENTER Oh1E38
Important: To enter a binary or hexadecimal number, always use the 0b or Oh prefix (zero, not the letter 0 ).
rotate(list1[,\#ofRotations]) $\Rightarrow$ list
Returns a copy of list1 rotated right or left by \#of Rotations elements. Does not alter list1.

If \#of Rotations is positive, the rotation is to the left. If \#of Rotations is negative, the rotation is to the right. The default is - 1 (rotate right one element).

In Dec base mode
rotate( $\{1,2,3,4\}$ ) ENTER
$\left\{\begin{array}{llll}4 & 1 & 3\end{array}\right\}$
rotate(\{1,2,3,4\},-2) ENTER
$\left\{\begin{array}{llll}3 & 4 & 1 & 2\end{array}\right\}$
$\left\{\begin{array}{llll}2 & 3 & 4 & 1\end{array}\right\}$
rotate(string1[,\#ofRotations]) $\Rightarrow$ string
Returns a copy of string1 rotated right or left by \#of Rotations characters. Does not alter string1.

If \#of Rotations is positive, the rotation is to the left. If \#of Rotations is negative, the rotation is to the right. The default is -1 (rotate right one character).

## round() MATH/Number menu

round(expression1[, digits]) $\Rightarrow$ expression
round(1.234567,3) ENTER
1.235

Returns the argument rounded to the specified number of digits after the decimal point.
digits must be an integer in the range $0-12$. If digits is not included, returns the argument rounded to 12 significant digits.

Note: Display digits mode may affect how this is displayed.
round (list $1[$ digits $]) \Rightarrow$ list $\quad$ round $(\{\pi, \sqrt{ }(2), 1 n(2)\}, 4)$ ENTER
Returns a list of the elements rounded to the specified number of digits.
round(matrix 1 [, digits]) $\Rightarrow$ matrix
Returns a matrix of the elements rounded to the specified number of digits.
round([1n(5), $\left.\left.\ln (3) ; \pi, e^{\wedge}(1)\right], 1\right)$ ENTER

$$
\left[\begin{array}{ll}
1.6 & 1.1 \\
3.1 & 2.7
\end{array}\right]
$$

## rowAdd() MATH/Matrix/Row ops menu

rowAdd(matrix1, rIndex1, rIndex2) $\Rightarrow$ matrix
Returns a copy of matrix 1 with row rIndex2 replaced by the sum of rows rIndex1 and rIndex2.
rowAdd([3,4;-3,-2],1,2) ENTER

rowAdd([a,b;c,d],1,2) ENTER


## rowDim() MATH/Matrix/Dimensions menu

rowDim(matrix) $\Rightarrow$ expression
Returns the number of rows in matrix.
Note: See also colDim().
$[1,2 ; 3,4 ; 5,6] \rightarrow$ M1 ENTER
rowdim(M1) ENTER

3
rowNorm(matrix) $\Rightarrow$ expression
Returns the maximum of the sums of the absolute values of the elements in the rows in matrix.
Note: All matrix elements must simplify to numbers. See also colNorm().
rowNorm([-5,6,-7;3,4,9;9,-9,-7])
ENTER 25

$$
[1,2 ; 3,4 ; 5,6] \rightarrow \text { Mat ENTER }
$$

rowSwap(matrix1, rIndex1, rindex2) $\Rightarrow$ matrix
Returns matrix 1 with rows rindex1 and rindex2 exchanged.

## RplcPic CATALOG

RplcPic picVar[, row][, column]
Clears the Graph screen and places picture picVar at pixel coordinates (row, column). If you do not want to clear the screen, use RcIPic.
picVar must be a picture data type variable. row and column, if included, specify the pixel coordinates of the upper left corner of the picture. Default coordinates are ( 0,0 ).
Note: For less than full-screen pictures, only the area affected by the new picture is cleared.

## rref() MATH/Matrix menu

$\operatorname{rref}($ matrix $[$ [tol $]) \Rightarrow$ matrix
Returns the reduced row echelon form of matrix1.

Optionally, any matrix element is treated as zero if its absolute value is less than tol. This tolerance is used only if the matrix has floating-point entries and does not contain any symbolic variables that have not been assigned a value. Otherwise, tol is ignored.

- If you use ENTER or set the mode to Exact/Approx=APPROXIMATE, computations are done using floating-point arithmetic.
- If tol is omitted or not used, the default tolerance is calculated as:
$5 \mathrm{E}-14 * \max (\operatorname{dim}($ matrix1) ) * rowNorm(matrix1)
Note: See also ref().


## rref([-2,-2,0,-6;1,-1,9,-9;

$-5,2,4,-4])$ ENTER

$$
\left[\begin{array}{rrrr}
1 & 0 & 0 & 66 / 71 \\
0 & 1 & 0 & \frac{147}{71} \\
0 & 0 & 1 & -62 / 71
\end{array}\right]
$$

$\operatorname{rref}([a, b, x ; c, d, y])$ ENTER

$$
\left[\begin{array}{lll}
1 & 0 & \frac{d \cdot x-b \cdot y}{a \cdot d-b \cdot c} \\
0 & 1 & \frac{-(c \cdot x-a \cdot y)}{a \cdot d-b \cdot c}
\end{array}\right]
$$



SendCalc var[[port]
Sends contents of var from a TI-89 Titanium to another TI-89 Titanium.

If the port is not specified, or port $=0$ is specified, the TI-89 Titanium sends data using the USB port if connected, if not, it will send using the I/O port.

If port = 1, the TI-89 Titanium sends data using the USB port only.

If port $=2$, the TI-89 Titanium sends data using the I/O port only.

## SendChat CATALOG

SendChat var
A general alternative to SendCalc, this is useful if the receiving unit is a TI-92 (or for a generic "chat" program that allows either a TI-92, Voyage ${ }^{\text {TM }} 200$, or TI-92 Plus to be used). Refer to SendCalc for more information.

SendChat sends a variable only if that variable is compatible with the TI-92, which is typically true in "chat" programs. However, SendChat will not send an archived variable, a $\mathrm{TI}-89$ graph data base, etc.
seq() MATH/List menu
seq(expression, var, low, high[, step]) $\Rightarrow$ list
Increments var from low through high by an increment of step, evaluates expression, and returns the results as a list. The original contents of var are still there after seq() is completed. var cannot be a system variable.

The default value for $\operatorname{step}=1$.

Program segment:

```
        \vdots
    :a+b
    :SendChat x
```

    \(\operatorname{seq}\left(n^{\wedge} 2, n, 1,6\right)\) ENTER
        \(\left\{\begin{array}{llllll}1 & 4 & 9 & 16 & 25 & 36\end{array}\right\}\)
    \(\operatorname{seq}(1 / n, n, 1,10,2)\) ENTER
                                    \(\{1 \quad 1 / 3 \quad 1 / 5 \quad 1 / 7 \quad 1 / 9\}\)
    sum(seq(1/n^2,n,1,10,1)) ENTER
$\frac{196 \ldots}{127 \ldots}$
or press ENTER to get:
1.549...
setDate(year,month,day) $\Rightarrow$ listold
Sets the clock to the date given in the argument and returns a list. (Note: The year must fall in the range 1997-2132.) The returned list is in \{yearold,monthold, dayold\} format. The returned date is the previous clock value.
Enter the year as a four-digit integer. The month and day can be either one- or two-digit integers.

## setDtFmt() CATALOG

setDtFmt(integer) $\Rightarrow$ integerold
Sets the date format for the desktop according to the argument and returns the previous date format value.
setDate(2001,10,31) ENTER
$\{2001 \quad 11 \quad 1\}$

Integer values:
$1=M M / D D / Y Y$
$2=D D / M M / Y Y$
$3=M M . D D . Y Y$
$4=D D . M M . Y Y$
$5=Y Y . M M . D D$
$6=M M-D D-Y Y$
$7=D D-M M-Y Y$
$8=Y Y-M M-D D$

## setFold() CATALOG

setFold(newfolderName) $\Rightarrow$ oldfolderString
Returns the name of the current folder as a string and sets newfolderName as the current folder.

The folder newfolderName must exist.
newFold chris ENTER Done
setFold(main) ENTER "chris"
setFold(chris) $\rightarrow 01 d f o l d r$ ENTER
"main"
$1 \rightarrow$ a ENTER 1
setFold(非oldfoldr) ENTER "chris"
$a$ ENTER a
chris \a ENTER 1

## setGraph() CATALOG

setGraph(modeNameString, settingString) $\Rightarrow$ string
Sets the Graph mode modeNameString to settingString, and returns the previous setting of the mode. Storing the previous setting lets you restore it later.
modeNameString is a character string that specifies which mode you want to set. It must be one of the mode names from the table below.
settingString is a character string that specifies the new setting for the mode. It must be one of the settings listed below for the specific mode you are setting.
setGraph("Graph Order","Seq")
ENTER "SEQ"
setGraph("Coordinates","Off")
ENTER "RECT"
Note: Capitalization and blank spaces are optional when entering mode names.

| Mode Name | Settings |
| :--- | :--- |
| "Coordinates" | "Rect", "Polar", "Off" |
| "Graph Order" | "Seq", "Simul" 1 |

\(\left.\begin{array}{lll}\hline "Grid" \& "Off", "On"{ }^{2} <br>
\hline "Axes" \& "Off", "On" \& (not 3D graph mode) <br>

\& "Off", "Axes", "Box" \& (3D graph mode)\end{array}\right]\)| "Leading Cursor" | "Off", "On" ${ }^{2}$ | "Off", "On" |
| :--- | :--- | :--- |
| "Labels" | "Wire Frame", "Hidden Surface", "Contour Levels", "Wire and Contour", |  |
| "Style" | "Tmlicit Plot" 3 |  |

${ }^{1}$ Not available in Sequence, 3D, or Diff Equations graph mode.
${ }^{2}$ Not available in 3D graph mode.
${ }^{3}$ Applies only to 3D graph mode.
${ }^{4}$ Applies only to Sequence graph mode.
${ }^{5}$ Applies only to Diff Equations graph mode.

## setMode() CATALOG

```
setMode(modeNameString, settingString) }\quad=>\mathrm{ string
setMode(list) = stringList
```

Sets mode modeNameString to the new setting settingString, and returns the current setting of that mode.
modeNameString is a character string that specifies which mode you want to set. It must be one of the mode names from the table below.
settingString is a character string that speciiies the new setting for the mode. It must be one of the settings listed below for the specific mode you are setting.
list contains pairs of keyword strings and will set them all at once. This is recommended for multiplemode changes. The example shown may not work if each of the pairs is entered with a separate setMode() in the order shown.

Use setMode(var) to restore settings saved with getMode("ALL") $\rightarrow$ var.
Note: To set or return information about the Unit System mode, use setUnits() or getUnits() instead of setMode() or getMode().

```
setMode("Angle","Degree")
ENTER "RADIAN"
sin(45) ENTER
\sqrt{}{2}
setMode("Angle","Radian")
ENTER
sin(\pi/4) ENTER
\sqrt{}{2}
setMode("Display Digits",
"Fix 2") ENTER "FLOAT"
\pi ENTER
                                    3.14
setMode ("Display Digits",
"Float") ENTER "FIX 2"
\piENTER 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"}
```

Note: Capitalization and blank spaces are optional when entering mode names. Also, the results in these examples may be different on your unit.

| Mode Name | Settings |
| :--- | :--- |
| "Graph" | "Function", "Parametric", "Polar", "Sequence", "3D", "Diff Equations" |
| "Display 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 App" |

## setTable() CATALOG <br> setTable(modeNameString, settingString) $\Rightarrow$ string <br> Sets the table parameter modeNameString to settingString, and returns the previous setting of the parameter. Storing the previous setting lets you restore it later. <br> modeNameString is a character string that specifies which parameter you want to set. It must be one of the parameters from the table below. <br> settingString is a character string that specifies the new setting for the parameter. It must be one of the settings listed below for the specific parameter you are setting.

```
setTable("Graph <->
Table","ON")
ENTER "OFF"
setTable("Independent","AUT0")
ENTER
"ASK"
- [TbISet]
```



Note: Capitalization and blank spaces are optional when entering parameters.

| Parameter Name | Settings |
| :--- | :--- |
| "Graph <-> Table" | "Off", "On" |
| "Independent" | "Auto", "Ask" |

## setTime() CATALOG

setTime(hour,minute,second) $\Rightarrow$ listold
setTime(11,32,50)
$\left\{\begin{array}{lll}10 & 44 & 49\end{array}\right\}$
Sets the clock to the time given in the argument and returns a list. The list is in \{hourold, minuteold,secondold\} format. The returned time is the previous clock value.

Enter the hour in the 24 hour format, in which $13=$ 1 p.m.

## setTmFmt() CATALOG

setTmFmt(integer) $\Rightarrow$ integerold
Sets the time format for the desktop according to the argument and returns the previous time format value.

Integer values:
$12=12$ hour clock
$24=24$ hour clock

## setTmZn() CATALOG

setTmZn(integer) $\Rightarrow$ integerold
Sets the time zone according to the argument and returns the previous time zone value.
The time zone is defined by an integer that gives the minutes offset from Greenwich Mean Time (GMT), as established in Greenwich, England. For example, if the time zone is offset from GMT by two hours, the device would return 120 (minutes).

Integers for time zones west of GMT are negative.
Integers for time zones east of GMT are positive.

If Greenwich Mean Time is $14: 07: 07$, it is:
7:07:07 a.m. in Denver, Colorado (Mountain Standard Time)
(-420 minutes from GMT)
15:07:07 p.m. in Brussels, Belgium (Central European Standard Time)
(+60 minutes from GMT)

## setUnits() CATALOG

setUnits(list1) $\Rightarrow$ list
Sets the default units to the values specified in list1, and returns a list of the previous defaults.

- To specify the built-in SI (metric) or ENG/US system, list1 uses the form:
\{"SI"\} or \{"ENG/US"\}
- To specify a custom set of default units, list1 uses the form:
\{"CUSTOM", "cat1", "unit1"[, "cat2", "unit2", ...]\}
where each cat and unit pair specifies a category and its default unit. (You can specify built-in units only, not user-defined units.) Any category not specified will use its previous custom unit.

All unit names must begin with an underscore _.

[ [im [2nd [_]
You can also select units from a menu by pressing:


표 [i: [UNITS]
setUnits(\{"SI"\}) ENTER
\{"SI" "Area" "NONE"
"Capacitance" "_F" ...\}
setUnits(\{"CUSTOM", "Length",
"_cm", "Mass","_gm"\}) ENTER
$\{$ "SI" "Length" "_m"
"Mass" "_kg" ...\}
Note: Your screen may display different units.

- To return to the previous custom default units, list1 uses the form:
\{"CUSTOM"\}
If you want different defaults depending on the situation, create separate lists and save them to unique list names. To use a set of defaults, specify that list name in setUnits().

You can use setUnits() to restore settings previously saved with setUnits() $\rightarrow$ var or with getUnits() $\rightarrow$ var.

Shade expr1, expr2, [xlow], [xhigh], [pattern], [patRes]
Displays the Graph screen, graphs expr1 and expr2, and shades areas in which expr1 is less than expr2. (expr1 and expr2 must be expressions that use x as the independent variable.)
xlow and xhigh, if included, specify left and right boundaries for the shading. Valid inputs are between xmin and xmax. Defaults are xmin and xmax.
pattern specifies one of four shading patterns:
1 = vertical (default)
2 = horizontal
$3=$ negative-slope $45^{\circ}$
$4=$ positive-slope $45^{\circ}$
patRes specifies the resolution of the shading patterns:
1= solid shading
2=1 pixel spacing (default)
$3=2$ pixels spacing

10=9 pixels spacing
Note: Interactive shading is available on the Graph screen through the Shade instruction. Automatic shading of a specific function is available through the Style instruction. Shade is not valid in 3D graphing mode.

In the ZoomTrig viewing window:
Shade $\cos (x), \sin (x)$ ENTER

[- HOME
E [CALC HOME]
C1rDraw ENTER Done
Shade cos(x),sin(x),0,5 ENTER

$\begin{array}{ll}\text { HOME } \\ \text { Hin } & \text { [CALC HOME] }\end{array}$
ClrDraw ENTER Done
Shade $\cos (x), \sin (x), 0,5,2$ ENTER

$\begin{array}{ll}\text { HOME } \\ \text { Hin } & \text { [CALC HOME] }\end{array}$
ClrDraw ENTER Done
Shade $\cos (x), \sin (x), 0,5,2,1$
ENTER

shift(integer1[,\#ofShifts]) $\Rightarrow$ integer
Shifts the bits in a binary integer. You can enter integer1 in any number base; it is converted automatically to a signed, 32-bit binary form. If the magnitude of integer1 is too large for this form, a symmetric modulo operation brings it within the range.

If \#ofShifts is positive, the shift is to the left. If \#ofShifts is negative, the shift is to the right. The default is - 1 (shift right one bit).

In a right shift, the rightmost bit is dropped and 0 or 1 is inserted to match the leftmost bit. In a left shift, the leftmost bit is dropped and 0 is inserted as the rightmost bit.

For example, in a right shift:

Each bit shifts right
Ob00000000000001111010110000110101


Dropped or 1 if leftmost bit is 1 .
produces:
Ob00000000000000111101011000011010
The result is displayed according to the Base mode. Leading zeros are not shown.
shift(list1 [,\#ofShifts]) $\Rightarrow$ list
Returns a copy of list1 shifted right or left by \#ofShifts elements. Does not alter list1.

If \#ofShifts is positive, the shift is to the left. If \#ofShifts is negative, the shift is to the right. The default is - 1 (shift right one element).

Elements introduced at the beginning or end of list by the shift are set to the symbol "undef".

In Bin base mode:
shift(0b1111010110000110101) ENTER

Ob111101011000011010
shift(256,1) [ENTER
0b1000000000
In Hex base mode:

```
shift(0h78E) ENTER
shift(0h78E,-2) [ENTER

Important: To enter a binary or hexadecimal number, always use the Ob or Oh prefix (zero, not the letter O ).

In Dec base mode:
```

shift({1,2,3,4}) ENTER

```
\{undef 12 3\}
shift(\{1,2,3,4\},-2) ENTER
\{undef undef 12\(\}\)
shift(\{1,2,3,4\},1) ENTER
\{2 34 undef
shift(string1 [,\#ofShifts]) \(\Rightarrow\) string
Returns a copy of string1 shifted right or left by \#ofShifts characters. Does not alter string1.

If \#ofShifts is positive, the shift is to the left. If \#ofShifts is negative, the shift is to the right. The default is -1 (shift right one character).

Characters introduced at the beginning or end of string by the shift are set to a space.
shift("abcd") ENTER " abc"
shift("abcd",-2) ENTER " ab"
shift("abcd",1) ENTER "bcd "

ShowStat
Displays a dialog box containing the last computed statistics results if they are still valid. Statistics results are cleared automatically if the data to compute them has changed.

Use this instruction after a statistics calculation, such as LinReg.
\(\{1,2,3,4,5\} \rightarrow\) L1 ENTER \(\left\{\begin{array}{lllll}1 & 2 & 3 & 4 & 5\end{array}\right\}\)
\(\{0,2,6,10,25\} \rightarrow\) L2 ENTER
\(\{0261025\}\)
TwoVar L1, L2 ENTER
ShowStat ENTER


\section*{sign() MATH/Number menu}
sign(expression1) \(\Rightarrow\) expression
\(\boldsymbol{\operatorname { s i g n }}(\) list1) \(\Rightarrow\) list
\(\boldsymbol{\operatorname { s i g n }}(\) matrix1) \(\Rightarrow\) matrix
For real and complex expression1, returns expression \(1 / a b s(\) expression 1\()\) when expression \(1 \neq 0\).

Returns 1 if expression 1 is positive.
Returns - 1 if expression 1 is negative.
\(\boldsymbol{s i g n}(0)\) returns \(\pm 1\) if the complex format mode is REAL; otherwise, it returns itself. \(\boldsymbol{\operatorname { s i g n }}(0)\) represents the unit circle in the complex domain.

For a list or matrix, returns the signs of all the elements.

\section*{simult() MATH/Matrix menu}
simult(coeffMatrix, constVector[, tol]) \(\Rightarrow\) matrix
Returns a column vector that contains the solutions to a system of linear equations.
coeffMatrix must be a square matrix that contains the coefficients of the equations.
constVector must have the same number of rows (same dimension) as coeffMatrix and contain the constants.

Optionally, any matrix element is treated as zero if its absolute value is less than tol. This tolerance is used only if the matrix has floating-point entries and does not contain any symbolic variables that have not been assigned a value. Otherwise, tol is ignored.
- If you use ENTER or set the mode to Exact/Approx=APPROXIMATE, computations are done using floating-point arithmetic.
- If tol is omitted or not used, the default tolerance is calculated as:

5E-14*max(dim(coeffMatrix))
* rowNorm(coeffMatrix)
\(\operatorname{sign}(-3.2)\) ENTER -1.
\(\operatorname{sign}(\{2,3,4,-5\})\) ENTER
\(\left\{\begin{array}{llll}1 & 1 & 1 & -1\end{array}\right\}\)
\(\operatorname{sign}(1+a b s(x))\) ENTER 1

If complex format mode is REAL:
\(\operatorname{sign}([-3,0,3])\) ENTER \(\quad\left[\begin{array}{lll}-1 & \pm 1 & 1\end{array}\right]\)

Solve for \(x\) and \(y: \begin{aligned} x+2 y & =1 \\ 3 x+4 y & =-1\end{aligned}\)
simult \(([1,2 ; 3,4],[1 ;-1])\) ENTER
\(\left[\begin{array}{c}-3 \\ 2\end{array}\right]\)
The solution is \(\mathrm{x}=-3\) and \(\mathrm{y}=2\).

Solve: \(\quad a x+b y=1\)
\[
c x+d y=2
\]
\([a, b ; c, d] \rightarrow \operatorname{matx} 1\left[\right.\) ENTER \(\quad\left[\begin{array}{ll}a & b \\ c & d\end{array}\right]\)
simult(matx1,[1;2]) ENTER
\(\left[\begin{array}{c}\frac{-(2 \cdot b-d)}{a \cdot d-b \cdot c} \\ \frac{2 \cdot a-c}{a \cdot d-b \cdot c}\end{array}\right]\)
simult(coeffMatrix, constMatrix[, tol]) \(\Rightarrow\) matrix
Solves multiple systems of linear equations, where each system has the same equation coefficients but different constants.

Each column in constMatrix must contain the constants for a system of equations. Each column in the resulting matrix contains the solution for the corresponding system.
\begin{tabular}{|c|c|c|}
\hline \(\sin ()\) & [ [2nd [sin] key & - Cil \(_{\text {SIN key }}\) \\
\hline & pression1) \(\Rightarrow\) expression & \\
\hline
\end{tabular}
\(\boldsymbol{\operatorname { s i n }}\) (expression1) returns the sine of the argument as an expression.
\(\boldsymbol{\operatorname { s i n }}\) (list1) returns a list of the sines of all elements in list1.

Note: The argument is interpreted as either a degree or radian angle, according to the current angle mode. You can use \({ }^{\circ}\) or \({ }^{r}\) to override the angle mode setting temporarily.

\section*{\(\boldsymbol{\operatorname { s i n }}\) (squareMatrix1) \(\Rightarrow\) squareMatrix}

Returns the matrix sine of squareMatrix1. This is not the same as calculating the sine of each element. For information about the calculation method, refer to \(\boldsymbol{\operatorname { c o s }}(\) ).
squareMatrix1 must be diagonalizable. The result always contains floating-point numbers.

\section*{} \(\boldsymbol{\operatorname { s i n }}^{-1}\) (expression1) \(\Rightarrow\) expression \(\boldsymbol{\operatorname { s i n }}^{-1}\) (list1) \(\Rightarrow\) list
\(\boldsymbol{s i n}^{-1}\) (expression1) returns the angle whose sine is expression1 as an expression.
\(\boldsymbol{s i n}^{-1}\) (list1) returns a list of the inverse sines of each element of list1.

Note: The result is returned as either a degree or radian angle, according to the current angle mode setting.
```

    sin}\mp@subsup{}{}{-1}(\mathrm{ squareMatrix1) }=>\mathrm{ squareMatrix
    ```

Returns the matrix inverse sine of squareMatrix1. This is not the same as calculating the inverse sine of each element. For information about the calculation method, refer to \(\boldsymbol{\operatorname { c o s }}(\mathbf{)}\).
squareMatrix1 must be diagonalizable. The result always contains floating-point numbers.

Solve: \(\begin{array}{rlrl}x+2 y & =1 & x+2 y & =2 \\ 3 x+4 y & =-1 & 3 x+4 y & =-3\end{array}\)
simult([1,2;3,4],[1,2;-1,-3]) ENTER \(\left[\begin{array}{cc}-3 & -7 \\ 2 & 9 / 2\end{array}\right]\)

For the first system, \(x=-3\) and \(y=2\). For the second system, \(x=-7\) and \(y=9 / 2\).

In Degree angle mode:
\begin{tabular}{lr}
\(\sin \left((\pi / 4)^{r}\right)\) ENTER & \(\frac{\sqrt{ } 2}{2}\) \\
\(\sin (45)\) ENTER & \(\frac{\sqrt{2}}{2}\) \\
\(\sin (\{0,60,90\})\) ENTER \(\left\{0 \quad \frac{\sqrt{3}}{2}\right.\) & 1
\end{tabular}

In Radian angle mode:
\(\sin (\pi / 4)\) ENTER \(\frac{\sqrt{2}}{2}\)
\(\sin \left(45^{\circ}\right)\) ENTER \(\frac{\sqrt{2}}{2}\)
In Radian angle mode:
\(\sin ([1,5,3 ; 4,2,1 ; 6,-2,1])\) ENTER
\(\left[\begin{array}{lll}.942 \ldots & -.045 \ldots & -.031 \ldots \\ -.045 \ldots . & .949 \ldots & -.020 . . \\ -.048 . . . & -.005 \ldots & .961 \ldots\end{array}\right]\)

In Degree angle mode:
\(\sin ^{-1}\) (1) ENTER
In Radian angle mode:
\(\sin ^{-1}(\{0, .2, .5\})\) ENTER
\{0 . 201... .523...\}

In Radian angle mode and Rectangular complex format mode:
\(\sin ^{-1}([1,5,3 ; 4,2,1 ; 6,-2,1])\)
ENTER
\(\left[\begin{array}{lll}-.164 \ldots-. .064 \ldots \cdot . . \boldsymbol{i} & 1.490 \ldots-2.105 \ldots \cdot \boldsymbol{i} & \ldots \\ . . . \\ 725 \ldots-1.515 \ldots \cdot \boldsymbol{i} & .947 \ldots-.778 \ldots \cdot \boldsymbol{i} & \ldots \\ 2.083 \ldots-2.632 \ldots \cdot \boldsymbol{i} & -1.790 \ldots+1.271 \ldots \cdot . & \ldots\end{array}\right]\)
\(\sinh () \quad\) MATH/Hyperbolic menu
\(\boldsymbol{\operatorname { s i n h }}\) (expression1) \(\Rightarrow\) expression \(\boldsymbol{\operatorname { s i n h }}\) (list1) \(\Rightarrow\) list
sinh (expression1) returns the hyperbolic sine of the argument as an expression.
sinh (list) returns a list of the hyperbolic sines of each element of list1.
\(\boldsymbol{\operatorname { s i n h }}\) (squareMatrix1) \(\Rightarrow\) squareMatrix
Returns the matrix hyperbolic sine of squareMatrix1. This is not the same as calculating the hyperbolic sine of each element. For information about the calculation method, refer to \(\boldsymbol{\operatorname { c o s }}(\mathbf{)}\).
squareMatrix 1 must be diagonalizable. The result always contains floating-point numbers.
\(\sinh (1.2)\) ENTER
1.509...
\(\sinh (\{0,1.2,3\}\).\() ENTER\)
\{0 1.509... 10.017...\}

\section*{\(\boldsymbol{s i n h}^{-1}\) () MATH/Hyperbolic menu}
sinh \(^{-1}\) (expression1) \(\Rightarrow\) expression \(\boldsymbol{s i n h}^{-1}\) (list1) \(\Rightarrow\) list
\(\boldsymbol{s i n h}^{-1}\) (expression1) returns the inverse hyperbolic sine of the argument as an expression.
\(\boldsymbol{s i n h}^{-1}\) (list1) returns a list of the inverse hyperbolic sines of each element of list1.
\(\boldsymbol{s i n h}^{-1}\) (squareMatrix1) \(\Rightarrow\) squareMatrix
Returns the matrix inverse hyperbolic sine of squareMatrix1. This is not the same as calculating the inverse hyperbolic sine of each element. For information about the calculation method, refer to \(\boldsymbol{\operatorname { c o s }}(\) ).
squareMatrix 1 must be diagonalizable. The result always contains floating-point numbers.

In Radian angle mode:
\(\sinh ([1,5,3 ; 4,2,1 ; 6,-2,1])\)
ENTER
\(\left[\begin{array}{lll}360.954 & 305.708 & 239.604 \\ 352.912 & 233.495 & 193.564 \\ 298.632 & 154.599 & 140.251\end{array}\right]\)
\(\sinh ^{-1}(0)\) ENTER 0
\(\sinh ^{-1}(\{0,2.1,3\})\) ENTER
\{0 1.487... sinh \(\left.^{-1}(3)\right\}\)

In Radian angle mode:
\(\sinh ^{-1}([1,5,3 ; 4,2,1 ; 6,-2,1])\)
ENTER
\(\left[\begin{array}{lll}.041 \ldots . & 2.155 \ldots & 1.158 \ldots . . \\ 1.463 . . . & .926 \ldots & .112 \ldots \\ 2.750 . . . & -1.528 . . . & .572 \ldots\end{array}\right]\)

SinReg list1, list2 [, [iterations], [ period] [, list3, list4]]
Calculates the sinusoidal regression and updates al the system statistics variables.

All the lists must have equal dimensions except for list4.
list1 represents xlist.
list2 represents ylist. list3 represents category codes.
list4 represents category include list.
iterations specifies the maximum number of times (1 through 16) a solution will be attempted. If omitted, 8 is used. Typically, larger values result in better accuracy but longer execution times, and vice versa.
period specifies an estimated period. If omitted, the difference between values in list1 should be equal and in sequential order. If you specify period, the differences between \(x\) values can be unequal.

Note: list1 through list3 must be a variable name or c1-c99 (columns in the last data variable shown in the Data/Matrix Editor). list4 does not have to be a variable name and cannot be c1-c99.

The output of SinReg is always in radians, regardless of the angle mode setting.

In function graphing mode:
\(\operatorname{seq}(x, x, 1,361,30) \rightarrow L 1\) ENTER
\(\left\{\begin{array}{llll}1 & 31 & 61 & \text {... }\end{array}\right\}\)
\(\{5.5,8,11,13.5,16.5,19,19.5,17\),
\(14.5,12.5,8.5,6.5,5.5\} \rightarrow\) L2 ENTER
\(\{5.5811\)... \(\}\)
SinReg L1,L2 ENTER Done
ShowStat ENTER


\section*{ENTER}
regeq \((x) \rightarrow y 1(x)\) ENTER Done
NewPlot 1,1,L1,L2 ENTER Done - [GRAPH]

F2 9

solve(a* \(\left.x^{\wedge} 2+b * x+c=0, x\right)\) ENTER


ans(1)| \(a=1\) and \(b=1\) and \(c=1\) ENTER

Error: Non-real result
solve( \(\left.(x-a) e^{\wedge}(x)=-x *(x-a), x\right)\)
ENTER
\[
x=a \text { or } x=-.567 \ldots
\]
\((x+1)(x-1) /(x-1)+x-3\) ENTER
\begin{tabular}{lr} 
solve(entry \((1)=0, x)\) ENTER & \(2 \cdot x-2\) \\
entry \((2) \mid \operatorname{ans}(1)\) ENTER & undef \\
limit(entry \((3), x, 1)\) ENTER & 0 \\
solve( \(5 x-2 \geq 2 x, x)\) ENTER & \(x \geq 2 / 3\)
\end{tabular}
exact (solve( (x-a) \(e^{\wedge}(x)=-x *\) \((x-a), x))\) ENTER
\[
e^{x}+x=0 \text { or } x=a
\]

Use the "l" operator to restrict the solution interval and/or other variables that occur in the equation or inequality. When you find a solution in one interval, you can use the inequality operators to exclude that interval from subsequent searches.
false is returned when no real solutions are found. true is returned if solve() can determine that any finite real value of var satisfies the equation or inequality.

Since solve() always returns a Boolean result, you can use "and," "or," and "not" to combine results from solve() with each other or with other Boolean expressions.

Solutions might contain a unique new undefined variable of the form @nj with \(j\) being an integer in the interval 1-255. Such variables designate an arbitrary integer.

In real mode, fractional powers having odd denominators denote only the real branch. Otherwise, multiple branched expressions such as fractional powers, logarithms, and inverse trigonometric functions denote only the principal branch. Consequently, solve() produces only solutions corresponding to that one real or principal branch.

In Radian angle mode:
```

solve( $\tan (x)=1 / x, x) \mid x>0$ and $x<1$
ENTER $x=.860 \ldots$

```
solve \((x=x+1, x)\) ENTER false
solve \((x=x, x)\) ENTER true
\(2 x-1 \leq 1\) and solve \(\left(x^{\wedge} 2 \neq 9, x\right)\) ENTER
\(x \leq 1\) and \(x \neq-3\)

In Radian angle mode:
solve( \(\sin (x)=0, x)\) ENTER \(x=@ n 1 \cdot \pi\)
solve \(\left(x^{\wedge}(1 / 3)=-1, x\right)\) ENTER \(\quad x=-1\)
solve( \(\sqrt{ }(x)=-2, x)\) ENTER false
solve \((-\sqrt{ }(x)=-2, x)\) ENTER \(\quad x=4\)

Note: See also cSolve(), cZeros(), nSolve(), and zeros().
solve(equation1 and equation2 [and ... ], \{varOrGuess1, varOrGuess2 \([, \ldots]\}\) ) \(\Rightarrow\) Boolean expression

Returns candidate real solutions to the simultaneous algebraic equations, where each varOrGuess specifies a variable that you want to solve for.

Optionally, you can specify an initial guess for a variable. Each varOrGuess must have the form:
variable
- or -
variable \(=\) real or non-real number
For example, x is valid and so is \(\mathrm{x}=3\).
solve( \(y=x^{\wedge} 2-2\) and
\(x+2 y=-1,\{x, y\})\) ENTER
\(\mathrm{x}=1\) and \(\mathrm{y}=-1\) or \(x=-3 / 2\) and \(y=1 / 4\)

If all of the equations are polynomials and if you do NOT specify any initial guesses, solve() uses the lexical Gröbner/Buchberger elimination method to attempt to determine all real solutions.

For example, suppose you have a circle of radius \(r\) at the origin and another circle of radius \(r\) centered where the first circle crosses the positive x -axis. Use solve() to find the intersections.

As illustrated by \(r\) in the example to the right, simultaneous polynomial equations can have extra variables that have no values, but represent given numeric values that could be substituted later.

You can also (or instead) include solution variables that do not appear in the equations. For example, you can include \(z\) as a solution variable to extend the previous example to two parallel intersecting cylinders of radius \(r\).

The cylinder solutions illustrate how families of solutions might contain arbitrary constants of the form @ \(k\), where \(k\) is an integer suffix from 1 through 255. The suffix resets to 1 when you use CIrHome or F1 8:Clear Home.

For polynomial systems, computation time or memory exhaustion may depend strongly on the order in which you list solution variables. If your initial choice exhausts memory or your patience, try rearranging the variables in the equations and/or varOrGuess list.

If you do not include any guesses and if any equation is non-polynomial in any variable but all equations are linear in the solution variables, solve() uses Gaussian elimination to attempt to determine all real solutions.

If a system is neither polynomial in all of its variables nor linear in its solution variables, solve() determines at most one solution using an approximate iterative method. To do so, the number of solution variables must equal the number of equations, and all other variables in the equations must simplify to numbers.

\[
\left.\begin{array}{l}
\text { solve }\left(x^{\wedge} 2+y^{\wedge} 2=r^{\wedge} 2\right. \text { and } \\
(x-r)^{\wedge} 2+y^{\wedge} 2=r^{\wedge} 2,
\end{array} \quad\{x, y\}\right) \text { ENTER } .
\]
\[
\begin{aligned}
& \text { Solve }\left(x^{\wedge} 2+y^{\wedge} 2=r^{\wedge} 2\right. \text { and } \\
& \begin{array}{c}
\left.(x-r)^{\wedge} 2+y^{\wedge} 2=r^{\wedge} 2,\{x, y, z\}\right) \text { ENTER } \\
\\
x=\frac{r}{2} \text { and } y=\frac{\sqrt{3} \cdot r}{2} \text { and } z=@ 1 \\
\text { or } x
\end{array}=\frac{r}{2} \text { and } y=\frac{-\sqrt{3} \cdot r}{2} \text { and } z=@ 1
\end{aligned}
\]
solve( \(x+e^{\wedge}(z) * y=1\) and
\(x-y=\sin (z),\{x, y\})\) ENTER
\(\mathrm{x}=\frac{e^{z} \cdot \sin (\mathrm{z})+1}{e^{\mathrm{z}+1}}\) and \(\mathrm{y}=\frac{-(\sin (\mathrm{z})-1)}{e^{z+1}}\)
solve( \(e^{\wedge}(z) * y=1\) and
\(-\mathrm{y}=\sin (\mathrm{z}),\{\mathrm{y}, \mathrm{z}\})\) ENTER
\[
y=.041 \ldots \text { and } z=3.183 \ldots
\]

Each solution variable starts at its guessed value if there is one; otherwise, it starts at 0.0 .

Use guesses to seek additional solutions one by one. For convergence, a guess may have to be rather close to a solution.
solve( \(e^{\wedge}(z) * y=1\) and
\(-y=\sin (z),\{y, z=2 \pi\})\) ENTER \(y=.001 \ldots\) and \(z=6.281 \ldots\)

\section*{SortA MATH/List menu}

SortA listName1[, listName2] [, listName3] ...
SortA vectorName1[, vectorName2] [, vectorName3] ...
Sorts the elements of the first argument in ascending order.

If you include additional arguments, sorts the elements of each so that their new positions match the new positions of the elements in the first argument.
All arguments must be names of lists or vectors. All arguments must have equal dimensions.
\(\{2,1,4,3\} \rightarrow 1\) ist1 ENTER \(\{2,1,4,3\}\)
SortA list1 ENTER Done
1 ist1 ENTER \(\left\{\begin{array}{llll}1 & 2 & 3 & 4\end{array}\right\}\)
\(\{4,3,2,1\} \rightarrow 1\) ist2 ENTER \(\quad\left\{\begin{array}{llll}4 & 3 & 2 & 1\end{array}\right\}\)
SortA 1ist2,1ist1 ENTER Done
\(\left.\begin{array}{ll}1 \text { ist2 ENTER } \\ 1 \text { ist1 ENTER } & \left\{\begin{array}{llll}1 & 2 & 3 & 4\end{array}\right\} \\ 4 & 3\end{array} 2 \begin{array}{l}1\end{array}\right\}\)

SortD MATH/List menu
SortD listName1[, listName2] [, listName3] ...
SortD vectorName1[,vectorName 2] [,vectorName 3] ...
Identical to SortA, except SortD sorts the elements in descending order.

\section*{Sphere MATH/Matrix/Vector ops menu}

\section*{vector Sphere}

Displays the row or column vector in spherical form [ \(\rho \angle \theta \angle \phi\) ].
vector must be of dimension 3 and can be either a row or a column vector.
Note: \(>\) Sphere is a display-format instruction, not a conversion function. You can use it only at the end of an entry line.
\begin{tabular}{|c|c|}
\hline \(\{2,1,4,3\} \rightarrow 1\) ist1 ENTER & \(\left\{\begin{array}{llll}2 & 1 & 4 & 3\end{array}\right\}\) \\
\hline \(\{1,2,3,4\} \rightarrow 1\) ist2 ENTER & \(\left\{\begin{array}{llll}1 & 2 & 3 & 4\end{array}\right\}\) \\
\hline SortD list1,1ist2 ENTER & Done \\
\hline 1ist1 ENTER & \{lllll \(\left.\begin{array}{llll}4 & 2 & 1\end{array}\right\}\) \\
\hline 1 ist2 ENTER & \(\left\{\begin{array}{llll}3 & 4 & 1 & 2\end{array}\right\}\) \\
\hline
\end{tabular}
[1,2,3]Sphere
© ENTER [3.741... \(\angle 1.107 \ldots \angle .640 \ldots]\)
[2, \(\angle \pi / 4,3]\) Sphere
\(\rightarrow\) ENTER [3.605... \(\angle .785 \ldots \angle\). 588...]
ENTER \(\left[\sqrt{ } 13 \quad \angle \frac{\pi}{4} \quad \angle \cos ^{-1}\left(\frac{3 \cdot \sqrt{13}}{13}\right)\right]\)


\section*{startTmr() CATALOG}
startTmr() \(\Rightarrow\) integer
Returns the current value of the clock in its integer representation, giving the startime for a timer. You can enter the starttime as an argument in checkTmr() to determine how many seconds have elapsed.
You can run multiple timers simultaneously.
Note: See also checkTmr() and timeCnv().
startTmr() ENTER
148083315
checkTmr(148083315)
34
```

startTmr()->Timer1
\vdots
startTmr()->Timer2
:
checkTmr(Timer1)->TimerIValue
\vdots
checkTmr(Timer2)->Timer2Value

```
stdDev(list[, freqlisf]) \(\Rightarrow\) expression
Returns the standard deviation of the elements in list.

Each freqlist element counts the number of consecutive occurrences of the corresponding element in list.

Note: list must have at least two elements.

Returns a row vector of the standard deviations of the columns in matrix1.

Each freqmatrix element counts the number of consecutive occurrences of the corresponding element in matrix1.

Note: matrix1 must have at least two rows.

\section*{StoGDB CATALOG}

\section*{StoGDB GDBvar}

Creates a Graph database (GDB) variable that contains the current:
* Graphing mode
* \(Y=\) functions
* Window variables
* Graph format settings 1- or 2-Graph setting (split screen and ratio settings if 2-Graph mode) Angle mode Real/complex mode
* Initial conditions if Sequence or Diff Equations mode
* Table flags
* tblStart, \(\Delta \mathrm{tbl}\), tbllnput

You can use RcIGDB GDBvar to restore the graph environment.
*Note: These items are saved for both graphs in 2Graph mode.

\section*{Stop CATALOG}

\section*{Stop}

Used as a program instruction to stop program execution.
stdDev (\{a,b,c\}) ENTER
stdDev(\{1,2,5,-6,3,-2\}) ENTER

\(\operatorname{stdDev}(\{1.3,2.5,-6.4\},\{3,2,5\})\)
ENTER
4.33345
stdDev([1, 2,5;-3, 0, 1; .5,.7,3]) ENTER
\[
\left[\begin{array}{ll}
2.179 \ldots & 1.014 \ldots
\end{array}\right]
\]
stdDev([-1.2,5.3;2.5,7.3;6,-4], \([4,2 ; 3,3 ; 1,7])\) ENTER
[2.7005,5.44695]

Program segment:
```

For i,1,10,1
If i=5
Stop
EndFor
\vdots

```

\section*{StoPic CATALOG}

StoPic picVar [, pxIRow, pxICol] [, width, height]
Displays the graph screen and copies a rectangular area of the display to the variable picVar.
pxIRow and pxICol, if included, specify the upper-left corner of the area to copy (defaults are 0,0 ).
width and height, if included, specify the dimensions, in pixels, of the area. Defaults are the width and height, in pixels, of the current graph screen.

\section*{Store \(\quad\) See \(\rightarrow\) (store), page 885.}

\section*{string() MATH/String menu}
string(expression) \(\Rightarrow\) string
Simplifies expression and returns the result as a character string.
\[
\begin{aligned}
& \text { string(1.2345) ENTER } \\
& \text { "1.2345" } \\
& \text { string }(1+2) \text { ENTER "3" } \\
& \text { string }(\cos (x)+\sqrt{ }(3)) \text { ENTER } \\
& " \cos (x)+\sqrt{ }(3) "
\end{aligned}
\]

\section*{Style catalog}

Style equanum, stylePropertyString
Sets the system graphing function equanum in the current graph mode to use the graphing property stylePropertyString.
equanum must be an integer from 1-99 and the function must already exist.
stylePropertyString must be one of: "Line", "Dot", "Square", "Thick", "Animate", "Path", "Above", or "Below".

Note that in parametric graphing, only the \(x t\) half of the pair contains the style information

Valid style names vs. graphing mode:
Function: all styles
Parametric/Polar: line, dot, square, thick, animate, path Sequence: line, dot, square, thick 3D: none Diff Equations: line, dot, square, thick, animate, path

Note: Capitalization and blank spaces are optional when entering stylePropertyString names.

\section*{subMat() CATALOG}
subMat(matrix1[, startRow] [, startCol] [, endRow] [, endCol]) \(\quad \Rightarrow\) matrix

Returns the specified submatrix of matrix1.
Defaults: startRow=1, startCol=1, endRow=last row, endCol=last column

Style 1,"thick" ENTER
Done
Style 10,"path" ENTER
Done
Note: In function graphing mode, these
examples set the style of \(\mathrm{y} 1(\mathrm{x})\) to "Thick" and y10(x) to "Path".
\([1,2,3 ; 4,5,6 ; 7,8,9] \rightarrow\) m1 ENTER


456
\(\left.\begin{array}{lll}7 & 8 & 9\end{array}\right]\)
subMat(m1,2,1,3,2) ENTER
subMat(m1,2,2) ENTER
\(\left[\begin{array}{ll}5 & 6\end{array}\right]\)

\section*{sum() MATH/List menu}
\[
\operatorname{sum}(\text { list[, start[, end]]) } \Rightarrow \text { expression }
\]

Returns the sum of the elements in list.
Start and end are optional. They specify a range of elements.
\(\operatorname{sum}(\{1,2,3,4,5\})\) ENTER 15
\(\operatorname{sum}(\{a, 2 a, 3 a\})\) ENTER 6•a
\(\operatorname{sum}(\operatorname{seq}(\mathrm{n}, \mathrm{n}, 1,10))\) ENTER 55
\(\operatorname{sum}(\{1,3,5,7,9\}, 3)\) ENTER 21
\(\operatorname{sum}(\) matrix \(1[\), startI, end]]]) \(\Rightarrow\) matrix
Returns a row vector containing the sums of the elements in the columns in matrix1.
\(\operatorname{sum}([1,2,3 ; 4,5,6])\) ENTER \([57 c c]\)
sum( \([1,2,3 ; 4,5,6 ; 7,8,9])\) ENTER
\(\left[\begin{array}{lll}12 & 15 & 18\end{array}\right]\)
Start and end are optional. They specify a range of rows.
[11, 13, 15]

\section*{switch() CATALOG}
switch([integer1]) \(\Rightarrow\) integer
Returns the number of the active window. Also can set the active window.

Note: Window 1 is left or top; Window 2 is right or bottom.

If integer1 \(=0\), returns the active window number.
If integer \(=1\), activates window 1 and returns the previously active window number.

If integer1 = 2, activates window 2 and returns the previously active window number.

If integer1 is omitted, switches windows and returns the previously active window number.
integer1 is ignored if the
TI-89 Titanium/Voyage \({ }^{\text {TM }} 200\) is not displaying a split screen.

switch () ENTER

\([1,2,3 ; 4,5,6 ; 7,8,9] \rightarrow\) mat1 ENTER
\(\left[\begin{array}{lll}1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9\end{array}\right]\)
mat1 \({ }^{\top}\) ENTER
[a,b;c,d] mat2 ENTER
mat2 \({ }^{\top}\) ENTER
\([1+i, 2+i ; 3+i, 4+i] \rightarrow\) mat3 ENTER
\(\left[\begin{array}{cc}1+\boldsymbol{i} & 2+\boldsymbol{i} \\ 3+\boldsymbol{i} & 4+\boldsymbol{i}\end{array}\right]\)
mat3 \({ }^{\top}\) ENTER
\[
\left[\begin{array}{cc}
1-\boldsymbol{i} & 3-\boldsymbol{i} \\
2-\boldsymbol{i} & 4-\boldsymbol{i}
\end{array}\right]
\]

Table CATALOG

Table expression1[, expression2] [, var1]
Builds a table of the specified expressions or functions.

The expressions in the table can also be graphed. Expressions entered using the Table or Graph commands are assigned increasing function numbers starting with 1 . The expressions can be modified or individually deleted using the edit functions available when the table is displayed by pressing [F4 Header. The currently selected functions in the \(\mathrm{Y}=\) Editor are temporarily ignored.

To clear the functions created by Table or Graph, execute the CIrGraph command or display the \(\mathrm{Y}=\) Editor.

In function graphing mode.
Table 1.25x* \(\cos (x)\) ENTER


Table cos(time), time ENTER


If the var parameter is omitted, the current graphmode independent variable is assumed. Some valid variations of this instruction are:

Function graphing: Table expr, \(x\)
Parametric graphing: Table \(x\) Expr, yExpr, \(t\)
Polar graphing: Table expr, \(\theta\)
Note: The Table command is not valid for 3D, sequence, or diff equations graphing. As an alternative, you may want to use BIdData.
\begin{tabular}{|c|c|c|}
\hline \(\tan ()\) & 2nd [TAN] key Eim TAN key & \\
\hline & \begin{tabular}{l}
\(\boldsymbol{\operatorname { t a n }}\) (expression1) \(\Rightarrow\) expression \\
\(\boldsymbol{\operatorname { t a n }}\) (list1) \(\Rightarrow\) list \\
\(\boldsymbol{\operatorname { t a n }}\) (expression1) returns the tangent of the argument as an expression. \\
\(\boldsymbol{\operatorname { t a n }}\) (list1) returns a list of the tangents of all elements in list1. \\
Note: The argument is interpreted as either a degree or radian angle, according to the current angle mode. You can use \({ }^{\circ}\) or \({ }^{r}\) to override the angle mode temporarily.
\end{tabular} & \begin{tabular}{l}
In Degree angle mode: \\
In Radian angle mode: \\
\(\tan (\pi / 4)\) ENTER \\
\(\tan \left(45^{\circ}\right)\) ENTER \\
\(\tan (\{\pi, \pi / 3,-\pi, \pi / 4\})\) ENTER
\[
\left\{\begin{array}{lll}
0 & \sqrt{3} & 0
\end{array} 1\right\}
\]
\end{tabular} \\
\hline & \begin{tabular}{l}
\(\boldsymbol{\operatorname { t a n }}\) (squareMatrix1) \(\Rightarrow\) squareMatrix \\
Returns the matrix tangent of squareMatrix1. This is not the same as calculating the tangent of each element. For information about the calculation method, refer to \(\boldsymbol{\operatorname { c o s } ( ) .}\) \\
squareMatrix1 must be diagonalizable. The result always contains floating-point numbers.
\end{tabular} & In Radian angle mode: \(\tan ([1,5,3 ; 4,2,1 ; 6,-2,1])\) ENTER
\[
\left[\begin{array}{lll}
-28.291 . . . & 26.088 \ldots . . & 11.114 \ldots . \\
12.117 \ldots . . & -7.835 . . & -5.481 . . \\
36.818 \ldots . & -32.806 \ldots & -10.459 \ldots
\end{array}\right]
\] \\
\hline \(\tan ^{-1}(\) & 园 [TAN-1] key nemerien [TAN-1] key & \\
\hline & \begin{tabular}{l}
\(\boldsymbol{t a n}^{-1}\) (expression1) \(\Rightarrow\) expression \\
\(\boldsymbol{t a n}^{-1}(\) list1) \(\Rightarrow\) list \\
\(\boldsymbol{t a n}^{-1}\) (expression1) returns the angle whose tangent is expression1 as an expression. \\
\(\boldsymbol{t a n}^{-1}\) (list1) returns a list of the inverse tangents of each element of list1. \\
Note: The result is returned as either a degree or radian angle, according to the current angle mode setting.
\end{tabular} & \begin{tabular}{l}
In Degree angle mode: \\
\(\tan ^{-1}(1)\) ENTER \\
In Radian angle mode: \\
\(\tan ^{-1}(\{0, .2, .5\})\) ENTER \\
\{0 .197... .463...\}
\end{tabular} \\
\hline & \begin{tabular}{l}
\(\boldsymbol{t a n}^{-1}\) (squareMatrix1) \(\Rightarrow\) squareMatrix \\
Returns the matrix inverse tangent of squareMatrix1. This is not the same as calculating the inverse tangent of each element. For information about the calculation method, refer to \(\boldsymbol{\operatorname { c o s }}()\). \\
squareMatrix1 must be diagonalizable. The result always contains floating-point numbers.
\end{tabular} & \begin{tabular}{l}
In Radian angle mode:
\[
\tan ^{-1}([1,5,3 ; 4,2,1 ; 6,-2,1])
\] \\
ENTER
\[
\left[\begin{array}{lll}
-.083 . . . & 1.266 \ldots . . & .622 \ldots . \\
.748 . . . & .630 \ldots & -.070 \ldots \\
1.686 . . . & -1.182 \ldots & .455 \ldots
\end{array}\right]
\]
\end{tabular} \\
\hline tanh() & MATH/Hyperbolic menu & \\
\hline & \begin{tabular}{l}
\(\boldsymbol{\operatorname { t a n h }}\) (expression1) \(\Rightarrow\) expression \\
\(\boldsymbol{\operatorname { t a n h }}\) (list1) \(\Rightarrow\) list \\
\(\boldsymbol{t a n h}\) (expression1) returns the hyperbolic tangent of the argument as an expression. \\
\(\boldsymbol{\operatorname { t a n h }}\) (list) returns a list of the hyperbolic tangents of each element of list1.
\end{tabular} & \(\tanh (1.2)\) ENTER
\(\tanh (\{0,1\})\) ENTER \(\{0 \quad \tanh (1)\}\) \\
\hline
\end{tabular}
\(\tanh (\) squareMatrix1) \(\Rightarrow\) squareMatrix
Returns the matrix hyperbolic tangent of squareMatrix1. This is not the same as calculating the hyperbolic tangent of each element. For information about the calculation method, refer to \(\boldsymbol{\operatorname { c o s } ( ) .}\)
squareMatrix1 must be diagonalizable. The result always contains floating-point numbers.

In Radian angle mode
\(\tanh ([1,5,3 ; 4,2,1 ; 6,-2,1])\)
ENTER
\[
\left[\begin{array}{lll}
-.097 \ldots . . & .933 \ldots & .425 \ldots \\
.488 \ldots . . & .538 \ldots & -.129 \ldots \\
1.282 \ldots . . & -1.034 \ldots & .428 . . .
\end{array}\right]
\]

\section*{tanh \({ }^{-1}\) () MATH/Hyperbolic menu}
```

tanh-1}\mp@subsup{}{}{-1}\mathrm{ (expression1) }=>\mathrm{ expression
tanh-1 (list1) = list

```
\(\boldsymbol{t a n h}^{-1}\) (expression1) returns the inverse hyperbolic tangent of the argument as an expression.
\(\boldsymbol{t a n h}^{-1}\) (list1) returns a list of the inverse hyperbolic tangents of each element of list1.
\(\tanh ^{-1}\) (squareMatrix1) \(\Rightarrow\) squareMatrix
Returns the matrix inverse hyperbolic tangent of squareMatrix1. This is not the same as calculating the inverse hyperbolic tangent of each element. For information about the calculation method, refer to \(\cos ()\).
squareMatrix1 must be diagonalizable. The result always contains floating-point numbers.

\section*{taylor() MATH/Calculus menu}
taylor(expression1, var, order[, poinf]) \(\Rightarrow\) expression
Returns the requested Taylor polynomial. The polynomial includes non-zero terms of integer degrees from zero through order in (var minus point). taylor() returns itself if there is no truncated power series of this order, or if it would require negative or fractional exponents. Use substitution and/or temporary multiplication by a power of (varminus point) to determine more general power series.
point defaults to zero and is the expansion point.

In rectangular complex format mode:
tanh \(^{-1}\) (0) ENTER
0
\(\tanh ^{-1}(\{1,2.1,3\})\) ENTER
\(\left\{\infty \quad .518 \ldots-1.570 \ldots \cdot i \frac{\ln (2)}{2}-\frac{\pi}{2} \cdot i\right\}\)

In Radian angle mode and Rectangular complex format mode:
\(\tanh ^{-1}([1,5,3 ; 4,2,1 ; 6,-2,1])\)
ENTER
taylor \(\left(e^{\wedge}(\sqrt{ }(x)), x, 2\right)\) ENTER
taylor \(\left(e^{\wedge}(t), t, 4\right) \mid t=\sqrt{(x)}\) ENTER

taylor(1/(x*(x-1)), x,3) ENTER

expand(taylor(x/(x*(x-1)),
x,4)/x,x) ENTER

tCollect(expression1) \(\Rightarrow\) expression
Returns an expression in which products and integer powers of sines and cosines are converted to a linear combination of sines and cosines of multiple angles, angle sums, and angle differences. The transformation converts trigonometric polynomials into a linear combination of their harmonics.

Sometimes tCollect() will accomplish your goals when the default trigonometric simplification does not. tCollect() tends to reverse transformations done by tExpand(). Sometimes applying tExpand() to a result from tCollect(), or vice versa, in two separate steps simplifies an expression.

\section*{tExpand() MATHIAIgebralTrig menu}
tExpand(expression1) \(\Rightarrow\) expression
Returns an expression in which sines and cosines of integer-multiple angles, angle sums, and angle differences are expanded. Because of the identity \((\sin (x))^{2}+(\cos (x))^{2}=1\), there are many possible equivalent results. Consequently, a result might differ from a result shown in other publications.

Sometimes tExpand() will accomplish your goals when the default trigonometric simplification does not. tExpand() tends to reverse transformations done by tCollect(). Sometimes applying tCollect() to a result from tExpand(), or vice versa, in two separate steps simplifies an expression.
Note: Degree-mode scaling by \(\pi / 180\) interferes with the ability of tExpand() to recognize expandable forms. For best results, tExpand() should be used in Radian mode.
tExpand \((\sin (3 \phi))\) ENTER
\[
4 \cdot \sin (\phi) \cdot(\cos (\phi))^{2}-\sin (\phi)
\]
tExpand \((\cos (\alpha-\beta))\) ENTER \(\cos (\alpha) \cdot \cos (\beta)+\sin (\alpha) \cdot \sin (\beta)\)

\section*{Text CATALOG}

Text promptString
Displays the character string promptString dialog box.
If used as part of a Dialog...EndDlog block, promptString is displayed inside that dialog box. If used as a standalone instruction, Text creates a dialog box to display the string.
\[
\begin{aligned}
& \operatorname{tCol1ect}\left((\cos (\alpha))^{\wedge} 2\right) \frac{\cos T E R}{} \\
& \frac{\cos (2 \cdot \alpha)+1}{2}
\end{aligned}
\]
tCollect \((\sin (\alpha) \cos (\beta))\) ENTER
\[
\frac{\sin (\alpha-\beta)+\sin (\alpha+\beta)}{2}
\]

Then See If, page 805.

\section*{timeCnv() CATALOG}

Text "Have a nice day." ENTER
Done


Converts seconds to units of time that can be more easily understood for evaluation. The list is in \{days,hours,minutes,seconds\} format.
Note: See also checkTmr() and startTmr().

Title titleString, [Lb]]
Creates the title of a pull-down menu or dialog box when used inside a Toolbar or Custom construct, or a Dialog...EndDlog block.

Note: Lbl is only valid in the Toolbar construct. When present, it allows the menu choice to branch to a specified label inside the program.

Program segment:
:Dialog
:Title "This is a dialog
box"
: Request "Your name",Str1
:Dropdown "Month you were
born",
seq(string(i),i,1,12),Var1
: EndDlog
\begin{tabular}{|c|c|}
\hline \multicolumn{2}{|l|}{This is a dialdos tox} \\
\hline \multicolumn{2}{|l|}{\multirow[t]{2}{*}{\begin{tabular}{l}
Tour manis: \(\square\) \\
Honth you ware born 1 子
\end{tabular}}} \\
\hline & \\
\hline
\end{tabular}

\section*{tmpCnv() CATALOG}
tmpCnv(expression1_ qempUnit1, _ qempUnit2)
\(\Rightarrow\) expression_ YempUnit2
Converts a temperature value specified by expression1 from one unit to another. Valid temperature units are:


tmpCnv(100_- \({ }^{\circ}\), _ \(\left.{ }^{\circ} \mathrm{f}\right)\) ENTER 212..\(^{\circ}{ }^{\circ} \mathrm{F}\)
tmpCnv ( 32 _ \(^{\circ} \mathrm{f}, \mathrm{O}^{\circ} \mathrm{C}\) ) ENTER \(0 . \cdot{ }^{\circ} \mathrm{C}\)
tmpCnv( \(0{ }_{-}^{\circ}{ }^{\circ}\), \({ }^{\circ}{ }^{\circ} \mathrm{K}\) ) ENTER \(273.15 \cdot{ }^{\circ} \mathrm{K}\)
tmpCnv(0_o \(\left.\mathrm{f},{ }^{\circ} \mathrm{r}\right)\) ENTER 459.67•_ \({ }^{\circ} \mathrm{R}\)

Note: To select temperature units from a menu, press:
ne 2nd [UNITS]

For example, \(100{ }^{\circ} \mathrm{C}\) converts to \(212{ }_{-}{ }^{\circ} \mathrm{F}\) :


To convert a temperature range, use \(\Delta \mathrm{tmpCnv}()\) instead.

\section*{\(\Delta\) tmpCnv() CATALOG}
\(\Delta \mathbf{t m p C n v}\) (expression1_ qempUnit1, _ qempUnit2)
\(\Rightarrow\) expression_ YempUnit2
Converts a temperature range (the difference between two temperature values) specified by expression 1 from one unit to another. Valid temperature units are:

\(1_{-}{ }^{\circ} \mathrm{C}\) and 1 - K have the same magnitude, as do \(1_{-}{ }^{\circ} \mathrm{F}\) and \(1_{-}{ }^{\circ} \mathrm{R}\). However, \(1_{-}{ }^{\circ} \mathrm{C}\) is \(9 / 5\) as large as \(1_{-}{ }^{\circ} \mathrm{F}\).

To get \(\Delta\), you can press \(\rightarrow \square\) [D]
(or 2nd [CHAR] 15). (or 2nd [CHAR] 15 ).
\(\Delta\) tmpCnv ( \(100{ }^{\circ} \mathrm{C},{ }^{\circ} \mathrm{f}\) ) ENTER
\(180 .{ }^{\circ}{ }^{\circ} \mathrm{F}\)
\(\Delta t m p C n v\left(180 \_{ }^{\circ} \mathrm{f},{ }^{\circ} \mathrm{C}\right)\) ENTER
\(100 . \cdot^{\circ} \mathrm{C}\)
\(\Delta\) tmpCnv(100_ \({ }^{\circ} \mathrm{c},{ }^{\circ} \mathrm{k}\) ) ENTER
\(100 . \cdot{ }^{\circ} \mathrm{K}\)
\(\Delta t m p C n v\left(100 \_{ }^{\circ} \mathrm{f},{ }^{\circ} \mathrm{r}\right)\) ENTER
100 •• \({ }^{\circ} \mathrm{R}\)
\(\Delta\) tmpCnv (1_o \({ }^{\circ}\),_ \(\left.{ }^{\circ} \mathrm{f}\right)\) ENTER
\[
1.8^{\cdot}{ }^{\circ} \mathrm{F}
\]

Note: To select temperature units from a menu, press:


For example, a \(100{ }^{\circ}{ }^{\circ} \mathrm{C}\) range (from \(0{ }_{-}^{\circ} \mathrm{C}\) to \(100{ }^{\circ} \mathrm{C}\) ) is equivalent to a \(180^{\circ} \mathrm{F}\) range:


To convert a particular temperature value instead of a range, use tmpCnv().

\section*{Toolbar CATALOG}

\section*{Toolbar}
block

\section*{EndTBar}

Creates a toolbar menu.
block can be either a single statement or a sequence of statements separated with the ":" character. The statements can be either Title or Item.

Items must have labels. A Title must also have a label if it does not have an item.

\section*{Program segment:}
```

:Toolbar

```
: Title "Examples"
: Item "Trig", t
: Item "Calc", c
: Item "Stop", Pexit
:EndTbar
    !

Note: When run in a program, this segment creates a menu with three choices that branch to three places in the program.

\section*{Trace CATALOG}

Trace
Draws a Smart Graph and places the trace cursor on the first defined \(Y=\) function at the previously defined cursor position, or at the reset position if regraphing was necessary.

Allows operation of the cursor and most keys when editing coordinate values. Several keys, such as the function keys, APPS, and MODE, are not activated during trace.

Note: Press ENTER to resume operation.

Try
block1
Else
block2
EndTry
Executes block1 unless an error occurs. Program execution transfers to block2 if an error occurs in block1. Variable errornum contains the error number to allow the program to perform error recovery.
block1 and block2 can be either a single statement or a series of statements separated with the "." character.

Program segment:

:Try
: NewFold(temp)
: Else
: OAlready exists
: Clrerr
: EndTry
\(\vdots\)

Note: See CIrErr and PassErr.

\section*{TwoVar MATH/Statistics menu}

TwoVar list1, list2[, [list3] [, list4, list5]]
Calculates the TwoVar statistics and updates all the system statistics variables.
All the lists must have equal dimensions except for list5.
list1 represents xlist. list2 represents ylist. list3 represents frequency. list4 represents category codes. list5 represents category include list.

Note: list1 through list4 must be a variable name or c1-c99 (columns in the last data variable shown in the Data/Matrix Editor). list5 does not have to be a variable name and cannot be c1-c99.

\section*{Unarchiv CATALOG}

Unarchiv var1 [, var2] [, var3] ...
Moves the specified variables from the user data archive memory to RAM.

You can access an archived variable the same as you would a variable in RAM. However, you cannot delete, rename, or store to an archived variable because it is locked automatically.

To archive variables, use Archive.
\(\{0,1,2,3,4,5,6\} \rightarrow \operatorname{L1}\) ENTER
\(\left\{\begin{array}{llll}0 & 1 & 2 & \text {... }\end{array}\right.\)
\(\{0,2,3,4,3,4,6\} \rightarrow\) L2 ENTER
\(\{023\)... \(\}\)
TwoVar L1,L2 ENTER Done
ShowStat ENTER
\begin{tabular}{|c|c|}
\hline & STAT YARS \\
\hline \(\overline{8}\) & \(=3\). \\
\hline \(\overline{7}\) & =3.14\% \({ }^{\text {a }}\) \\
\hline Ex & =21. \\
\hline Ex & =91. \\
\hline Ey & =22. \\
\hline 5yz & =90. \\
\hline Exy & =咟. \\
\hline Ex & -2.160247 \\
\hline 5 & =1.164454 \\
\hline nstat & \(=7\). \\
\hline mind & =0. \\
\hline min' & \(=0\). \\
\hline maxh & = \({ }^{\text {. }}\) \\
\hline mox' & -6. \\
\hline
\end{tabular}
\(10 \rightarrow\) arctest ENTER 10
Archive arctest ENTER Done
5*arctest ENTER
50
\(15 \rightarrow\) arctest ENTER


ESC
Unarchiv arctest ENTER Done
\(15 \rightarrow\) arctest ENTER
unitV(vector1) \(\Rightarrow\) vector
Returns either a row- or column-unit vector, depending on the form of vector1.
vector1 must be either a single-row matrix or a single-column matrix.
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
\(\left[\begin{array}{lll}\frac{\sqrt{6}}{6} & \frac{\sqrt{6}}{3} & \frac{\sqrt{6}}{6}\end{array}\right]\)
\(\operatorname{unitV}([1 ; 2 ; 3])\) ENTER \(\left[\begin{array}{l}\frac{\sqrt{14}}{14} \\ \frac{\sqrt{14}}{7} \\ \frac{3 \cdot \sqrt{14}}{14}\end{array}\right]\)

\section*{Unlock catalog}

Unlock var1[, var2][, var3]...
Unlocks the specified variables.
Note: The variables can be locked using the Lock command.

\section*{variance() MATH/Statistics menu}
variance (list[, freqlist]) \(\Rightarrow\) expression
Returns the variance of list.
Each freqlist element counts the number of consecutive occurrences of the corresponding element in list.

Note: list must contain at least two elements.
variance(matrix \(1[\), freqmatrix]) \(\Rightarrow\) matrix
Returns a row vector containing the variance of each column in matrix1.

Each freqmatrix element counts the number of consecutive occurrences of the corresponding element in matrix1.

Note: matrix1 must contain at least two rows.

\section*{when() CATALOG}
when(condition, trueResult [, falseResulf]
[, unknownResulf] \(\quad \Rightarrow\) expression
Returns trueResult, falseResult, or unknownResult, depending on whether condition is true, false, or unknown. Returns the input if there are too few arguments to specify the appropriate result.

Omit both falseResult and unknownResult to make an expression defined only in the region where condition is true.
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([1,2,5;-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
\[
[3.91731,2.08411]
\]

Use an undef falseResult to define an expression that graphs only on an interval.

Omit only the unknownResult to define a two-piece expression. than two pieces.

ClrGraph ENTER
Graph when( \(x \geq^{-} \pi\) and
\(x<0, x+3\), undef) ENTER


Graph when \(\left(x<0, x+3,5-x^{\wedge} 2\right)\) ENTER


C1rGraph ENTER
Done
Graph when \((x<0\), when \((x<-\pi\),
4* \(\left.\sin (x), 2 x+3), 5-x^{\wedge} 2\right)\) ENTER

when \((n>0, n *\) factoral \((n-1), 1)\)
\(\rightarrow\) factoral \((\mathrm{n})\) ENTER Done
factoral(3) ENTER 6
3! ENTER 6

\section*{While catalog}

While condition
block
EndWhile
Executes the statements in block as long as condition is true.
block can be either a single statement or a sequence of statements separated with the "." character.

Program segment:
\(\vdots\)
\(: 1 \rightarrow i\)
: \(0 \rightarrow\) temp
:While i<=20
: temp+1/i \(\rightarrow\) temp
: \(\mathrm{i}+1 \rightarrow \mathrm{i}\)
: EndWhile
:Disp "sum of reciprocals up to 20", temp
\(\vdots\)

\section*{"With" See I, page 885.}

Returns true if Boolean expression1 is true and Boolean expression2 is false, or vice versa. Returns false if Boolean expression1 and Boolean expression2 are both true or both false. Returns a simplified Boolean expression if either of the original Boolean expressions cannot be resolved to true or false.

Note: See or
integer1 xor integer2 \(\Rightarrow\) integer
Compares two real integers bit-by-bit using an xor operation. Internally, both integers are converted to signed, 32-bit binary numbers. When corresponding bits are compared, the result is 1 if either bit (but not both) is 1 ; the result is 0 if both bits are 0 or both bits are 1. The returned value represents the bit results, and is displayed according to the Base mode.

You can enter the integers in any number base. For a binary or hexadecimal entry, you must use the 0b or Oh prefix, respectively. Without a prefix, integers are treated as decimal (base 10)

If you enter a decimal integer that is too large for a signed, 32-bit binary form, a symmetric modulo operation is used to bring the value into the appropriate range.

Note: See or.

In Hex base mode:
Oh7AC36 xor Oh3D5F ENTER Oh79169
Important: Zero, not the letter O .
In Bin base mode:
Ob100101 xor Ob100 ENTEROb100001
Note: A binary entry can have up to 32 digits (not counting the Ob prefix). A hexadecimal entry can have up to 8 digits.

\section*{XorPic Catalog}

\section*{XorPic picVar[, row] [, column]}

Displays the picture stored in picVar on the current Graph screen.

Uses xor logic for each pixel. Only those pixel positions that are exclusive to either the screen or the picture are turned on. This instruction turns off pixels that are turned on in both images. picVar must contain a pic data type.
row and column, if included, specify the pixel coordinates for the upper left corner of the picture. Defaults are ( 0,0 ).
zeros(expression, var) \(\Rightarrow\) list
Returns a list of candidate real values of var that make expression \(=0\). zeros() does this by computing exp>list(solve(expression \(=0\), var), var).

For some purposes, the result form for zeros() is more convenient than that of solve(). However, the result form of zeros() cannot express implicit solutions, solutions that require inequalities, or solutions that do not involve var.

Note: See also cSolve(), cZeros(), and solve().
zeros(\{expression1, expression2\}, \{varOrGuess1, varOrGuess \(2[, \ldots]\}) \Rightarrow\) matrix

Returns candidate real zeros of the simultaneous algebraic expressions, where each varOrGuess specifies an unknown whose value you seek.

Optionally, you can specify an initial guess for a variable. Each varOrGuess must have the form:
variable
- or -
variable \(=\) real or non-real number
For example, x is valid and so is \(\mathrm{x}=3\).
If all of the expressions are polynomials and if you
do NOT specify any initial guesses, zeros() uses
If all of the expressions are polynomials and if you
do NOT specify any initial guesses, zeros() uses the lexical Gröbner/Buchberger elimination method to attempt to determine all real zeros.
For example, suppose you have a circle of radius \(r\)
at the origin and another circle of radius \(r\) centered
For example, suppose you have a circle of radius \(r\)
at the origin and another circle of radius \(r\) centered where the first circle crosses the positive \(x\)-axis. Use zeros() to find the intersections.

As illustrated by \(r\) in the example to the right, simultaneous polynomial expressions can have extra variables that have no values, but represent given numeric values that could be substituted later.
Each row of the resulting matrix represents an alternate zero, with the components ordered the same as the varOrGuess list. To extract a row, index the matrix by [row]. -
\(z e r o s\left(a * x^{\wedge} 2+b * x+c, x\right)\) 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* \(x^{\wedge} 2+b * x+c \mid x=a n s(1)[2]\) ENTER
exact(zeros(a* ( \(\left.e^{\wedge}(x)+x\right)\) (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 \(a=0\)

\(z e r o s\left(\left\{x^{\wedge} 2+y^{\wedge} 2-r^{\wedge} 2\right.\right.\),
\(\left.\left.(x-r)^{\wedge} 2+y^{\wedge} 2-r^{\wedge} 2\right\},\{x, y\}\right)\) ENTER
\[
\left[\begin{array}{cc}
\frac{r}{2} & \frac{\sqrt{3} \cdot r}{2} \\
\frac{r}{2} & \frac{-\sqrt{3} \cdot r}{2}
\end{array}\right]
\]

Extract row 2:
ans(1)[2] ENTER
\[
\left[\begin{array}{cc}
\frac{r}{2} & -\frac{\sqrt{3} \cdot r}{2}
\end{array}\right]
\]

You can also (or instead) include unknowns that do not appear in the expressions. For example, you can include \(z\) as an unknown to extend the previous example to two parallel intersecting cylinders of radius \(r\). The cylinder zeros illustrate how families of zeros might contain arbitrary constants in the form @ \(k\), where \(k\) is an integer suffix from 1 through 255. The suffix resets to 1 when you use CIrHome or F1 8:Clear Home.

For polynomial systems, computation time or memory exhaustion may depend strongly on the order in which you list unknowns. If your initial choice exhausts memory or your patience, try rearranging the variables in the expressions and/or varOrGuess list.

If you do not include any guesses and if any expression is non-polynomial in any variable but all expressions are linear in the unknowns, zeros() uses Gaussian elimination to attempt to determine all real zeros.

If a system is neither polynomial in all of its variables nor linear in its unknowns, zeros() determines at most one zero using an approximate iterative method. To do so, the number of unknowns must equal the number of expressions, and all other variables in the expressions must simplify to numbers.

Each unknown starts at its guessed value if there is one; otherwise, it starts at 0.0.

Use guesses to seek additional zeros one by one. For convergence, a guess may have to be rather close to a zero.
zeros \(\left(\left\{x^{\wedge} 2+y^{\wedge} 2-r^{\wedge} 2\right.\right.\),
\(\left.\left.(x-r)^{\wedge} 2+y^{\wedge} 2-r^{\wedge} 2\right\},\{x, y, z\}\right)\)
ENTER
\[
\left[\begin{array}{ccc}
\frac{r}{2} & \frac{\sqrt{3} \cdot r}{2} & @ 1 \\
\frac{r}{2} & \frac{-\sqrt{3} \cdot r}{2} & @ 1
\end{array}\right]
\]
```

zeros({x+e^(z)*y-1,x-y- sin(z)
},{x,y}) ENTER
[\frac{\mp@subsup{e}{}{z}\cdot\operatorname{sin}(z)+1}{\mp@subsup{e}{}{z}+1}}\frac{-(\operatorname{sin}(z)-1)}{\mp@subsup{e}{}{z}+1}
zeros({\mp@subsup{e}{}{\wedge}(z)*y-1,-y-sin(z)},
{y,z}) ENTER
[.041... 3.183...]
zeros({\mp@subsup{e}{}{\wedge}(z)*y-1,-y-sin(z)},
{y,z=2\pi}) ENTER
[.001... 6.281...]

```

\section*{ZoomBox catalog}

\section*{ZoomBox}

Displays the Graph screen, lets you draw a box that defines a new viewing window, and updates the window.

In function graphing mode:


The display after defining ZoomBox by pressing ENTER the second time.

\section*{ZoomData}

Adjusts the window settings based on the currently defined plots (and data) so that all statistical data points will be sampled, and displays the Graph screen.

Note: Does not adjust ymin and ymax for histograms.

In function graphing mode:
\(\left.\begin{array}{l}\begin{array}{l}\{1,2,3,4\} \rightarrow L 1 \text { ENTER } \\ \{2,3,4,5\} \rightarrow L 2 \text { ENTER }\end{array} \quad\left\{\begin{array}{llll}\{1 & 2 & 3 & 4\end{array}\right\} \\ \text { newP1ot } 1,1, \text { L1,L2 ENTER }\end{array} \quad \begin{array}{lll} & 4 & 5\end{array}\right\}\)
ZoomStd ENTER

( HOME
뵤i: [CALC HOME]
ZoomData ENTER


\section*{ZoomDec catalog}

\section*{ZoomDec}

Adjusts the viewing window so that \(\Delta x\) and \(\Delta y=0.1\) and displays the Graph screen with the origin centered on the screen.

In function graphing mode:
1.25x* \(\cos (x) \rightarrow y 1(x)\) ENTER Done ZoomStd ENTER

[ HOME
E [CALC HOME]
ZoomDec ENTER


\section*{ZoomFit}

Displays the Graph screen, and calculates the necessary window dimensions for the dependent variables to view all the picture for the current independent variable settings.

In function graphing mode:
1.25x* \(\cos (x) \rightarrow y 1(x)\) ENTER Done ZoomStd ENTER


HOME
ZoomFit ENTER


\section*{Zoomin CATALOG}

\section*{ZoomIn}

Displays the Graph screen, lets you set a center point for a zoom in, and updates the viewing window.

The magnitude of the zoom is dependent on the Zoom factors xFact and yFact. In 3D Graph mode, the magnitude is dependent on \(x\) Fact, yFact , and zFact.

In function graphing mode:
\(1.25 x * \cos (x) \rightarrow y 1(x)\) ENTER Done ZoomStd:ZoomIn ENTER


ENTER


\section*{Zoomint CATALOG}

\section*{Zoomint}

Displays the Graph screen, lets you set a center point for the zoom, and adjusts the window settings so that each pixel is an integer in all directions.

In function graphing mode:
1.25x* \(\cos (x) \rightarrow y 1(x)\) ENTER Done ZoomStd:ZoomInt ENTER


ENTER


\section*{ZoomOut}

Displays the Graph screen, lets you set a center point for a zoom out, and updates the viewing window.

The magnitude of the zoom is dependent on the Zoom factors xFact and yFact. In 3D Graph mode, the magnitude is dependent on \(\mathrm{xFact}, \mathrm{yFact}\), and zFact.

In function graphing mode:
\(1.25 \mathrm{x} * \cos (\mathrm{x}) \rightarrow \mathrm{y} 1\) ( x ) ENTER
Done
ZoomStd:Zoom0ut ENTER


ENTER


\section*{ZoomPrev CATALOG}

\section*{ZoomPrev}

Displays the Graph screen, and updates the viewing window with the settings in use before the last zoom.

\section*{ZoomRcl CATALOG}

\section*{ZoomRcl}

Displays the Graph screen, and updates the viewing window using the settings stored with the ZoomSto instruction.

\section*{ZoomSqr CATALOG}

\section*{ZoomSqr}

Displays the Graph screen, adjusts the x or y window settings so that each pixel represents an equal width and height in the coordinate system, and updates the viewing window.
In 3D Graph mode, ZoomSqr lengthens the shortest two axes to be the same as the longest axis.

In function graphing mode:
\(1.25 x * \cos (x) \rightarrow y 1(x)\) ENTER Done ZoomStd ENTER


HOME
ZoomSqr ENTER


\section*{ZoomStd}

Sets the window variables to the following standard values, and then updates the viewing window.
Function graphing:
\(x:[-10,10,1], y:[-10,10,1]\) and xres=2
Parametric graphing:
\(\mathrm{t}:[0,2 \pi, \pi / 24], \mathrm{x}:[-10,10,1], \mathrm{y}:[-10,10,1]\)
Polar graphing:
\(\theta:[0,2 \pi, \pi / 24], \mathrm{x}:[-10,10,1], \mathrm{y}:[-10,10,1]\)

In function graphing mode:
\(1.25 x * \cos (x) \rightarrow y 1(x)\) ENTER ZoomStd ENTER


Sequence graphing:
nmin=1, nmax=10, plotStrt=1, plotStep=1,
\(x:[-10,10,1], y:[-10,10,1]\)
3D graphing:
еуе \(\theta^{\circ}=20\), еуе \(\phi^{\circ}=70\), еуе \(\psi^{\circ}=0\)
x: [-10, 10, 14], y: [-10, 10, 14],
z: [-10, 10], ncontour=5
Differential equations graphing:
\(\mathrm{t}:[0,10, .1,0], \mathrm{x}:[-1,10,1], \mathrm{y}:[-10,10,1]\), ncurves=0, Estep=1, diftol=.001, fldres=14, dtime \(=0\)

\section*{ZoomSto CATALOG}

\section*{ZoomSto}

Stores the current Window settings in the Zoom memory. You can use ZoomRcl to restore the settings.

\section*{ZoomTrig CATALOG}

\section*{ZoomTrig}

Displays the Graph screen, sets \(\Delta x\) to \(\pi / 24\), and xscl to \(\pi / 2\), centers the origin, sets the \(y\) settings to \([-4,4, .5]\), and updates the viewing window.

In function graphing mode:
\(1.25 x * \cos (x) \rightarrow y 1(x)\) ENTER Done ZoomStd ENTER

(-) HOME
( \(\square\) [CALC HOME]
ZoomTrig ENTER

(add) \(\dagger\) key expression1 + expression2 \(\Rightarrow\) expression

Returns the sum of expression1 and expression2.
56 ENTER ..... 56
ans (1)+4 ENTER ..... 60
ans (1)+4 ENTER ..... 64
ans (1)+4 ENTER ..... 68
ans (1) 1 + ENTER ..... 72


\begin{tabular}{lll} 
expression + matrix1 & \(\Rightarrow\) matrix & \(20+[1,2 ; 3,4]\) ENTER \\
matrix \(1+\) expression & \(\Rightarrow\) matrix
\end{tabular}\(\quad 2\)

Returns a matrix with expression added to each element on the diagonal of matrix1. matrix1 must be square.

Note: Use .+ (dot plus) to add an expression to each element.
\begin{tabular}{rrrr}
- (subtract) \\
expression1- expression2 \(\Rightarrow\) expression & & \\
Returns expression1 minus expression2. & \(6-2\) ENTER & 4 \\
& \(\pi-\pi / 6\) ENTER & \(\frac{5 \cdot \pi}{6}\) \\
\hline
\end{tabular}
list1-list2 \(\Rightarrow\) list
\(\{22, \pi, \pi / 2\}-\{10,5, \pi / 2\}\) ENTER
matrix1-matrix2 \(\Rightarrow\) matrix
\(\{12 \pi-50\}\)
Subtracts each element in list2 (or matrix2) from the
\([3,4]-[1,2]\) ENTER
\(\left[\begin{array}{ll}2 & 2\end{array}\right]\) corresponding element in list1 (or matrix1), and returns the results.

Dimensions of the arguments must be equal.
\begin{tabular}{|c|c|c|}
\hline expression- list1 \(\Rightarrow\) list & 15- \{10, 15,20\} ENTER & \(\left\{\begin{array}{lll}5 & 0 & -5\end{array}\right\}\) \\
\hline list1- expression \(\Rightarrow\) list & \(\{10,15,20\}-15\) ENTER & \\
\hline Subtracts each list1 element from expression or subtracts expression from each list1 element, and returns a list of the results. & & \\
\hline
\end{tabular}
\(\begin{array}{ll}\text { expression - matrix } 1 & \Rightarrow \text { matrix } \\ \text { matrix } 1-\text { expression } & \Rightarrow \text { matrix }\end{array}\)
expression - matrix 1 returns a matrix of expression times the identity matrix minus matrix1. matrix1 must be square.
matrix1 - expression returns a matrix of expression times the identity matrix subtracted from matrix1. matrix1 must be square.

Note: Use .- (dot minus) to subtract an expression from each element
\(20-[1,2 ; 3,4]\) ENTER
(multiply) 区 key
expression \(1 *\) expression \(2 \Rightarrow\) expression
\(2 * 3.45\) ENTER
6.9

Returns the product of expression1 and expression2.

Returns a list containing the products of the corresponding elements in list1 and list2.
\(\{2 / a, 3 / 2\} *\left\{a^{2}, b / 3\right\}\) ENTER \(\left\{2 \cdot a \frac{b}{2}\right\}\)
Dimensions of the lists must be equal.


Returns a list containing the products of expression and each element in list1.
\begin{tabular}{l} 
expression \(*\) matrix1 \\
matrix1 \(*\) expression
\end{tabular}\(\Rightarrow\) matrix \(\quad[1,2 ; 3,4] * .01\) ENTER \(\quad\left[\begin{array}{cc}.01 & .02 \\
.03 & .04\end{array}\right]\)

Returns a matrix containing the products of expression and each element in matrix1.

Note: Use .* (dot multiply) to multiply an expression \(\lambda *\) identity (3) ENTER by each element.

expression1/ expression2 \(\Rightarrow\) expression
Returns the quotient of expression1 divided by expression2.
```

list1 / list2 = list

```
\[
\{1.0,2,3\} /\{4,5,6\} \text { ENTER }
\]
\(\{.25\) 2/5 1/2\}
Returns a list containing the quotients of list1 divided by list2.
\(2 / 3.45\) ENTER . 57971
\(x^{\wedge} 3 / x\) ENTER \(x^{2}\)

Dimensions of the lists must be equal.
\begin{tabular}{|c|c|c|c|}
\hline \begin{tabular}{l}
expression / list1 \(\Rightarrow\) list \\
list1 / expression \(\Rightarrow\) list \\
Returns a list containing the quotients of expression divided by list1 or list1 divided by expression.
\end{tabular} & \(a /\{3, a, \sqrt{ }(a)\}\) ENTER
\[
\{a, b, c\} /(a * b * c) \text { ENTER }
\] & \multicolumn{2}{|l|}{\[
\left\{\begin{array}{ccc}
\frac{a}{3} & 1 & \sqrt{a}
\end{array}\right\}
\]} \\
\hline & \(\left\{\frac{1}{b \cdot c}\right.\) & & \(\left.\frac{1}{a \cdot b}\right\}\) \\
\hline
\end{tabular}
matrix 1 / expression \(\Rightarrow\) matrix
Returns a matrix containing the quotients of matrix1/ expression.
\([\mathrm{a}, \mathrm{b}, \mathrm{c}] /(\mathrm{a} * \mathrm{~b} * \mathrm{c})\) ENTER
\(\left[\begin{array}{ccc}\frac{1}{b \cdot c} & \frac{1}{a \cdot c} & \frac{1}{a \cdot b}\end{array}\right]\)

Note: Use . I (dot divide) to divide an expression by each element.
\begin{tabular}{|c|c|}
\hline \(\wedge\) (power) \(\wedge_{\text {key }}\) & \\
\hline \begin{tabular}{l}
expression \(1^{\wedge}\) expression \(2 \Rightarrow\) expression \\
list1 ^ list2 \(\Rightarrow\) list \\
Returns the first argument raised to the power of the second argument. \\
For a list, returns the elements in list1 raised to the power of the corresponding elements in list2. \\
In the real domain, fractional powers that have reduced exponents with odd denominators use the real branch versus the principal branch for complex mode.
\end{tabular} & \begin{tabular}{lr}
\(4^{\wedge} 2\) ENTER & 16 \\
\(\{a, 2, c\}^{\wedge}\{1, b, 3\}\) ENTER \(\quad\left\{a \quad 2 b \quad c^{3}\right\}\)
\end{tabular} \\
\hline \begin{tabular}{l}
expression ^ list1 \(\Rightarrow\) list \\
Returns expression raised to the power of the elements in list1.
\end{tabular} & \(p^{\wedge}\{a, 2,-3\}\) ENTER \(\quad\left\{p^{a} \quad p^{2} \frac{1}{p^{3}}\right\}\) \\
\hline \begin{tabular}{l}
list1^ expression \(\Rightarrow\) list \\
Returns the elements in list1 raised to the power of expression.
\end{tabular} & \[
\{1,2,3,4\}^{\wedge-2} \begin{aligned}
& \{\text { ENTER } \\
& \left\{\begin{array}{llll}
1 & 1 / 4 & 1 / 9 & 1 / 16
\end{array}\right\}
\end{aligned}
\] \\
\hline \begin{tabular}{l}
squareMatrix \(1^{\wedge}\) integer \(\Rightarrow\) matrix \\
Returns squareMatrix1 raised to the integer power. squareMatrix1 must be a square matrix. \\
If integer \(=-1\), computes the inverse matrix. If integer < - 1, computes the inverse matrix to an appropriate positive power.
\end{tabular} & \[
\begin{aligned}
& {[1,2 ; 3,4]^{\wedge} 2 \text { ENTER }} \\
& {[1,2 ; 3,4]^{\wedge-1 ~ E N T E R ~}} \\
& {[1,2 ; 3,4]^{\wedge-2 ~ E N T E R ~}}
\end{aligned}
\]
\[
\left[\begin{array}{lc}
{\left[\begin{array}{ll}
1 & 2 \\
3 & 4
\end{array}\right]^{2}} & {\left[\begin{array}{ll}
7 & 107 \\
15 & 2
\end{array}\right]} \\
-\left[\begin{array}{ll}
1 & 2 \\
3 & 4
\end{array}\right]^{-1} & {\left[\begin{array}{ll}
-2 & 1 \\
3 / 2 & -1 / 2
\end{array}\right]} \\
-\left[\begin{array}{ll}
1 & 2 \\
3 & 4
\end{array}\right]^{-2} & {\left[\begin{array}{ll}
11 / 2 & -5 / 2 \\
-15 / 4 & 7 / 4
\end{array}\right]}
\end{array}\right.
\] \\
\hline \multicolumn{2}{|l|}{.+ (dot add) \(\square \pm\) keys} \\
\hline \begin{tabular}{l}
matrix1 .+ matrix2 \(\Rightarrow\) matrix \\
expression .+ matrix1 \(\Rightarrow\) matrix \\
matrix1 .+ matrix2 returns a matrix that is the sum of each pair of corresponding elements in matrix1 and matrix2. \\
expression .+ matrix1 returns a matrix that is the sum of expression and each element in matrix1.
\end{tabular} & [a,2;b,3].+[c,4;5,d][ENTER x. \(+[\mathrm{c}, 4 ; 5, \mathrm{~d}]\) ENTER \\
\hline \multicolumn{2}{|l|}{.- (dot subt.) \(\square \square\) keys} \\
\hline \begin{tabular}{l}
matrix1.- matrix2 \(\Rightarrow\) matrix \\
expression.- matrix1 \(\Rightarrow\) matrix \\
matrix1 .- matrix2 returns a matrix that is the difference between each pair of corresponding elements in matrix1 and matrix2. \\
expression.- matrix1 returns a matrix that is the difference of expression and each element in matrix1.
\end{tabular} & [a,2;b,3].-[c,4;d,5] ENTER x.- \([c, 4 ; d, 5]\) ENTER \\
\hline
\end{tabular}

\begin{tabular}{|c|c|c|}
\hline \(=\) (equal) & \(\square\) key & \\
\hline & expression1 \(=\) expression2 \(\Rightarrow\) Boolean expression list1 \(=\) list2 \(\Rightarrow\) Boolean list matrix1 \(=\) matrix2 \(\Rightarrow\) Boolean matrix & Example function listing using math test symbols: \(=, \neq,<, \leq,>, \geq\) \\
\hline & Returns true if expression 1 is determined to be equal to expression2. & ```
:g(x)
:Func
:If x\leq-5 Then
``` \\
\hline & Returns false if expression 1 is determined to not be equal to expression2. & \begin{tabular}{l}
: Return 5 \\
: ElseIf \(x>-5\) and \(x<0\) Then
\end{tabular} \\
\hline & Anything else returns a simplified form of the equation. & \begin{tabular}{l}
: Return - \(x\) \\
: ElseIf \(x \geq 0\) and \(x \neq 10\) Then
\end{tabular} \\
\hline & For lists and matrices, returns comparisons element by element. & ```
    ElseIf x=10 Then
    Return 3
: EndIf
: EndFunc
``` \\
\hline & & Graph \(\mathrm{g}(\mathrm{x})\) ENTER \\
\hline & &  \\
\hline \# & - \(\because\) key & \\
\hline & \begin{tabular}{l}
expression \(1 \neq\) expression2 \(\Rightarrow\) Boolean expression \\
list1 \(=\) list2 \(\Rightarrow\) Boolean list \\
matrix \(1 \neq\) matrix2 \(\Rightarrow\) Boolean matrix
\end{tabular} & See "=" (equal) example. \\
\hline & Returns true if expression 1 is determined to be not equal to expression2. & \\
\hline & Returns false if expression1 is determined to be equal to expression2. & \\
\hline & Anything else returns a simplified form of the equation. & \\
\hline & For lists and matrices, returns comparisons element by element. & \\
\hline \(<\) & 2nd [<] key & \\
\hline & \[
\begin{aligned}
& \text { expression1 < expression2 } \Rightarrow \text { Boolean expression } \\
& \text { list1< list2 } \Rightarrow \text { Boolean list } \\
& \text { matrix1 < matrix2 } \Rightarrow \text { Boolean matrix }
\end{aligned}
\] & See "=" (equal) example. \\
\hline
\end{tabular}

Returns true if expression 1 is determined to be less than expression2.

Returns false if expression1 is determined to be greater than or equal to expression2.

Anything else returns a simplified form of the equation.

For lists and matrices, returns comparisons element by element.

\& (append) ■ -
string1 \& string2 \(\Rightarrow\) string
"Hello " \& "Nick" ENTER
"Hello Nick"

Returns a text string that is string2 appended to string1.
\(\int\) (expression1, var \([\), lower \(][\),upper \(\left.]\right) \Rightarrow\) expression
\(\int\) (list1,var[,order]) \(\Rightarrow\) list
\(\int\) (matrix 1, var [,order]) \(\Rightarrow\) matrix
Returns the integral of expression1 with respect to the variable varfrom lower to upper.
\(\int\left(x^{\wedge} 2, x, a, b\right)\) ENTER
\(\frac{b^{3}}{3}-\frac{a^{3}}{3}\)
Returns an anti-derivative if lower and upper are omitted. A symbolic constant of integration such as C is omitted

However, lower is added as a constant of integration if only upper is omitted.

Equally valid anti-derivatives might differ by a numeric constant. Such a constant might be disguised-particularly when an anti-derivative contains logarithms or inverse trigonometric functions. Moreover, piecewise constant expressions are sometimes added to make an antiderivative valid over a larger interval than the usual formula.
() returns itself for pieces of expression1 that it cannot determine as an explicit finite combination of its built-in functions and operators.

When lower and upper are both present, an attempt is made to locate any discontinuities or discontinuous derivatives in the interval lower < var < upper and to subdivide the interval at those places.

For the AUTO setting of the Exact/Approx mode, numerical integration is used where applicable when an anti-derivative or a limit cannot be determined.

For the APPROX setting, numerical integration is tried first, if applicable. Anti-derivatives are sought only where such numerical integration is inapplicable or fails.
() can be nested to do multiple integrals. Integration limits can depend on integration variables outside them.

Note: See also nInt().
\(\int\left(b * e^{\wedge}\left(-x^{\wedge} 2\right)+a /\left(x^{\wedge} 2+a^{\wedge} 2\right), x\right)\)
ENTER
\[
\begin{aligned}
& \int\left(b \cdot e^{-x^{2}}+\frac{a}{x^{2}+a^{2}}\right) d x \\
& b \cdot \int\left(e^{-x^{2}}\right) d x+t-n^{-1}\left(\frac{x}{a}\right)
\end{aligned}
\]
\(\int\left(\int(\ln (x+y), y, 0, x), x, 0, a\right)\) ENTER
\[
\begin{aligned}
& \qquad\left\{\begin{array}{l}
a \cdot \begin{array}{l}
x \\
\theta \\
\theta
\end{array} \ln (x+9) d y d x \\
\frac{a^{2} \cdot \ln (a)}{2}+a^{2} \cdot(\ln (2)-3 / 4)
\end{array}\right.
\end{aligned}
\]
\(\sqrt{ }(4)\) ENTER

Returns the square root of the argument.
For a list, returns the square roots of all the elements in list1.
\(\Pi()\) (product) MATH/Calculus menu
\(\Pi\) (expression1, var, low, high) \(\Rightarrow\) expression
Evaluates expression 1 for each value of var from low to high, and returns the product of the results.
\begin{tabular}{lr}
\(\Pi(1 / n, n, 1,5)\) ENTER & \(\frac{1}{120}\) \\
\(\Pi(k \wedge 2, k, 1, n)\) ENTER & \((n!)^{2}\) \\
\(\Pi(\{1 / n, n, 2\}, n, 1,5) \underset{ }{ }\)\begin{tabular}{lll} 
ENTER
\end{tabular} \\
& \(\left\{\frac{1}{120} 120\right.\) \\
& \(32\}\)
\end{tabular}
\(\Pi(k, k, 4,3)\) ENTER 1
\(\Pi(1 / k, k, 4,1)\) ENTER 6
\(\Pi(1 / k, k, 4,1) * \Pi(1 / k, k, 2,4)\) ENTER \(1 / 4\)

\section*{\(\Sigma(\) ) (sum) MATH/Calculus menu}
\(\Sigma\) (expression1, var, low, high) \(\Rightarrow\) expression
Evaluates expression 1 for each value of var from low to high, and returns the sum of the results.
\(\Sigma(1 / n, n, 1,5)\) ENTER \(\frac{137}{60}\)
\(\Sigma\left(k^{\wedge} 2, k, 1, n\right)\) ENTER
\(\frac{n \cdot(n+1) \cdot(2 \cdot n+1)}{6}\)
\(\Sigma\left(1 / n^{\wedge} 2, n, 1, \infty\right)\) ENTER
\(\frac{\pi^{2}}{6}\)
\(\Sigma(\mathrm{k}, \mathrm{k}, 4,3)\) ENTER 0
\(\Sigma(\mathrm{k}, \mathrm{k}, 4,1)\) ENTER -5
\(\Sigma(\mathrm{k}, \mathrm{k}, 4,1)+\Sigma(\mathrm{k}, \mathrm{k}, 2,4)\) ENTER 4

\section*{\# (indirection) CATALOG}
\# varNameString
Refers to the variable whose name is varNameString. This lets you create and modify variables from a program using strings.

Program segment:
:Request "Enter Your Name",str1
:NewFold 非str1
\(\vdots\)
:For i,1,5,1
: ClrGraph
: Graph i*x
: StoPic 非("pic" \& string(i))
: EndFor
\begin{tabular}{|c|c|}
\hline \({ }^{r}\) (radian) MATH/Angle menu & \\
\hline \begin{tabular}{l}
expression \(1^{r} \Rightarrow\) expression \\
list1 \(^{r} \Rightarrow\) list \\
matrix \({ }^{r} \Rightarrow\) matrix \\
In Degree angle mode, multiplies expression 1 by \(180 / \pi\). In Radian angle mode, returns expression1 unchanged. \\
This function gives you a way to use a radian angle while in Degree mode. (In Degree angle mode, \(\boldsymbol{\operatorname { s i n }}(), \boldsymbol{\operatorname { c o s }}(), \boldsymbol{\operatorname { t a n }}()\), and polar-to-rectangular conversions expect the angle argument to be in degrees.) \\
Hint: Use \({ }^{r}\) if you want to force radians in a function or program definition regardless of the mode that prevails when the function or program is used.
\end{tabular} & In Degree or Radian angle mode:
\[
\begin{aligned}
& \cos \left((\pi / 4)^{r}\right) \text { ENTER } \\
& \cos \left(\left\{0^{r},(\pi / 12)^{r},-\pi^{r}\right\}\right) \text { ENTER } \\
& \left\{1 \frac{(\sqrt{3}+1) \cdot \sqrt{2}}{4}-1\right\}
\end{aligned}
\] \\
\hline \({ }^{\circ}\) (degree) 2nd [ \({ }^{\circ}\) ] key & \\
\hline \begin{tabular}{l}
expression \({ }^{\circ} \Rightarrow\) value \\
list \(^{\circ} \Rightarrow\) list \\
matrix \(1^{\circ} \Rightarrow\) matrix \\
In Radian angle mode, multiplies expression by \(\pi / 180\). In Degree angle mode, returns expression unchanged. \\
This function gives you a way to use a degree angle while in Radian mode. (In Radian angle mode, \(\boldsymbol{\operatorname { s i n }}(), \boldsymbol{\operatorname { c o s }}(), \boldsymbol{\operatorname { t a n }}()\), and polar-to-rectangular conversions expect the angle argument to be in radians.)
\end{tabular} & In Radian angle mode: \\
\hline \(\angle\) (angle) 2nd [ \(\angle\) ] key & \\
\hline \begin{tabular}{l}
[radius, \(\angle \theta\) _angle] \(\Rightarrow\) vector (polar input) \\
[radius, \(\angle \theta\) _angle,Z_coordinate] \(\Rightarrow\) vector \\
(cylindrical input) \\
[radius, \(\angle \theta\) _angle, \(\angle \phi\) _angle] \(\Rightarrow\) vector (spherical input) \\
Returns coordinates as a vector depending on the Vector Format mode setting: rectangular, cylindrical, or spherical.
\end{tabular} & \begin{tabular}{l}
[ \(\left.5, \angle 60^{\circ}, \angle 45^{\circ}\right]\) ENTER \\
In Radian mode and vector format set to:
\end{tabular} \\
\hline
\end{tabular}
(magnitude \(\angle\) angle) \(\Rightarrow\) complexValue (polar input)
Enters a complex value in ( \(r \angle \theta\) ) polar form. The angle is interpreted according to the current Angle mode setting.

In Radian angle mode and Rectangular complex format mode:
\(5+3 i-(10 \angle \pi / 4)\) ENTER
\(5-5 \cdot \sqrt{2}+(3-5 \cdot \sqrt{2}) \cdot i\)
-2.071...-4.071... \(i\)


expression I Boolean expression1 [and Boolean expression2]...[and Boolean expressionN]

The "with" (I) symbol serves as a binary operator. The operand to the left of I is an expression. The operand to the right of I specifies one or more relations that are intended to affect the simplification of the expression. Multiple relations after I must be joined by a logical "and"

The "with" operator provides three basic types of functionality: substitutions, interval constraints, and exclusions.

Substitutions are in the form of an equality, such as \(x=3\) or \(y=\sin (x)\). To be most effective, the left side should be a simple variable. expression \(\mid\) variable \(=\) value will substitute value for every occurrence of variable in expression.

Interval constraints take the form of one or more inequalities joined by logical "and" operators. Interval constraints also permit simplification that otherwise might be invalid or not computable.

Exclusions use the "not equals" ( \(/=\) or \(\neq\) ) relational operator to exclude a specific value from consideration. They are used primarily to exclude an exact solution when using cSolve(), cZeros(),
fMax(), fMin(), solve(), zeros(), etc.

\section*{\(\rightarrow\) (store) STOD key}
expression \(\rightarrow\) var
list \(\rightarrow\) var
matrix \(\rightarrow\) var
expression \(\rightarrow\) fun_name(parameter1,...)
list \(\rightarrow\) fun_name(parameter1,...)
matrix \(\rightarrow\) fun_name(parameter1,...)
If variable var does not exist, creates var and initializes it to expression, list, or matrix.

If var already exists and if it is not locked or protected, replaces its contents with expression, list, or matrix.

Hint: If you plan to do symbolic computations using undefined variables, avoid storing anything into commonly used, one-letter variables such as \(a, b, c\), \(x, y, z\), etc.
© [text]
© processes text as a comment line, which can be used to annotate program instructions.
© can be at the beginning or anywhere in the line. Everything to the right of \(\odot\), to the end of the line, is the comment.

\section*{Program segment:}
: © Get 10 points from the Graph
screen
:For i,1,10 © This loops 10
times
!


Ob binaryNumber
Oh hexadecimalNumber

Denotes a binary or hexadecimal number, respectively. To enter a binary or hex number, you must enter the Ob or Oh prefix regardless of the Base mode. Without a prefix, a number is treated as decimal (base 10).

Results are displayed according to the Base mode.

\section*{Appendix B: Technical Reference}

This section contains a comprehensive list of TI-89 Titanium / Voyage \({ }^{\text {TM }} 200\) error messages and character codes. It also includes information about how certain TI-89 Titanium / Voyage \({ }^{\text {TM }} 200\) operations are calculated.
TI-89 Titanium / Voyage \({ }^{\text {TM }} 200\) Error Messages ..... 888
TI-89 Titanium / Voyage \({ }^{\text {™ }} 200\) Modes ..... 896
TI-89 Titanium / Voyage \({ }^{\text {TM }} 200\) Character Codes ..... 902
TI-89 Titanium Key Codes ..... 903
Voyage \({ }^{\text {TM }} 200\) Key Codes ..... 906
Entering Complex Numbers ..... 910
Accuracy Information ..... 913
System Variables and Reserved Names ..... 914
EOS (Equation Operating System) Hierarchy ..... 915
Regression Formulas ..... 917
Contour Levels and Implicit Plot Algorithm ..... 919
Runge-Kutta Method ..... 920
Battery Information ..... 921
In Case of Difficulty ..... 924
Texas Instruments (TI) Support and Service ..... 926

\section*{TI-89 Titanium / Voyage \({ }^{\text {TM }} 200\) Error Messages}

This section lists error messages that may be displayed when input or internal errors are encountered. The number to the left of each error message represents an internal error number that is not displayed. If the error occurs inside a Try...EndTry block, the error number is stored in system variable errornum. Many of the error messages are selfexplanatory and do not require descriptive information. However, additional information has been added for some error messages.

\section*{Error \\ Number Description}

10 A function did not return a value
20 A test did not resolve to TRUE or FALSE
Generally, undefined variables cannot be compared. For example, the test If \(a<b\) will cause this error if either \(a\) or \(b\) is undefined when the If statement is executed.

\section*{Argument cannot be a folder name}

Argument mismatch
Two or more arguments must be of the same type. For example, PtOn expression1,expression2 and PtOn list1,list2 are both valid, but PtOn expression,list is a mismatch.

60 Argument must be a Boolean expression or integer
70 Argument must be a decimal number
80 Argument must be a label name
Argument must be a list
100 Argument must be a matrix
110 Argument must be a Pic
120 Argument must be a Pic or string

Argument must be an empty folder name

\section*{Error \\ Number Description}

160 Argument must be an expression
For example, zeros \((2 x+3=0, x)\) is invalid because the first argument is an equation.
161 ASAP or Exec string too long
163 Attribute (8-digit number) of object (8-digit number) not found
165 Batteries too low for sending or receiving
Install new batteries before sending or receiving.

\section*{Bound}

For the interactive graph math functions like \(2:\) Zero, the lower bound must be less than the upper bound to define the search interval.

Break
The 0 N key was pressed during a long calculation or during program execution.

\section*{Checksum error}

\section*{Circular definition}

This message is displayed to avoid running out of memory during infinite replacement of variable values during simplification. For example, \(a+1 \rightarrow a\), where \(a\) is an undefined variable, will cause this error.

Data is too big to save to a variable. Please use F6 Util to reduce the size.
The size of the data in the editor exceeds the maximum size that can be saved in a variable. The F6 Util menu provides operations that can be used to reduce the size of the data.

Data type
An argument is of the wrong data type.

\section*{Dependent limit}

A limit of integration is dependent on the integration variable. For example, \(\int\left(x^{\wedge} 2, x, 1, x\right)\) is not allowed.

Diff Eq setup

\section*{Dimension}

A list or matrix index is not valid. For example, if the list \(\{1,2,3,4\}\) is stored in L1, then L1[5] is a dimension error because L1 only contains four elements.

\section*{Error \\ Number Description}

\section*{240 Dimension mismatch}

Two or more arguments must be of the same dimension. For example, [1,2]+[1,2,3] is a dimension mismatch because the matrices contain a different number of elements.

Divide by zero
260 Domain error
An argument must be in a specified domain. For example, ans(100) is not valid because the argument for ans() must be in the range 1-99.

Duplicate variable name

Else and Elself invalid outside of If..Endlf block

EndTry is missing the matching Else statement

\section*{Excessive iteration}

Expected 2 or 3-element list or matrix

307 Flash application extension (function or program) not found

First argument of solve or cSolve must be an equation or inequality
For example, solve \(\left(3 x^{\wedge} 2-4, x\right)\) is invalid because the first argument is not an equation.

Folder
An attempt was made in the VAR-LINK menu to store a variable in a folder that does not exist.

Graph functions \(\mathrm{y} 1(\mathrm{x}) . . \mathrm{y} 99(\mathrm{x})\) not available in Diff Equations mode
Inconsistent units

Indirection string is not a valid variable name
Invalid ans()
Invalid assignment
```

Error
Number Description
400 Invalid assignment value
405 Invalid axes
410 Invalid command
420 Invalid folder name
430 Invalid for the current mode settings
440 Invalid implied multiply

```

For example, \(x(x+1)\) is invalid; whereas, \(x^{*}(x+1)\) is the correct syntax. This is to avoid confusion between implied multiplication and function calls.

Invalid in a function or current expression
Only certain commands are valid in a user-defined function. Entries that are made in the Window Editor, Table Editor, Data/Matrix Editor, and Solver as well as system prompts such as Lower Bound cannot contain any commands or a colon (:). See also "Creating and Evaluating User-Defined Functions" in the Calculator Home Screen module.

Invalid in Custom..EndCustm block
470 Invalid in Dialog..EndDlog block
480 Invalid in Toolbar..EndTBar block

490 Invalid in Try..EndTry block
500 Invalid label
Label names must follow the same rules used for naming variables.
510 Invalid list or matrix
For example, a list inside a list such as \(\{2,\{3,4\}\}\) is not valid.
520 Invalid outside Custom..EndCustm or ToolBar..EndTbar blocks
For example, an Item command is attempted outside a Custom or ToolBar structure.

530 Invalid outside Dialog..EndDlog, Custom..EndCustm, or ToolBar..EndTBar blocks For example, a Title command is attempted outside a Dialog, Custom, or ToolBar structure.

540 Invalid outside Dialog..EndDlog block
For example, the DropDown command is attempted outside a Dialog structure.
550 Invalid outside function or program
A number of commands are not valid outside a program or a function. For example, Local cannot be used unless it is in a program or function.

\section*{Error \\ Number Description}

560 Invalid outside Loop..EndLoop, For..EndFor, or While..EndWhile blocks
For example, the Exit command is valid only inside these loop blocks.
570 Invalid pathname
For example, \lvar is invalid.
575 Invalid polar complex

Invalid program reference
Programs cannot be referenced within functions or expressions such as \(1+p(x)\) where \(p\) is a program.

Invalid relocation data in ASM program
The necessary relocation data in the ASM (Assembly) program is missing or corrupted.

590 Invalid syntax block
A Dialog..EndDlog block is empty or has more than one title. A Custom..EndCustm block cannot contain PIC variables, and items must be preceded by a title. A Toolbar..EndTBar block must have a second argument if no items follow; or items must have a second argument and must be preceded by a title.

Invalid table

605 Invalid use of units
610 Invalid variable name in a Local statement

620 Invalid variable or function name

630 Invalid variable reference

640 Invalid vector syntax
650 Link transmission
A transmission between two units was not completed. Verify that the connecting cable is connected firmly to both units.

Matrix not diagonalizable

\section*{Memory}
```

Error
Number Description
690 Missing )
700 Missing "
710 Missing ]
720 Missing }
730 Missing start or end of block syntax
740 Missing Then in the If..Endlf block
750 Name is not a function or program
765 No functions selected
780 No solution found
Using the interactive math features (F5:Math) in the Graph application can give this
error. For example, if you attempt to find an inflection point of the parabola
y1(x)=x2}\mathrm{ , which does not exist, this error will be displayed.
790 Non-algebraic variable in expression
If a is the name of a PIC, GDB, MAC, FIG, etc., a+1 is invalid. Use a different
variable name in the expression or delete the variable.
800 Non-real result
For example, if the unit is in the REAL setting of the Complex Format mode, In(- 2)
is invalid.
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\mathrm{ to
clear editor.
This error message is caused by very low memory conditions inside the Data/Matrix Editor.
830 Overflow
840 Plot setup

```
```

Error
Number Description
850 Program not found
A program reference inside another program could not be found in the provided
path during execution.
Recursion is limited to $\mathbf{2 5 5}$ calls deep
870 Reserved name or system variable
875 ROM-resident routine not available
880 Sequence setup
885 Signature error
890 Singular matrix
895 Slope fields need one selected function and are used for 1st-order equations only
900 Stat
910 Syntax
The structure of the entry is incorrect. For example, $x+-y$ ( $x$ plus minus $y$ ) is invalid; whereas, $x+^{-} y$ ( $x$ plus negative $y$ ) is correct.
930 Too few arguments
The expression or equation is missing one or more arguments. For example, $d(f(x))$ is invalid; whereas, $d(\mathrm{f}(\mathrm{x}), \mathrm{x})$ is the correct syntax.
940 Too many arguments
The expression or equation contains an excessive number of arguments and cannot be evaluated.
950 Too many subscripts
955 Too many undefined variables
960 Undefined variable
965 Unlicensed OS or Flash application
$970 \quad$ Variable in use so references or changes are not allowed
980 Variable is locked, protected, or archived
990 Variable name is limited to 8 characters
Zoom

```

Warning: \(\infty^{\wedge} 0\) or undef^0 replaced by 1
Warning: 0^0 replaced by 1
Warning: \(1^{\wedge} \infty\) or \(1^{\wedge}\) undef replaced by 1
Warning: cSolve may specify more zeros
Warning: May produce false equation
Warning: Expected finite real integrand
Warning: May not be fully simplified
Warning: More solutions may exist
Warning: May introduce false solutions
Warning: Operation may lose solutions
Warning: Requires \& returns 32 bit value
Warning: Overflow replaced by \(\infty\) or \(-\infty\)
Warning: Questionable accuracy
Warning: Questionable solution
Warning: Solve may specify more zeros
Warning: Trig argument too big to reduce

\section*{TI-89 Titanium / Voyage \({ }^{\text {TM }} 200\) Modes}

This section describes the modes of the TI-89 Titanium / Voyage \({ }^{\text {TM }} 200\) and lists the possible settings of each mode. These mode settings are displayed when you press MODE.

\section*{Graph}

\section*{Current Folder}

Note: For detailed information about using folders, see Calculator Home Screen.

Display Digits

Specifies the type of graphs you can plot.
\begin{tabular}{ll}
\hline 1:FUNCTION & \(y(x)\) functions \\
\hline 2:PARAMETRIC & \(x(t)\) and \(y(t)\) parametric equations \\
\hline 3:POLAR & \(r(\theta)\) polar equations \\
\hline 4:SEQUENCE & \(u(n)\) sequences \\
\hline 5:3D & \(z(x, y)\) 3D equations \\
\hline 6:DIFF EQUATIONS & \(y^{\prime}(t)\) differential equations \\
\hline
\end{tabular}

Note: If you use a split screen with Number of Graphs = 2, Graph is for the top or left part of the screen and Graph 2 is for the bottom or right part.

Specifies the current folder. You can set up multiple folders with unique configurations of variables, graph databases, programs, etc.
\begin{tabular}{ll}
\hline 1:main & \begin{tabular}{l} 
Default folder included with the TI-89 \\
\\
Titanium / Voyage \({ }^{\text {TM }} 200\).
\end{tabular} \\
\hline \(2:-\) & \begin{tabular}{l} 
Other folders are available only if they have been \\
(custom folders) \\
created by a user.
\end{tabular} \\
\hline
\end{tabular}

Selects the number of digits. These decimal settings affect only how results are displayed-you can enter a number in any format.
Internally, the TI-89 Titanium / Voyage \({ }^{\text {TM }} 200\) retains decimal numbers with 14 significant digits. For display purposes, such numbers are rounded to a maximum of 12 significant digits.
\begin{tabular}{ll}
\hline 1:FIX 0 & \begin{tabular}{l} 
Results are always displayed with the selected \\
2:FIX 1
\end{tabular} \\
\begin{tabular}{ll}
\(\ldots\) & number of decimal places. \\
D:FIX 12 & \\
\hline E:FLOAT & \begin{tabular}{l} 
The number of decimal places varies, depending on \\
the result.
\end{tabular} \\
\hline F:FLOAT 1 & \begin{tabular}{l} 
If the integer part has more than the selected number \\
of digits, the result is rounded and displayed in
\end{tabular} \\
G:FLOAT 2 & \begin{tabular}{l} 
scientific notation.
\end{tabular} \\
Q:FLOAT 12 & For example, in FLOAT 4: \\
& 12345. is shown as 1.235E4
\end{tabular} \\
\hline
\end{tabular}
\begin{tabular}{|c|c|}
\hline Angle & Specifies the units in which angle values are interpreted and displayed in trig functions and polar/rectangular conversions. \\
\hline & 1:RADIAN \\
\hline & 2:DEGREE \\
\hline Exponential Format & Specifies which notation format should be used. These formats affect only how an answer is displayed; you can enter a number in any format. Numeric answers can be displayed with up to 12 digits and a 3 -digit exponent. \\
\hline & 1:NORMAL Expresses numbers in standard format. For example,
\[
12345.67
\] \\
\hline & \begin{tabular}{l}
2:SCIENTIFIC Expresses numbers in two parts: \\
- The significant digits display with one digit to the left of the decimal. \\
- The power of 10 displays to the right of E . \\
For example, 1.234567 E4 means \(1.234567 \times 10^{4}\)
\end{tabular} \\
\hline & \begin{tabular}{l}
3:ENGINEERING Similar to scientific notation. However: \\
- The number may have one, two, or three digits before the decimal. \\
- The power-of-10 exponent is a multiple of three. \\
For example, 12.34567 E 3 means \(12.34567 \times 10^{3}\)
\end{tabular} \\
\hline & \begin{tabular}{l}
Note: If you select NORMAL, but the answer cannot be displayed in the number of digits selected by Display Digits, the TI-89 \\
Titanium / Voyage \({ }^{\text {TM }} 200\) displays the answer in SCIENTIFIC notation. If Display Digits = FLOAT, scientific notation will be used for exponents of 12 or more and exponents of -4 or less.
\end{tabular} \\
\hline Complex Format & Specifies whether complex results are displayed and, if so, their format. \\
\hline & 1:REAL Does not display complex results. (If a result is a complex number and the input does not contain the complex unit \(i\), an error message is displayed.) \\
\hline & 2:RECTANGULAR Displays complex numbers in the form: \(\mathbf{a + b i}\) \\
\hline & 3:POLAR Displays complex numbers in the form: re \({ }^{\text {i } \theta}\) \\
\hline
\end{tabular}

\section*{Complex Format}

\section*{Vector Format}

\section*{Pretty Print}

Determines how results are displayed on the Home screen.
\begin{tabular}{ll}
\hline 1:OFF & \begin{tabular}{l} 
Results are displayed in a linear, one-dimensional \\
form.
\end{tabular} \\
\hline For example, \(\pi^{\wedge} 2, \pi / 2\), or \(\sqrt{ }((x-3) / x)\) \\
\hline & \begin{tabular}{l} 
Results are displayed in conventional mathematical \\
format.
\end{tabular} \\
For example, \(\pi^{2}, \frac{\pi}{2}\), or \(\sqrt{\frac{x-3}{x}}\) \\
\hline
\end{tabular}

Note: For a complete description of these settings, refer to "Formats of Displayed Results" in the Operating the Calculator module.

Split Screen
Determines how 2-element and 3-element vectors are displayed. You can enter vectors in any of the coordinate systems.
\begin{tabular}{ll}
\hline 1:RECTANGULAR & \begin{tabular}{l} 
Coordinates are in terms of \(x, y\), and \(z\). For example, \\
\\
{\([3,5,2]\) represents \(x=3, y=5\), and \(z=2\).}
\end{tabular} \\
\hline 2:CYLINDRICAL & \begin{tabular}{l} 
Coordinates are in terms of \(r, \theta\), and \(z\). For example, \\
\\
\end{tabular}\([3, \angle 45,2]\) represents \(r=3, \theta=45\), and \(z=2\). \\
\hline 3:SPHERICAL & Coordinates are in terms of \(r, \theta\), and \(\phi\). For example, \\
& {\([3, \angle 45, \angle 90]\) represents \(r=3, \theta=45\), and \(\phi=90\).} \\
\hline
\end{tabular}

Lets you split the screen into two parts. For example, you can display a graph and see the \(\mathrm{Y}=\) Editor at the same time.
\begin{tabular}{ll}
\hline 1:FULL & The screen is not split. \\
\hline 2:TOP-BOTTOM & \begin{tabular}{l} 
The applications are shown in two screens that are \\
above and below each other.
\end{tabular} \\
\hline 3:LEFT-RIGHT & \begin{tabular}{l} 
The applications are shown in two screens that are to \\
the left and right of each other.
\end{tabular} \\
\hline
\end{tabular}

To determine what and how information is displayed on a split screen, use this mode in conjunction with other modes such as Split 1 App, Split 2 App, Number of Graphs, and Split Screen Ratio. (Split Screen Ratio is available on the Voyage \({ }^{\mathrm{TM}} 200\) only.)

Split 1 App
and Split 2 App

Number of Graphs

\section*{Graph 2}

Split Screen Ratio (Voyage \({ }^{\text {TM }} 200\) only)

Specifies which application is displayed on the screen.
- For a full screen, only Split 1 App is active.
- For a split screen, Split 1 App is the top or left part of the screen and Split 2 App is the bottom or right part.

The available application choices are those listed when you press ( © from the Page 2 mode screen or when you press APPS. You must have different applications in each screen unless you are in 2-graph mode.

Specifies whether both parts of a split screen can display graphs at the same time.
\begin{tabular}{ll}
\hline 1 & Only one part can display graphs. \\
\hline 2 & \begin{tabular}{l} 
Both parts can display an independent graph screen \\
\\
\end{tabular} \\
\hline
\end{tabular}

Specifies the type of graphs that you can plot for the second graph on a two-graph split screen. This is active only when Number of Graphs = 2. In this two-graph setting, Graph sets the type of graph for the top or left part of the split screen, and Graph 2 sets the bottom or right part. The available choices are the same as for Graph.

Specifies the proportional sizes of the two parts of a split screen.
\begin{tabular}{ll}
\hline \(1: 1\) & The screen is split evenly. \\
\hline \(1: 2\) & \begin{tabular}{l} 
The bottom or right part is approximately twice the \\
size of the top or left part.
\end{tabular} \\
\hline \(2: 1\) & \begin{tabular}{l} 
The top or left part is approximately twice the size of \\
the bottom or right part.
\end{tabular} \\
\hline
\end{tabular}

Specifies how fractional and symbolic expressions are calculated and displayed. By retaining rational and symbolic forms in the EXACT setting, the TI-89 Titanium / Voyage \({ }^{\mathrm{TM}} 200\) increases precision by eliminating most numeric rounding errors.
\begin{tabular}{ll}
\hline \(1:\) AUTO & \begin{tabular}{l} 
Uses EXACT setting in most cases. However, uses \\
APPROXIMATE if the entry contains a decimal point.
\end{tabular} \\
\hline 2:EXACT & \begin{tabular}{l} 
Displays non-whole-number results in their rational or \\
symbolic form.
\end{tabular} \\
\hline 3:APPROXIMATE & Displays numeric results in floating-point form. \\
\hline
\end{tabular}

Note: For a complete description of these settings, refer to "Formats of Displayed Results" in the Operating the Calculator module.
\begin{tabular}{|c|c|c|}
\hline \multirow[t]{4}{*}{Base} & \multicolumn{2}{|l|}{Lets you perform calculations by entering numbers in decimal, binary, or hexadecimal form.} \\
\hline & 1:DEC & Decimal numbers use 0-9 in the base 10 format \\
\hline & 2:HEX & Hexadecimal numbers use \(0-9\) and \(\mathrm{A}-\mathrm{F}\) in the base 16 format. \\
\hline & 3:BIN & Binary numbers use 0 and 1 in the base 2 format. \\
\hline \multirow[t]{4}{*}{Unit System} & \multicolumn{2}{|l|}{Lets you enter a unit for values in an expression, such as 6_m * 4_m or \(23 \_m / \_\)* \(10 \_s\), convert values from one unit to another within the same category, and create your own user-defined units.} \\
\hline & 1:SI & Select SI for the metric system of measurements \\
\hline & 2:ENG/US & Select ENG/US for the non-metric system of measurements \\
\hline & 3:CUSTOM & Allows you to select custom defaults. \\
\hline Custom Units & \multicolumn{2}{|l|}{Lets you select custom defaults. This mode is dimmed until you select Unit System, 3:CUSTOM.} \\
\hline \multirow[t]{3}{*}{Language} & \multicolumn{2}{|l|}{Lets you localize the TI-89 Titanium / Voyage \({ }^{\text {TM }} 200\) into one of several languages, depending on which language Flash applications are installed.} \\
\hline & 1:English & \begin{tabular}{l}
Default language included with the TI-89 \\
Titanium / Voyage \({ }^{\text {TM }} 200\) operating system (OS).
\end{tabular} \\
\hline & \begin{tabular}{l}
2: - \\
(language Flash applications)
\end{tabular} & Alternate languages are available only if the respective language Flash applications have been installed. \\
\hline
\end{tabular}

\section*{Apps Desktop}

Lets you turn the display of the Apps desktop on or off.
ON Displays the navigable Apps desktop. The Apps desktop appears when you:
- Press APPS.
- Turn the unit on after it has been turned off by pressing 2nd [OFF].
- Press 2nd [QUIT] from an App that is displayed in full screen mode.

OFF
Does not display the navigable Apps desktop.
The unit defaults to the calculator Home screen.
The calculator Home screen displays when you press 2nd [QUIT].

The APPLICATIONS menu displays when you press APPS.

\section*{TI-89 Titanium / Voyage \({ }^{\text {TM }} 200\) Character Codes}

The char() function lets you refer to any character by its numeric character code. For example, to display on the Program I/O screen, use Disp char(127). You can use ord() to find the numeric code of a character. For example, ord ("A") returns 65.
\begin{tabular}{|c|c|}
\hline &  \\
\hline \multicolumn{2}{|l|}{\begin{tabular}{l}
 \\

\end{tabular}} \\
\hline \multicolumn{2}{|l|}{} \\
\hline & \begin{tabular}{l}
 \\

\end{tabular} \\
\hline \multicolumn{2}{|l|}{} \\
\hline \multicolumn{2}{|l|}{\begin{tabular}{l}
N \\

\end{tabular}} \\
\hline & \begin{tabular}{l}
 \\

\end{tabular} \\
\hline
\end{tabular}

\section*{TI－89 Titanium Key Codes}

The getKey（）function returns a value that corresponds to the last key pressed， according to the tables shown in this section．For example，if your program contains a getKey（）function，pressing 2nd［F6］will return a value of 273.

Table 1：Key Codes for Primary Keys
\begin{tabular}{|c|c|c|c|c|c|c|c|c|c|c|}
\hline Key & \multicolumn{10}{|c|}{Modifier} \\
\hline & \multicolumn{2}{|c|}{None} & \multicolumn{2}{|c|}{（} & \multicolumn{2}{|l|}{2nd} & \multicolumn{2}{|c|}{＊} & \multicolumn{2}{|c|}{alpha} \\
\hline & Assoc． & Value & Assoc． & Value & Assoc． & Value & Assoc． & Value & Assoc． & Value \\
\hline F1 & F1 & 268 & F1 & 268 & F6 & 273 & \(Y=\) & 8460 & F1 & 268 \\
\hline F2 & F2 & 269 & F2 & 269 & F7 & 274 & WINDOW & 8461 & F2 & 269 \\
\hline F3］ & F3 & 270 & F3 & 270 & F8 & 275 & GRAPH & 8462 & F3 & 270 \\
\hline ［F4 & F4 & 271 & F4 & 271 & F4 & 271 & TbISet & 8463 & F4 & 271 \\
\hline F5］ & F5 & 272 & F5 & 272 & F5 & 272 & TABLE & 8464 & F5 & 272 \\
\hline \(\square\) & & & COPY & 24576 & CUT & 12288 & & & & \\
\hline alpha & & & & & a－lock & & & & & \\
\hline ESC & ESC & 264 & ESC & 264 & QUIT & 4360 & PASTE & 8456 & ESC & 264 \\
\hline APPS & APPS & 265 & APPS & 265 & SWITCH & 4361 & & 8457 & APPS & 265 \\
\hline HOME & HOME & 277 & HOME & 277 & CUST & 4373 & HOME & 277 & HOME & 277 \\
\hline MODE & MODE & 266 & MODE & 266 & － & 18 & － & 95 & MODE & 266 \\
\hline CATALOG & CATLG & 278 & CATLG & 278 & \(i\) & 151 & \(\infty\) & 190 & CATLG & 278 \\
\hline \(\square\) & BS & 257 & BS & 257 & INS & 4353 & DEL & 8449 & BS & 257 \\
\hline CLEAR & CLEAR & 263 & CLEAR & 263 & CLEAR & 263 & & 8455 & CLEAR & 263 \\
\hline 区 & x & 120 & X & 88 & LN & 4184 & \(\mathrm{e}^{\mathrm{x}}\) & 8280 & x & 120 \\
\hline Y & y & 121 & Y & 89 & SIN & 4185 & \(\mathrm{SIN}^{-1}\) & 8281 & y & 121 \\
\hline Z & z & 122 & Z & 90 & cos & 4186 & \(\mathrm{COS}^{-1}\) & 8282 & z & 122 \\
\hline T & t & 116 & T & 84 & TAN & 4180 & TAN \({ }^{-1}\) & 8276 & t & 116 \\
\hline 囚 & \(\wedge\) & 94 & \(\wedge\) & 94 & \(\pi\) & 140 & \(\theta\) & 136 & \(\wedge\) & 94 \\
\hline \(\square\) & I & 124 & F & 70 & － & 176 & Format d／b & 8316 & f & 102 \\
\hline \(\square\) & \((\) & 40 & B & 66 & \｛ & 123 & & & b & 98 \\
\hline \(\square\) & ） & 41 & C & 67 & \} & 125 & \(\bigcirc\) & 169 & c & 99 \\
\hline \(\square\) & ， & 44 & D & 68 & ［ & 91 & & 8236 & d & 100 \\
\hline － & 1 & 47 & E & 69 & ］ & 93 & ！ & 33 & e & 101 \\
\hline 区 & ＊ & 42 & J & 74 & \(\checkmark\) & 4138 & \＆ & 38 & j & 106 \\
\hline \(\square\) & － & 45 & 0 & 79 & VAR－LNK & 4141 & Contr．－ & & \(\bigcirc\) & 111 \\
\hline ＋ & ＋ & 43 & U & 85 & CHAR & 4139 & Contr．＋ & & u & 117 \\
\hline
\end{tabular}

Table 1: Key Codes for Primary Keys (Continued)
\begin{tabular}{|c|c|c|c|c|c|c|c|c|c|c|}
\hline Key & \multicolumn{10}{|c|}{Modifier} \\
\hline & \multicolumn{2}{|c|}{None} & \multicolumn{2}{|c|}{+} & \multicolumn{2}{|c|}{2nd} & \multicolumn{2}{|c|}{\(\square\)} & \multicolumn{2}{|c|}{alpha} \\
\hline & Assoc. & Value & Assoc. & Value & Assoc. & Value & Assoc. & Value & Assoc. & Value \\
\hline ENTER & CR & 13 & CR & 13 & ENTRY & 4109 & APPROX & 8205 & CR & 13 \\
\hline STO- & STO & 258 & P & 80 & RCL & 4354 & @ & 64 & p & 112 \\
\hline \(\square\) & = & 61 & A & 65 & ' & 39 & \# & 157 & a & 97 \\
\hline EE & EE & 149 & K & 75 & \(\angle\) & 159 & SYMB & 8341 & k & 107 \\
\hline - & - & 173 & SPACE & 32 & ANS & 4372 & & 8365 & SPACE & 32 \\
\hline \(\square\) & . & 46 & W & 87 & \(>\) & 62 & \(\geq\) & 158 & w & 119 \\
\hline 0 & 0 & 48 & V & 86 & < & 60 & \(\leq\) & 156 & v & 118 \\
\hline 1 & 1 & 49 & Q & 81 & " & 34 & & 8241 & q & 113 \\
\hline 2 & 2 & 50 & R & 82 & 1 & 92 & & 8242 & r & 114 \\
\hline 3 & 3 & 51 & S & 83 & UNITS & 4147 & & 8243 & s & 115 \\
\hline 4 & 4 & 52 & L & 76 & : & 58 & & 8244 & 1 & 108 \\
\hline 5 & 5 & 53 & M & 77 & MATH & 4149 & & 8245 & m & 109 \\
\hline 6 & 6 & 54 & N & 78 & MEM & 4150 & & 8246 & n & 110 \\
\hline 7 & 7 & 55 & G & 71 & 1 & 4151 & & 8247 & g & 103 \\
\hline 8 & 8 & 56 & H & 72 & d & 4152 & & 8248 & h & 104 \\
\hline 9 & 9 & 57 & I & 73 & ; & 59 & & 8249 & i & 105 \\
\hline
\end{tabular}

Table 2: Arrow Keys (including diagonal movement)
\begin{tabular}{|c|c|c|c|c|c|}
\hline Key & Normal & ( & 2nd & \(\bullet\) & alpha \\
\hline \(\bigcirc\) & 338 & 16722 & 4434 & 8530 & 33106 \\
\hline (1) & 340 & 16724 & 4436 & 8532 & 33108 \\
\hline \(\odot\) & 344 & 16728 & 4440 & 8536 & 33112 \\
\hline (1) & 337 & 16721 & 4433 & 8529 & 33105 \\
\hline \(\bigcirc\) and (1) & 339 & 16723 & 4435 & 8531 & 33107 \\
\hline \(\bigcirc\) and (1) & 342 & 16726 & 4438 & 8534 & 33110 \\
\hline \(\bigcirc\) and (1) & 345 & 16729 & 4441 & 8537 & 33113 \\
\hline \(\bigcirc\) and (1) & 348 & 16732 & 4444 & 8540 & 33116 \\
\hline
\end{tabular}

Table 3：Greek Letters（prefixed by \(⿴ 囗 十 \square\) ）
\begin{tabular}{|c|c|c|c|c|}
\hline \multirow[t]{2}{*}{Keys} & \multicolumn{4}{|c|}{Second modifier} \\
\hline & \multicolumn{2}{|c|}{alpha} & \multicolumn{2}{|c|}{＋} \\
\hline & Assoc． & Value & Assoc． & Value \\
\hline \(\square \quad\)［A］ & \(\alpha\) & 128 & & \\
\hline \(\square \quad[B]\) & \(\beta\) & 129 & & \\
\hline \(\square \quad[\mathrm{D}]\) & \(\delta\) & 133 & \(\Delta\) & 132 \\
\hline ¢［E］ & \(\varepsilon\) & 134 & & \\
\hline ［1］［F］ & \(\phi\) & 145 & & \\
\hline 7［G］ & \(\gamma\) & 131 & \(\Gamma\) & 130 \\
\hline ［4］［L］ & \(\lambda\) & 137 & & \\
\hline ［5［M］ & \(\mu\) & 181 & & \\
\hline STOص［P］ & \(\pi\) & 140 & \(\Pi\) & 139 \\
\hline ［ \({ }^{\text {［ }}\)［R］ & \(\rho\) & 141 & & \\
\hline ［3］［S］ & \(\sigma\) & 143 & \(\Sigma\) & 142 \\
\hline T］［T］ & \(\tau\) & 144 & & \\
\hline \(\square \quad[\mathrm{W}]\) & \(\omega\) & 148 & \(\Omega\) & 147 \\
\hline 区 & \(\xi\) & 138 & & \\
\hline \(\square\) & \(\psi\) & 146 & & \\
\hline ［ & \(\zeta\) & 135 & & \\
\hline
\end{tabular}

\section*{Voyage \({ }^{\text {TM }} 200\) Key Codes}

The getKey() function returns a value that corresponds to the last key pressed, according to the tables shown in this section. For example, if your program contains a getKey() function, pressing 2nd F1 will return a value of 268.

Table 1: Key Codes for Primary Keys
\begin{tabular}{|c|c|c|c|c|c|c|c|c|}
\hline Key & \multicolumn{8}{|c|}{Modifier} \\
\hline & \multicolumn{2}{|c|}{None} & \multicolumn{2}{|c|}{\(\pm\)} & \multicolumn{2}{|c|}{2nd} & \multicolumn{2}{|c|}{\(\square\)} \\
\hline & Assoc. & Value & Assoc. & Value & Assoc. & Value & Assoc. & Value \\
\hline F1 & F1 & 268 & F1 & 268 & F1 & 268 & & 8460 \\
\hline F2 & F2 & 269 & F2 & 269 & F2 & 269 & & 8461 \\
\hline F3] & F3 & 270 & F3 & 270 & F3 & 270 & & 8462 \\
\hline F4 & F4 & 271 & F4 & 271 & F4 & 271 & & 8463 \\
\hline F5 & F5 & 272 & F5 & 272 & F5 & 272 & & 8464 \\
\hline F6 & F6 & 273 & F6 & 273 & F6 & 273 & & 8465 \\
\hline F7 & F7 & 274 & F7 & 274 & F7 & 274 & & 8466 \\
\hline F8 & F8 & 275 & F8 & 275 & F8 & 275 & & 8467 \\
\hline MODE & MODE & 266 & MODE & 266 & MODE & 266 & & 8458 \\
\hline CLEAR & CLEAR & 263 & CLEAR & 263 & CLEAR & 263 & & 8455 \\
\hline LN & LN & 262 & LN & 262 & \(\mathrm{e}^{\mathrm{x}}\) & 4358 & & 8454 \\
\hline ESC & ESC & 264 & ESC & 264 & QUIT & 4360 & & 8456 \\
\hline APPS & APPS & 265 & APPS & 265 & SWITCH & 4361 & & 8457 \\
\hline ENTER & CR & 13 & CR & 13 & ENTRY & 4109 & APPROX & 8205 \\
\hline SIN & SIN & 259 & SIN & 259 & SIN \({ }^{-1}\) & 4355 & & 8451 \\
\hline COS & COS & 260 & COS & 260 & \(\mathrm{COS}^{-1}\) & 4356 & & 8452 \\
\hline TAN & TAN & 261 & TAN & 261 & TAN \({ }^{-1}\) & 4357 & & 8453 \\
\hline \(\checkmark\) & \(\wedge\) & 94 & \(\wedge\) & 94 & \(\pi\) & 140 & & 8286 \\
\hline \(\square\) & \((\) & 40 & \((\) & 40 & \{ & 123 & & 8232 \\
\hline \(\square\) & ) & 41 & ) & 41 & \} & 125 & & 8233 \\
\hline \(\square\) & , & 44 & , & 44 & [ & 91 & & 8236 \\
\hline \(\div\) & / & 47 & 1 & 47 & ] & 93 & & 8239 \\
\hline 区 & * & 42 & * & 42 & \(\checkmark\) & 4138 & & 8234 \\
\hline \(\square\) & - & 45 & - & 45 & VAR-LNK & 4141 & Contrast - & \\
\hline \(\dagger\) & + & 43 & + & 43 & CHAR & 4139 & Contrast + & \\
\hline STOD & STO & 258 & STO & 258 & RCL & 4354 & & 8450 \\
\hline SPACE & & 32 & & 32 & & 32 & & 8224 \\
\hline \# & = & 61 & = & 61 & 1 & 92 & & 8253 \\
\hline \(\square\) & BS & 257 & BS & 257 & INS & 4353 & DEL & 8449 \\
\hline \(\theta\) & \(\theta\) & 136 & \(\theta\) & 136 & : & 58 & & 8328 \\
\hline (-) & - & 173 & - & 173 & ANS & 4372 & & 8365 \\
\hline \(\square\) & . & 46 & . & 46 & \(>\) & 62 & & 8238 \\
\hline
\end{tabular}

Table 1: Key Codes for Primary Keys (Continued)
\begin{tabular}{|c|c|c|c|c|c|c|c|c|}
\hline \multirow[t]{3}{*}{Key} & \multicolumn{8}{|c|}{Modifier} \\
\hline & \multicolumn{2}{|c|}{None} & \multicolumn{2}{|c|}{1} & \multicolumn{2}{|c|}{2nd} & \multicolumn{2}{|c|}{\(\square\)} \\
\hline & Assoc. & Value & Assoc. & Value & Assoc. & Value & Assoc. & Value \\
\hline 0 & 0 & 48 & 0 & 48 & < & 60 & & 8240 \\
\hline 1 & 1 & 49 & 1 & 49 & E & 149 & & 8241 \\
\hline 2 & 2 & 50 & 2 & 50 & CATALOG & 4146 & & 8242 \\
\hline 3 & 3 & 51 & 3 & 51 & CUST & 4147 & & 8243 \\
\hline 4 & 4 & 52 & 4 & 52 & \(\Sigma\) & 4148 & & 8244 \\
\hline 5 & 5 & 53 & 5 & 53 & MATH & 4149 & & 8245 \\
\hline 6 & 6 & 54 & 6 & 54 & MEM & 4150 & & 8246 \\
\hline 7 & 7 & 55 & 7 & 55 & J & 4151 & & 8247 \\
\hline 8 & 8 & 56 & 8 & 56 & \(d\) & 4152 & & 8248 \\
\hline 9 & 9 & 57 & 9 & 57 & \(\mathrm{x}^{-1}\) & 4153 & & 8249 \\
\hline A & a & 97 & A & 65 & Table 3 & & & 8257 \\
\hline B & b & 98 & B & 66 & - & 39 & & 8258 \\
\hline C & c & 99 & C & 67 & Table 4 & & COPY & 8259 \\
\hline D & d & 100 & D & 68 & - & 176 & & 8260 \\
\hline E & e & 101 & E & 69 & Table 5 & & WINDOW & 8261 \\
\hline F & \(f\) & 102 & F & 70 & \(\angle\) & 159 & FORMAT & 8262 \\
\hline G & g & 103 & G & 71 & Table 6 & & & 8263 \\
\hline H & h & 104 & H & 72 & \& & 38 & & 8264 \\
\hline I & i & 105 & 1 & 73 & i & 151 & & 8265 \\
\hline J & & 106 & J & 74 & \(\infty\) & 190 & & 8266 \\
\hline K & k & 107 & K & 75 & 1 & 124 & KEY & 8267 \\
\hline L & 1 & 108 & L & 76 & " & 34 & & 8268 \\
\hline M & m & 109 & M & 77 & ; & 59 & & 8269 \\
\hline N & n & 110 & N & 78 & Table 7 & & NEW & 8270 \\
\hline O & \(\bigcirc\) & 111 & 0 & 79 & Table 8 & & OPEN & 8271 \\
\hline P & p & 112 & P & 80 & - & 95 & UNITS & 8272 \\
\hline Q & q & 113 & Q & 81 & ? & 63 & CALCHOME & 8273 \\
\hline R & r & 114 & R & 82 & @ & 64 & GRAPH & 8274 \\
\hline S & s & 115 & S & 83 & \(\beta\) & 223 & SAVE & 8275 \\
\hline T & t & 116 & T & 84 & \# & 35 & TBLSET & 8276 \\
\hline U & u & 117 & U & 85 & Table 9 & & & 8277 \\
\hline V & v & 118 & V & 86 & \# & 157 & PASTE & 8278 \\
\hline W & w & 119 & W & 87 & ! & 33 & \(Y=\) & 8279 \\
\hline X & x & 120 & X & 88 & \(\bigcirc\) & 169 & CUT & 8280 \\
\hline Y & y & 121 & Y & 89 & , & 18 & TABLE & 8281 \\
\hline Z & z & 122 & Z & 90 & CAPS & & & 8282 \\
\hline
\end{tabular}

Table 2: Arrow Keys (including diagonal movement)
\begin{tabular}{|c|c|c|c|c|c|}
\hline Key & Normal & 团 & 2nd & - & O \\
\hline \(\bigcirc\) & 338 & 16722 & 4434 & 8530 & 33106 \\
\hline (1) & 340 & 16724 & 4436 & 8532 & 33108 \\
\hline \(\bigcirc\) & 344 & 16728 & 4440 & 8536 & 33112 \\
\hline (1) & 337 & 16721 & 4433 & 8529 & 33105 \\
\hline \(\bigcirc\) and (1) & 339 & 16723 & 4435 & 8531 & 33107 \\
\hline \(\bigcirc\) and (1) & 342 & 16726 & 4438 & 8534 & 33110 \\
\hline \(\bigcirc\) and (1) & 345 & 16729 & 4441 & 8537 & 33113 \\
\hline \(\bigcirc\) and (1) & 348 & 16732 & 4444 & 8540 & 33116 \\
\hline
\end{tabular}

Note: The Grab (园) modifier only affects the arrow keys.
Table 3: Grave Accent Letters (prefixed by 2nd A)
\begin{tabular}{|c|c|c|c|}
\hline Key & Assoc. & Normal & \(\boldsymbol{\uparrow}\) \\
\hline A & à & 224 & 192 \\
\hline E & è & 232 & 200 \\
\hline I & ì & 236 & 204 \\
\hline O & ò & 242 & 210 \\
\hline U & ù & 249 & 217 \\
\hline
\end{tabular}

Table 4: Cedilla Letters (prefixed by 2nd C)
\begin{tabular}{|c|c|c|c|}
\hline Key & Assoc. & Normal & † \\
\hline C & ç & 231 & 199 \\
\hline
\end{tabular}

Table 5: Acute Accent Letters (prefixed by 2 2nd E)
\begin{tabular}{|c|c|c|c|}
\hline Key & Assoc. & Normal & \(\boldsymbol{\dagger}\) \\
\hline A & á & 225 & 193 \\
\hline E & é & 233 & 201 \\
\hline I & í & 237 & 205 \\
\hline O & ó & 243 & 211 \\
\hline U & ú & 250 & 218 \\
\hline Y & ý & 253 & 221 \\
\hline
\end{tabular}

Table 6: Greek Letters (prefixed by 2nd G)
\begin{tabular}{|c|c|c|c|}
\hline Key & Assoc. & Normal & \(\boxed{\dagger}\) \\
\hline A & \(\alpha\) & 128 & \\
\hline B & \(\beta\) & 129 & \\
\hline D & \(\delta\) & 133 & 132 \\
\hline E & \(\varepsilon\) & 134 & \\
\hline F & \(\phi\) & 145 & \\
\hline G & \(\gamma\) & 131 & 130 \\
\hline L & \(\lambda\) & 137 & \\
\hline M & \(\mu\) & 181 & \\
\hline P & \(\pi\) & 140 & 139 \\
\hline R & \(\rho\) & 141 & \\
\hline T & \(\tau\) & 143 & 142 \\
\hline W & \(\omega\) & 144 & \\
\hline X & \(\xi\) & 148 & 147 \\
\hline Y & \(\psi\) & 138 & \\
\hline Z & \(\zeta\) & 146 & \\
\hline
\end{tabular}

Table 7: Tilde Letters (prefixed by [nd N)
\begin{tabular}{|c|c|c|c|}
\hline Key & Assoc. & Normal & \(\boxed{1}\) \\
\hline N & \(\tilde{\mathrm{n}}\) & 241 & 209 \\
\hline O & \(\tilde{\mathrm{o}}\) & 245 & \\
\hline
\end{tabular}

Table 8: Caret Letters (prefixed by 2nd O)
\begin{tabular}{|c|c|c|c|}
\hline Key & Assoc. & Normal & \(\boxed{\uparrow}\) \\
\hline A & â & 226 & 194 \\
\hline E & \(\hat{e}\) & 234 & 202 \\
\hline I & \(\hat{\imath}\) & 238 & 206 \\
\hline O & \(\hat{o}\) & 244 & 212 \\
\hline U & û & 251 & 219 \\
\hline
\end{tabular}

Table 9: Umlaut Letters (prefixed by \(2 n d\) U)
\begin{tabular}{|c|c|c|c|}
\hline Key & Assoc. & Normal & \(\boxed{\dagger}\) \\
\hline A & \(\ddot{a}\) & 228 & 196 \\
\hline E & \(\ddot{\mathrm{e}}\) & 235 & 203 \\
\hline I & \(\ddot{\mathrm{O}}\) & 239 & 207 \\
\hline O & ö & 246 & 214 \\
\hline U & ü & 252 & 220 \\
\hline Y & \(\ddot{\mathrm{y}}\) & 255 & \\
\hline
\end{tabular}

\section*{Entering Complex Numbers}

You can enter complex numbers in the polar form ( \(\mathrm{r} \angle \theta\) ), where \(r\) is the magnitude and \(\theta\) is the angle, or polar form \(r e^{i \theta}\). You can also enter complex numbers in rectangular form a+bi

\section*{Overview of Complex Numbers}

Note: To get the i symbol, press [nd [i], do not simply type an alphabetic i.

A complex number has real and imaginary components that identify a point in the complex plane. These components are measured along the real and imaginary axes, which are similar to the \(x\) and \(y\) axes in the real plane.

The point can be expressed in rectangular form or in either of two polar forms.

The \(\boldsymbol{i}\) symbol represents the imaginary number \(\sqrt{-1}\).


As shown below, the form that you can enter depends on the current Angle mode.
\begin{tabular}{cl}
\hline You can use the form: & When the Angle mode setting is: \\
\hline \(\mathrm{a}+\mathrm{bi}\) & Radian or Degree \\
\(r e^{i \theta}\) & \begin{tabular}{l} 
Radian only \\
\\
\\
\((\mathrm{r} \angle \theta)\)
\end{tabular} \\
& Domain error.) \\
& Radian or Degree \\
\hline
\end{tabular}

Use the following methods to enter a complex number.
\begin{tabular}{ll}
\hline To enter the: & Do this: \\
\hline Rectangular form & Substitute the applicable values or variable \\
\(\mathrm{a}+\mathrm{bi}\) & names for a and b. \\
& \(\mathrm{a} ⿴ \mathrm{t}\) b [nd \([i]\) \\
& For example:
\end{tabular}
\begin{tabular}{|c|c|c|c|}
\hline - \(2+3 \cdot 1\) & & \multicolumn{2}{|r|}{\(2+3 \cdot 1\)} \\
\hline \multicolumn{4}{|l|}{2+3+i} \\
\hline |ratill & Find illta & FUHIL & \(1 / 20\) \\
\hline
\end{tabular}

Important: Do not use the re \({ }^{i \theta}\) polar form in Degree angle mode. It will cause a Domain error.

Note: To get the e symbol, press:
TI-89 Titanium: \(\square[\mathrm{e} x]\). Voyage \({ }^{\text {TM }} 200\) : 2nd \(\left[\mathrm{e}^{x}\right]\) Do not simply type an alphabetic e.

Tip: To get the \(\angle\) symbol, press 2nd [ \(\angle]\).

Tip: To enter \(\theta\) in degrees for ( \(r \angle \theta\) ), you can type a \({ }^{\circ}\) symbol (such as \(45^{\circ}\) ). To get the \({ }^{\circ}\) symbol, press 2nd [ \({ }^{\circ}\) ]. You should not use degrees for \(r e^{i \theta}\).

\section*{Complex Format Mode for Displaying Results}

Note: You can enter complex numbers in any form (or a mixture of all forms) depending on the Angle mode.
\begin{tabular}{|c|c|}
\hline To enter the: & Do this: \\
\hline Polar form \(r e^{i \theta}\)
\(-\mathrm{or}-\)
\((\mathrm{r} \angle \theta)\)
\(\begin{aligned} & \text { Parentheses are } \\ & \text { required for the } \\ & (\mathrm{r} \angle \theta) \text { form. }\end{aligned}\) & \begin{tabular}{l}
Substitute the applicable values or variable names for \(r\) and \(\theta\), where \(\theta\) is interpreted according to the Angle mode setting. \\
TI-89 Titanium: \\
alpha \([\mathrm{R}] \rightarrow\left[\mathrm{e}^{x}\right]\) 2nd \([i] \square[\theta] \square\) - or -
\(\square\) alpha \([R]\) 2nd \([\angle] \square[\theta] \square\) \\
Voyage \(^{\text {TM }} 200\) : \\
R 2nd [ \(\mathrm{e}^{x}\) ] 2nd [i] 日 ( \\
- or - \\
\(\square R\) 2nd [ \(\angle 1\) ] 田
\end{tabular} \\
\hline
\end{tabular}

For example:


Results are shown in rectangular form, but you can select polar form.

Use MODE to set the Complex Format mode to one of three settings.


You can enter a complex number at any time, regardless of the Complex Format mode setting. However, the mode setting determines how results are displayed.
\begin{tabular}{|c|c|}
\hline If Complex Format is: & The TI-89 Titanium / Voyage \({ }^{\text {TM }}\) 200: \\
\hline REAL & \begin{tabular}{l}
Will not display complex results unless you: \\
- Enter a complex number. \\
- or - \\
- Use a complex function such as cFactor(), cSolve(), or cZeros(). \\
If complex results are displayed, they will be shown in either \(\mathrm{a}+\mathrm{bi}\) or \(\mathrm{r}^{i \theta}\) form.
\end{tabular} \\
\hline RECTANGULAR & Displays complex results as a+bi. \\
\hline POLAR & \begin{tabular}{l}
Displays complex results as: \\
- \(r e^{i \theta}\) if the Angle mode = Radian - or - \\
- \((r \angle \theta)\) if the Angle mode \(=\) Degree
\end{tabular} \\
\hline
\end{tabular}

\section*{Using Complex \\ Variables in \\ Symbolic Calculations}

Note: For best results in calculations such as cSolve() and cZeros(), use Method 1.

\section*{Complex Numbers and Degree Mode}

Note: If you use Degree angle mode, you must make polar entries in the form ( \(r \angle \theta\) ). In Degree angle mode, an re \({ }^{i \theta}\) entry causes an error.

Regardless of the Complex Format mode setting, variables that have no stored value and that do not end with an underscore (_) are treated as real numbers. To perform complex symbolic analysis, you can use either of the following methods to set up a complex variable.

Method 1: Use an underscore _
(TI-89 Titanium: [_] Voyage \({ }^{\text {TM }} 200\) 2nd [_]) as the last character in the variable name to designate a complex variable. For example:
\(z_{-}\)is treated as a complex variable if it does not have a stored value.
\begin{tabular}{|c|c|c|}
\hline - imag(z) - & \multicolumn{2}{|r|}{[} \\
\hline - imga(z) & & (z_) \\
\hline imgac \(\mathrm{z}_{-}\)) & & \\
\hline MAl|r kill & FIWIC & E \\
\hline
\end{tabular}

Method 2: Store an unreal value into any variable. For example:
\(\mathrm{x}+\mathrm{y} i \rightarrow \mathrm{z}\)
Then z is treated as a complex variable.


Radian angle mode is recommended for complex number calculations. Internally, the TI-89 Titanium / Voyage \({ }^{\text {TM }} 200\) converts all entered trig values to radians, but it does not convert values for exponential, logarithmic, or hyperbolic functions.

In Degree angle mode, complex identities such as \(e^{\wedge}(\boldsymbol{i} \theta)=\cos (\theta)+\boldsymbol{i} \sin (\theta)\) are not generally true because the values for cos and sin are converted to radians, while those for \(e^{\wedge}()\) are not. For example, \(e^{\wedge}(i 45)=\cos (45)+\) \(\boldsymbol{i} \sin (45)\) is treated internally as \(e^{\wedge}(\boldsymbol{i} 45)=\cos (\pi / 4)+\boldsymbol{i} \sin (\pi / 4)\). Complex identities are always true in Radian angle mode.

\section*{Accuracy Information}

To maximize accuracy, the TI-89 Titanium / Voyage \({ }^{\text {TM }} 200\) carries more digits internally than it displays.

\section*{Computational Accuracy}

Floating-point (decimal) values in memory are stored using up to 14 digits with a 3-digit exponent.
- For min and max Window variables (xmin, xmax, ymin, ymax, etc.), you can store values using up to 12 digits. Other Window variables use 14 digits.
- When a floating-point value is displayed, the displayed value is rounded as specified by the applicable mode settings (Display Digits, Exponential Format, etc.), with a maximum of 12 digits and a 3-digit exponent.
- RegEQ displays up to 14-digit coefficients.

Integer values in memory are stored using up to 614 digits.

\section*{Graphing Accuracy}

Note: For a table that lists the number of pixels in a full screen or split screen, refer to "Setting and Exiting the Split Screen Mode" in Split Screens.

The Window variable xmin is the center of the leftmost pixel used, and xmax is the center of the rightmost pixel used. \(\Delta x\) is the distance between the centers of two horizontally adjacent pixels.
- \(\Delta x\) is calculated as ( \(x\) max \(-x \min\) ) / (\# of \(x\) pixels -1 ).
- If \(\Delta x\) is entered from the Home screen or a program, \(x m a x\) is calculated as \(\mathrm{xmin}+\Delta \mathrm{x} *\) (\# of x pixels -1 ).

The Window variable ymin is the center of the bottom pixel used, and ymax is the center of the top pixel used. \(\Delta \mathrm{y}\) is the distance between the centers of two vertically adjacent pixels.
- \(\Delta \mathrm{y}\) is calculated as (ymax -ymin ) / (\# of y pixels -1 ).
- If \(\Delta y\) is entered from the Home screen or a program, ymax is calculated as ymin \(+\Delta \mathrm{y} *\) (\# of y pixels -1 ).

Cursor coordinates are displayed as eight characters (which may include a negative sign, decimal point, and exponent). The coordinate values (xc, yc, zc, etc.) are updated with a maximum of 12-digit accuracy.

\section*{System Variables and Reserved Names}

This section lists the names of system variables and reserved function names that are used by the TI-89 Titanium / Voyage \({ }^{\text {TM }} 200\). Only those system variables and reserved function names that are identified by an asterisk (*) can be deleted by using Delvar var on the entry line.
\begin{tabular}{|c|c|c|c|c|}
\hline Graph & \(\mathrm{y} 1(\mathrm{x})-\mathrm{y} 99(\mathrm{x})^{*}\) & y1 \({ }^{\prime}(\mathrm{t})-\mathrm{y} 99^{\prime}(\mathrm{t})^{*}\) & yi1-yi99* & \[
\mathrm{r} 1(\theta)-\mathrm{r} 99(\theta)^{*}
\] \\
\hline & xt1(t)-xt99(t)* & yt1 (t)- yt99(t)* & z1(x,y)-z99(x,y)* & u1(n)-u99(n)* \\
\hline & ui1-ui99* & xc & yc & zc \\
\hline & tc & rc & \(\theta c\) & nc \\
\hline & xfact & yfact & zfact & xmin \\
\hline & xmax & xscl & xgrid & ymin \\
\hline & \(y \max\) & yscl & ygrid & xres \\
\hline & \(\Delta \mathrm{x}\) & \(\Delta \mathrm{y}\) & zmin & zmax \\
\hline & zscl & eye \(\theta\) & eyed & eye\% \\
\hline & ncontour & \(\theta\) min & \(\theta\) max & \(\theta\) step \\
\hline & tmin & tmax & tstep & t0 \\
\hline & tplot & ncurves & diftol & dtime \\
\hline & Estep & fldpic & fldres & nmin \\
\hline & nmax & plotStrt & plotStep & sysMath \\
\hline Graph Zoom & zxmin & zxmax & zxscl & zxgrid \\
\hline & zymin & zymax & zyscl & zygrid \\
\hline & zxres & \(z \theta\) min & \(z \theta\) max & z 8 step \\
\hline & ztmin & ztmax & ztstep & zt0de \\
\hline & ztmaxde & ztstepde & ztplotde & zzmin \\
\hline & zzmax & zzscl & zeye \(\theta\) & zeye \(\phi\) \\
\hline & \begin{tabular}{l}
zeyeq \\
zpltstep
\end{tabular} & znmin & znmax & zpltstrt \\
\hline Statistics & \(\overline{\mathrm{x}}\) & \(\overline{\mathrm{y}}\) & \(\Sigma x\) & бX \\
\hline & \(\Sigma x^{2}\) & \(\Sigma \mathrm{xy}\) & \(\Sigma \mathrm{y}\) & \\
\hline & \(\Sigma y^{2}\) & corr & maxX & maxy \\
\hline & medStat & medx 1 & medx2 & medx3 \\
\hline & medy 1 & medy2 & medy 3 & minX \\
\hline & \(\min Y\) & nStat & q1 & q3 \\
\hline & regCoef* & \[
\operatorname{regEq}(x)^{\star}
\] & seed1 & seed2 \\
\hline & Sx & Sy & \(\mathrm{R}^{2}\) & \\
\hline Table & tblStart & \(\Delta \mathrm{tbl}\) & tbllnput & \\
\hline Data/Matrix & c1-c99 & sysData* & & \\
\hline Miscellaneous & main & ok & errornum & \\
\hline Solver & eqn* & exp* & & \\
\hline
\end{tabular}

\section*{EOS (Equation Operating System) Hierarchy}

This section describes the Equation Operating System (EOS \({ }^{\text {TM }}\) ) that is used by the TI-89 Titanium / Voyage \({ }^{\text {TM }} 200\). Numbers, variables, and functions are entered in a simple, straightforward sequence. EOS evaluates expressions and equations using parenthetical grouping and according to the priorities described below.

\section*{Order of Evaluation}
\begin{tabular}{|c|c|}
\hline Level & Operator \\
\hline 1 & Parentheses ( ), brackets [ ], braces \{ \} \\
\hline 2 & Indirection (\#) \\
\hline 3 & Function calls \\
\hline 4 & Post operators: degrees-minutes-seconds ( \({ }^{\circ},{ }^{\prime}\), '"), factorial (!), percentage (\%), radian ( \({ }^{r}\) ), subscript ([ ]), transpose ( \({ }^{\top}\) ) \\
\hline 5 & Exponentiation, power operator ( \(\wedge\) ) \\
\hline 6 & Negation (-) \\
\hline 7 & String concatenation (\&) \\
\hline 8 & Multiplication (*), division (/) \\
\hline 9 & Addition (+), subtraction (-) \\
\hline 10 & Equality relations: equal ( \(=\) ), not equal ( \(\neq\) or \(/=\) ), less than ( \(<\) ), less than or equal ( \(\leq\) or \(<=\) ), greater than ( \(>\) ), greater than or equal ( \(\geq\) or \(>=\) ) \\
\hline 11 & Logical not \\
\hline 12 & Logical and \\
\hline 13 & Logical or, exclusive logical xor \\
\hline 14 & Constraint "with" operator (I) \\
\hline 15 & Store ( \(\rightarrow\) ) \\
\hline
\end{tabular}

\section*{Parentheses, Brackets, and Braces}

All calculations inside a pair of parentheses, brackets, or braces are evaluated first. For example, in the expression 4(1+2), EOS first evaluates the portion of the expression inside the parentheses, \(1+2\), and then multiplies the result, 3 , by 4.

The number of opening and closing parentheses, brackets, and braces must be the same within an expression or equation. If not, an error message is displayed that indicates the missing element. For example, \((1+2) /(3+4\) will display the error message "Missing )."

Note: Because the TI-89 Titanium / Voyage \({ }^{\text {TM }} 200\) allows you to define your own functions, a variable name followed by an expression in parentheses is considered a "function call" instead of implied multiplication. For example \(a(b+c)\) is the function a evaluated by \(b+c\). To multiply the expression \(b+c\) by the variable \(a\), use explicit multiplication: \(a *(b+c)\).

\section*{Indirection}

\section*{Post Operators}

\section*{Exponentiation}

Negation

\section*{Constraint（I）}

The indirection operator（\＃）converts a string to a variable or function name For example，\＃（＂x＂\＆＂y＂\＆＂z＂）creates the variable name xyz．Indirection also allows the creation and modification of variables from inside a program．For example，if \(10 \rightarrow r\) and＂\(r\)＂\(\rightarrow s 1\) ，then \(\# s 1=10\) ．

Post operators are operators that come directly after an argument，such as 5 ！， \(25 \%\) ，or \(60^{\circ} 15^{\prime} 45^{\prime \prime}\) ．Arguments followed by a post operator are evaluated at the fourth priority level．For example，in the expression 4＾3！， 3 ！ is evaluated first．The result， 6 ，then becomes the exponent of 4 to yield 4096.

Exponentiation（ \(\wedge\) ）and element－by－element exponentiation（．\(\wedge\) ）are evaluated from right to left．For example，the expression \(2^{\wedge} 3^{\wedge} 2\) is evaluated the same as \(2^{\wedge}\left(3^{\wedge} 2\right)\) to produce 512 ．This is different from（ \(\left.2^{\wedge} 3\right)^{\wedge} 2\) ，which is 64.

To enter a negative number，press \(⿴ 囗-山\) followed by the number．Post operations and exponentiation are performed before negation．For example， the result of \(-x^{2}\) is a negative number，and \(-9^{2}=-81\) ．Use parentheses to square a negative number such as \((-9)^{2}\) to produce 81 ．Note also that negative \(5(-5)\) is different from minus \(5(-5)\) ，and -3 ！evaluates as \(-(3!)\) ．

The argument following the＂with＂（I）operator provides a set of constraints that affect the evaluation of the argument preceding the＂with＂operator．

\section*{Regression Formulas}

This section describes how the statistical regressions are calculated.

\section*{Least-Squares Algorithm}

Most of the regressions use non-linear recursive least-squares techniques to optimize the following cost function, which is the sum of the squares of the residual errors:
\(J=\sum_{i=1}^{N}[\text { residualExpression }]^{2}\)
where: residualExpression is in terms of \(x_{i}\) and \(y_{i}\)
\(x_{i}\) is the independent variable list
\(y_{i}\) is the dependent variable list
\(N\) is the dimension of the lists
This technique attempts to recursively estimate the constants in the model expression to make \(J\) as small as possible.

For example, \(y=a \sin (b x+c)+d\) is the model equation for SinReg. So its residual expression is:
\(a \sin \left(b x_{i}+c\right)+d-y_{i}\)
For SinReg, therefore, the least-squares algorithm finds the constants \(a, b\), \(c\), and \(d\) that minimize the function:
\(J=\sum_{i=1}^{N}\left[a \sin \left(b x_{i}+c\right)+d-y_{i}\right]^{2}\)

\section*{Regressions}
\begin{tabular}{ll}
\hline Regression & Description \\
\hline CubicReg & \begin{tabular}{l} 
Uses the least-squares algorithm to fit the third-order \\
polynomial: \\
\(y=a x^{3}+b x^{2}+c x+d\)
\end{tabular} \\
& \begin{tabular}{l} 
For four data points, the equation is a polynomial fit; for \\
five or more, it is a polynomial regression. At least four \\
data points are required.
\end{tabular} \\
\hline ExpReg & \begin{tabular}{l} 
Uses the least-squares algorithm and transformed values \\
\(x\) and In \((y)\) to fit the model equation: \\
\(y=a b^{x}\)
\end{tabular} \\
\hline LinReg & \begin{tabular}{l} 
Uses the least-squares algorithm to fit the model equation: \\
\(y=a x+b\) \\
where \(a\) is the slope and \(b\) is the \(y\)-intercept.
\end{tabular} \\
\hline
\end{tabular}
\begin{tabular}{|c|c|}
\hline Regression & Description \\
\hline LnReg & Uses the least-squares algorithm and transformed values \(\ln (x)\) and \(y\) to fit the model equation:
\[
y=a+b \ln (x)
\] \\
\hline Logistic & Uses the least-squares algorithm to fit the model equation:
\[
y=a /\left(1+b * e^{\wedge}(c * x)\right)+d
\] \\
\hline MedMed & \begin{tabular}{l}
Uses the median-median line (resistant line) technique to calculate summary points \(\mathrm{x} 1, \mathrm{y} 1, \mathrm{x} 2, \mathrm{y} 2, \mathrm{x} 3\), and y 3 , and fits the model equation:
\[
y=a x+b
\] \\
where \(a\) is the slope and \(b\) is the \(y\)-intercept.
\end{tabular} \\
\hline PowerReg & Uses the least-squares algorithm and transformed values \(\ln (x)\) and \(\ln (y)\) to fit the model equation:
\[
y=a x^{b}
\] \\
\hline QuadReg & \begin{tabular}{l}
Uses the least-squares algorithm to fit the second-order polynomial:
\[
y=a x^{2}+b x+c
\] \\
For three data points, the equation is a polynomial fit; for four or more, it is a polynomial regression. At least three data points are required.
\end{tabular} \\
\hline QuartReg & \begin{tabular}{l}
Uses the least-squares algorithm to fit the fourth-order polynomial:
\[
y=a x^{4}+b x^{3}+c x^{2}+d x+e
\] \\
For five data points, the equation is a polynomial fit; for six or more, it is a polynomial regression. At least five data points are required.
\end{tabular} \\
\hline SinReg & Uses the least-squares algorithm to fit the model equation:
\[
y=a \sin (b x+c)+d
\] \\
\hline
\end{tabular}

\section*{Contour Levels and Implicit Plot Algorithm}

Contours are calculated and plotted by the following method. An implicit plot is the same as a contour, except that an implicit plot is for the \(z=0\) contour only.

\section*{Algorithm}

Based on your \(x\) and \(y\) Window variables, the distance between \(x m i n\) and \(x m a x\) and between ymin and ymax is divided into a number of grid lines specified by xgrid and ygrid. These grid lines intersect to form a series of rectangles.


For each rectangle, the equation is evaluated at each of the four corners (also called vertices or grid points) and an average value ( E ) is calculated:

\(E=\frac{z_{1}+z_{2}+z_{3}+z_{4}}{4}\)
The \(E\) value is treated as the value of the equation at the center of the rectangle.

For each specified contour value \(\left(\mathrm{C}_{\mathrm{i}}\right)\) :
- At each of the five points shown to the right, the difference between the point's \(z\) value and the contour value is calculated.
- A sign change between any two adjacent poitas implies that a contour crosses the line that joins those two points. Linear interpolation is used to approximate where the zero crosses the line.
- Within the rectangle, any zero crossings are connected with straight lines.
- This process is repeated for each
 contour value.

Each rectangle in the grid is treated similarly.

For Runge-Kutta integrations of ordinary differential equations, the TI-89 Titanium / Voyage \({ }^{\text {TM }} 200\) uses the Bogacki-Shampine 3(2) formula as found in the journal Applied Math Letters, 2 (1989), pp. 1-9.

\section*{Bogacki-Shampine} 3(2) Formula

The Bogacki-Shampine 3(2) formula provides a result of 3rd-order accuracy and an error estimate based on an embedded 2nd-order formula. For a problem of the form:
\(y^{\prime}=f(x, y)\)
and a given step size \(h\), the Bogacki-Shampine formula can be written:
\(F_{1}=f\left(x_{n}, y_{n}\right)\)
\(F_{2}=f\left(x_{\mathrm{n}}+h \frac{1}{2}, y_{\mathrm{n}}+h \frac{1}{2} F_{1}\right)\)
\(F_{3}=f\left(x_{\mathrm{n}}+h \frac{3}{4}, y_{\mathrm{n}}+h \frac{3}{4} F_{2}\right)\)
\(y_{\mathrm{n}+1}=y_{\mathrm{n}}+h\left(\frac{2}{9} F_{1}+\frac{1}{3} F_{2}+\frac{4}{9} F_{3}\right)\)
\(x_{n+1}=x_{n}+h\)
\(F_{4}=f\left(x_{\mathrm{n}+1}, y_{\mathrm{n}+1}\right)\)
errest \(=h\left(\frac{5}{72} F_{1}-\frac{1}{12} F_{2}-\frac{1}{9} F_{3}+\frac{1}{8} F_{4}\right)\)
The error estimate errest is used to control the step size automatically. For a thorough discussion of how this can be done, refer to Numerical Solution of Ordinary Differential Equations by L. F. Shampine (New York: Chapman \& Hall, 1994).

The TI-89 Titanium / Voyage \({ }^{\text {TM }} 200\) software does not adjust the step size to land on particular output points. Rather, it takes the biggest steps that it can (based on the error tolerance diftol) and obtains results for \(x_{n} \leq x \leq x_{n+1}\) using the cubic interpolating polynomial passing through the point \(\left(x_{\mathrm{n}}, y_{\mathrm{n}}\right)\) with slope \(F_{1}\) and through \(\left(x_{n+1}, y_{n+1}\right)\) with slope \(F_{4}\). The interpolant is efficient and provides results throughout the step that are just as accurate as the results at the ends of the step.

\section*{Battery Information}

\section*{When to Replace the Batteries}

Note: To avoid loss of information stored in memory, the TI-89 Titanium / Voyage \({ }^{\text {TM }} 200\) must be off. Do not remove the alkaline batteries and the lithium battery at the same time.

\section*{Effects of Replacing the Batteries}

The TI-89 Titanium / Voyage \({ }^{\text {TM }} 200\) uses two types of batteries: four alkaline batteries, and a lithium battery as a backup for retaining memory while you change the alkaline batteries.

As the alkaline batteries run down, the display will begin to dim (especially during calculations). To compensate for this, you will need to adjust the contrast to a higher setting. If you find it necessary to increase the contrast setting frequently, you will need to replace the alkaline batteries. To assist you, a BATT indicator ( H HT ) will display in the status line area when the batteries have drained down to the point when you should replace them soon. When the BATT indicator is displayed in reverse text (嵑TT), you must replace the alkaline batteries immediately.


To avoid loss of data, do not remove the lithium battery unless four fresh alkaline batteries are installed. Replace the lithium backup battery about every three or four years.

If you do not remove both types of batteries at the same time or allow them to run down completely, you can change either type of battery without losing anything in memory.

Take these precautions when replacing batteries:
- Do not leave batteries within the reach of children.
- Do not mix new and used batteries. Do not mix brands (or types within brands) of batteries.
- Do not mix rechargeable and non-rechargeable batteries.
- Install batteries according to polarity (+ and -) diagrams.
- Do not place non-rechargeable batteries in a battery recharger.
- Properly dispose of used batteries immediately.
- Do not incinerate or dismantle batteries.

\section*{Replacing the} Alkaline Batteries in the TI-89 Titanium

Replacing the
Lithium Battery in the TI-89 Titanium
1. If the TI-89 Titanium is on, turn it off (press 2nd [OFF]) to avoid loss of information stored in memory.
2. Slide the protective cover over the keyboard and place the device face down.
3. Push down on the battery cover latch, and then pull up to remove the cover.
4. Remove all four discharged AAA batteries.
5. Install four new AAA alkaline batteries, arranged according to the polarity (+ and -) diagram inside the battery compartment.
6. Replace the battery cover by inserting the two prongs into the two slots at the bottom of the battery compartment, and then push the cover until the latch snaps closed.

To replace the lithium backup battery, remove the battery cover and unscrew the tiny screw holding the BACK UP BATTERY cover in place.

Remove the old battery and install a new WR44SW or 303 battery, positive (+) side up. Replace the cover and the screw.


\section*{Replacing the} Alkaline Batteries in the Voyage 200
1. If the Voyage \({ }^{\text {TM }} 200\) is on, turn it off (press 2nd [OFF]) to avoid loss of information stored in memory.
2. Slide the protective cover over the keyboard and place the device face down.
3. Press the notched battery cover and slide it off, away from the device.
4. Remove all four discharged AAA batteries.
5. Install four new AAA alkaline batteries, arranged according to the polarity (+ and -) diagram inside the battery compartment.
6. Slide the battery cover onto the device, prong side first. Gently push the cover until the prongs snap into place.

To replace the lithium backup battery, remove the battery cover. Insert a blunt object into the circular indentation next to the battery. Gently place a finger on the lithium battery and pry the battery out.

Slide in a new CR1616 or CR1620 battery, positive (+) side up. Press firmly to snap the new lithium battery into place.


If you have difficulty operating the TI-89 Titanium / Voyage \({ }^{\text {TM }} 200\), the following suggestions may help you correct the problem.
\begin{tabular}{|c|c|}
\hline If: & Suggested action: \\
\hline You cannot see anything on the display. & Press \(\square\) to darken or \(\square\) to lighten the display contrast. \\
\hline The BATT indicator is displayed. & Replace the batteries. If BATT is displayed in reverse text ([WiTI), replace the batteries as soon as possible. \\
\hline The BUSY indicator is displayed. & A calculation is in progress. If you want to stop the calculation, press 0 NN . \\
\hline The PAUSE indicator is displayed. & A graph or program is paused and the TI-89 Titanium / Voyage 200 is waiting for input; press ENTER. \\
\hline An error message is displayed. & Refer to the list of error messages in this module. Press ESC to clear. \\
\hline The TI-89 Titanium / Voyage 200 does not appear to be working properly. & \begin{tabular}{l}
Press ESC several times to exit any menu or dialog box and to return the cursor to the entry line. \\
- or - \\
Be sure that the batteries are installed properly and that they are fresh.
\end{tabular} \\
\hline
\end{tabular}

Note: Correcting a "lock up" will reset your TI-89 Titanium / Voyage 200 and clear its memory.

If:
The TI-89 Titanium appears to be "locked up" and will not respond to keyboard input.

\section*{Suggested action:}

The following action clears RAM. This erases all data, programs, and userdefined variables, functions, or folders.
Press and hold (9), (1), and 2nd. Then press and release 0 N .

The following action clears RAM and Flash ROM. This erases all data, programs, user-defined variables, functions, folders, Flash applications, and the user data archive.
1. Remove one of the four AAA batteries.
2. Press and hold \((-)\) and \(\square\) as you reinstall the battery.
3. Continue holding \((-)\) and \(\square\) for five seconds before releasing.

The Voyage \({ }^{\text {TM }} 200\) appears to be "locked up" and will not respond to keyboard input.

The following action clears RAM. This erases all data, programs, and userdefined variables, functions, or folders. Press and hold 2nd and ©. Then press and release ON .

The following action clears RAM and Flash ROM. This erases all data, programs, user-defined variables, functions, folders, Flash applications, and the user data archive.
1. Remove one of the four AAA batteries.
2. Press and hold \((-)\) and \(\square\) as you reinstall the battery.
3. Continue holding \((-)\) and \(\square\) for five seconds before releasing.

\section*{Texas Instruments Support and Service}

\section*{For general information}

\author{
For technical support
}
\begin{tabular}{ll} 
Home Page: & \(\underline{\text { education.ti.com }}\) \\
KnowledgeBase and & education.ti.com/support \\
e-mail inquiries: & \begin{tabular}{l} 
(800) TI-CARES; (800) 842-2737 \\
Fhone:
\end{tabular} \\
& \begin{tabular}{l} 
For U.S., Canada, Mexico, Puerto Rico, \\
and Virgin Islands only
\end{tabular} \\
International Information: & education.ti.com/international
\end{tabular}

KnowledgeBase and support by e-mail:

Phone (not toll-free):
education.ti.com/support
(972) 917-8324

Customers in the U.S., Canada, Mexico, Puerto Rico and Virgin Islands: Always contact Texas Instruments Customer Support before returning a product for service.

All other customers: Refer to the leaflet enclosed with this product (hardware) or contact your local Texas Instruments retailer/distributor.

\section*{Appendix C: Programmer's Guide}

The parameter/mode strings used in the setMode( ), getMode( ), setGraph( ), and setTable( ) functions do not translate into other languages when used in a program. For example, when you write a program in the French Language mode then switch to the Italian Language mode, the program will produce an error. To avoid this error, you must substitute digits for the alpha characters. These digits operate in all languages. This appendix contains the digit substitutions for these strings.

The following examples show how to substitute digits in the setMode( ) function.

Example 1: A program using alpha parameter/mode strings:
```

setMode("Graph","Sequence")

```

Example 2: The same program, substituting digits for those strings:
```

setMode("1","4")

```
\begin{tabular}{|c|c|}
\hline Parameter/Mode Setting & Strings \\
\hline ALL & 0 \\
\hline Graph & 1 \\
\hline FUNCTION & 1 \\
\hline PARAMETRIC & 2 \\
\hline POLAR & 3 \\
\hline SEQUENCE & 4 \\
\hline 3D & 5 \\
\hline DIFF EQUATIONS & 6 \\
\hline DisplayDigits & 2 \\
\hline FIX 0 & 1 \\
\hline FIX 1 & 2 \\
\hline FIX 2 & 3 \\
\hline FIX 3 & 4 \\
\hline FIX 4 & 5 \\
\hline FIX 5 & 6 \\
\hline FIX 6 & 7 \\
\hline FIX 7 & 8 \\
\hline FIX 8 & 9 \\
\hline FIX 9 & 10 \\
\hline FIX 10 & 11 \\
\hline FIX 11 & 12 \\
\hline FIX 12 & 13 \\
\hline FLOAT & 14 \\
\hline FLOAT 1 & 15 \\
\hline FLOAT 2 & 16 \\
\hline FLOAT 3 & 17 \\
\hline FLOAT 4 & 18 \\
\hline FLOAT 5 & 19 \\
\hline FLOAT 6 & 20 \\
\hline FLOAT 7 & 21 \\
\hline FLOAT 8 & 22 \\
\hline FLOAT 9 & 23 \\
\hline
\end{tabular}
\begin{tabular}{|c|c|}
\hline Parameter/Mode Setting & Strings \\
\hline FLOAT 10 & 24 \\
\hline FLOAT 11 & 25 \\
\hline FLOAT 12 & 26 \\
\hline Angle & 3 \\
\hline RADIAN & 1 \\
\hline degree & 2 \\
\hline Exponential Format & 4 \\
\hline NORMAL & 1 \\
\hline SCIENTIFIC & 2 \\
\hline ENGINEERING & 3 \\
\hline Complex Format & 5 \\
\hline REAL & 1 \\
\hline RECTANGULAR & 2 \\
\hline POLAR & 3 \\
\hline Vector Format & 6 \\
\hline RECTANGULAR & 1 \\
\hline CYLINDRICAL & 2 \\
\hline SPHERICAL & 3 \\
\hline Pretty Print & 7 \\
\hline OFF & 1 \\
\hline ON & 2 \\
\hline SplitScreen & 8 \\
\hline FULL & 1 \\
\hline TOP-BOTTOM & 2 \\
\hline LEFT-RIGHT & 3 \\
\hline \begin{tabular}{l}
Split1App \\
(applications are not numbered)
\end{tabular} & 9 \\
\hline \begin{tabular}{l}
Split2App \\
(applications are not numbered)
\end{tabular} & 10 \\
\hline Number of Graphs & 11 \\
\hline 1 & 1 \\
\hline 2 & 2 \\
\hline
\end{tabular}
\begin{tabular}{lc}
\hline Parameter/Mode Setting & Strings \\
\hline Graph 2 & 12 \\
FUNCTION & 1 \\
PARAMETRIC & 2 \\
POLAR & 3 \\
SEQUENCE & 4 \\
3D & 5 \\
DIFF_EQUATIONS & 6 \\
\hline Split Screen Ratio & 13 \\
1:1 & 1 \\
1:2 & 2 \\
2:1 & 3 \\
\hline Exact/Approx & 14 \\
AUTO & 1 \\
EXACT & 2 \\
APPROXIMATE & 3 \\
\hline Base & 15 \\
DEC & 1 \\
HEX & 2 \\
BIN & 3
\end{tabular}
\begin{tabular}{|c|c|}
\hline Parameter/Mode Setting & Strings \\
\hline Coordinates & 1 \\
\hline RECT & 1 \\
\hline POLAR & 2 \\
\hline OFF & 3 \\
\hline Graph Order & 2 \\
\hline SEQ & 1 \\
\hline SIMUL & 2 \\
\hline Grid & 3 \\
\hline OFF & 1 \\
\hline ON & 2 \\
\hline Axes & 4 \\
\hline \multicolumn{2}{|l|}{In 3D Mode:} \\
\hline OFF & 1 \\
\hline AXES & 2 \\
\hline BOX & 3 \\
\hline \multicolumn{2}{|l|}{Not in 3D Mode:} \\
\hline OFF & 1 \\
\hline ON & 2 \\
\hline Leading Cursor & 5 \\
\hline OFF & 1 \\
\hline ON & 2 \\
\hline Labels & 6 \\
\hline OFF & 1 \\
\hline ON & 1 \\
\hline Seq Axes & 7 \\
\hline time & 1 \\
\hline WEB & 2 \\
\hline Custom & 3 \\
\hline Solution Method & 8 \\
\hline RK & 1 \\
\hline EULER & 2 \\
\hline
\end{tabular}
\begin{tabular}{lc}
\hline Parameter/Mode Setting & Strings \\
\hline Fields & 9 \\
SLPFLD & 1 \\
DIRFLD & 2 \\
FLDOFF & 3 \\
\hline DE Axes & 10 \\
TIME & 1 \\
Y1-VS-Y2 & 2 \\
T-VS-Y' & 3 \\
Y-VS-Y' & 4 \\
Y1-VS-Y2' & 5 \\
Y1'-VS-Y2' & 6 \\
\hline XR Style & 11 \\
WIRE FRAME & 1 \\
HIDDEN SRUFACE & 2 \\
CONTOUR LEVELS & 3 \\
WIRE AND CONTOUR & 4 \\
IMPLICIT PLOT & 5
\end{tabular}
\begin{tabular}{lc}
\hline Parameter/Mode Setting & Strings \\
\hline Graph <->Table & 1 \\
OFF & 1 \\
ON & 2 \\
\hline Independent & 2 \\
AUTO & 1 \\
ASK & 2 \\
\hline Axes & 4
\end{tabular}

\section*{TI－89 Titanium Shortcut Keys}

\section*{General}
\(\square\) APPS
2nd［ \(\boxplus\) ］

> List of Flash applications
> Toggle between last two chosen applications or split screens

2nd［田］
®日，回
\(\rightarrow\) ENTER
\(\bullet \ominus, \oplus \ominus\)
Lighten or darken contrast Calculate approximate answer
Move cursor to top or bottom （in editors）
\(\uparrow \Theta, \uparrow \odot\) Scroll tall objects in history
（1）©（1）Highlight left or right from cursor
2nd \(\Theta\) ，2nd \(\odot\) Page up or page down （in editors）
2nd（1），2nd（1）Move cursor far left or far right

\section*{On－screen Keyboard Map（ \(\boldsymbol{\sim}^{\text {国）}}\) ）}

Press ESC to exit the map．


The keyboard map displays shortcuts that are not marked on the keyboard．As shown below，press \(\square\) and then the applicable key．
\begin{tabular}{ll} 
F & \begin{tabular}{l} 
Access Greek letters \\
（see next column）
\end{tabular} \\
O（comment）
\end{tabular}

\section*{Alpha Rules}
alpha
（
［2nd［a－lock］
Taloha
alpha

\section*{3D Graphing}
\(\odot, \odot,(1),(1)\)
\(\square, \square\)
\(X, Y, Z\)
0
\(\square\)
区

Type one lowercase letter
Type one uppercase letter
Lowercase alpha lock
Uppercase alpha lock
Exit alpha lock

Animate graph
Change animation speed
View along axis
Return to original view
Change graph format style
Expanded／normal view

\section*{Greek Letters}


To access the Greek character set
\(\square \square\) alpha＋letter To access lowercase Greek letters．Example： \(\square \square\) alpha［W］displays \(\omega\)
－团＋letter
To access uppercase Greek letters．Example：
© 0 ［ W ］displays \(\Omega\)
If you press a key combination that does not access a Greek letter，you get the normal letter for that key．


\section*{Voyage \(^{\text {TM }} 200\) Shortcut Keys}

\section*{General}
\begin{tabular}{|c|c|}
\hline －APPS & List of Flash applications（if desktop is off） \\
\hline 2nd［ \(\square\) ］ & Toggle between last two chosen applications or split screens \\
\hline －D & Copy graph coordinates to sysdata \\
\hline －F & Display FORMATS dialog box \\
\hline －H & Copy graph coordinates to Home screen history \\
\hline －N & Create new variable \\
\hline － 0 & Open existing variable \\
\hline － 5 & Save copy as \\
\hline \(\bullet \square, \square\) & Lighten or darken contrast \\
\hline －ENTER & Calculate approximate answer \\
\hline －ON & Turn off unit so that it returns to current application the next time you turn it on \\
\hline \(\square 1-\square 9\) & Run programs kbdprgm1（） through kbdprgm9（） \\
\hline
\end{tabular}

\section*{On－screen Keyboard Map（ \(\quad[\mathrm{KEY}]\) ）}

Press ESC to exit the map．


See the table below for shortcuts that are not marked on the Voyage \({ }^{\text {TM }} 200\) keyboard．See the next column for accent marks and Greek letters．


Editing
\begin{tabular}{|c|c|}
\hline \(\bullet\)－ & Move cursor to top \\
\hline \(\bullet \odot\) & Move cursor to bottom \\
\hline 2nd（1） & Move cursor to far left \\
\hline 2nd（1） & Move cursor to far right \\
\hline ＜\(\odot\) ，图 \(\odot\) & Scroll tall objects in history \\
\hline 2nd \(\Theta\) ，2nd \(\odot\) & Page up and page down \\
\hline \(\square \mathrm{x}\) & Cut \\
\hline －\(\square^{\circ}\) & Copy \\
\hline －V & Paste \\
\hline
\end{tabular}

\section*{3D Graphing}
\(\Theta, \odot, \odot(1) \quad\) Animate graph
田，\(\square\)
\(\mathrm{X}, \mathrm{Y}, \mathrm{Z}\)
0 （zero）
F
区

Change animation speed
View along axis
Return to original view Change graph format style
Expanded／normal view

\section*{Accent Marks}
\begin{tabular}{|c|c|}
\hline 2nd A＋letter & à，è，ì，ò，ù，À，È，ì，Ò，Ù \\
\hline 2nd C＋letter & ç，Ç \\
\hline 2nd \(\mathrm{E}+\) letter & á，é，í，ó，ú，ý，Á，É，Í，Ó，Ú，Y̌ \\
\hline 2nd \(\mathrm{N}+\) letter & ã，ñ，õ， \(\mathrm{A}, \tilde{\mathrm{N}}, \mathrm{O}\) \\
\hline 2nd \(\mathrm{O}+\) letter & â，ê，î，ô，û，Â，Ê，î，Ô，Û \\
\hline 2nd U＋letter & ä，ë，ï，ö，ü，ÿ，̈̈，ë，ï，\(\partial\) ，Ü \\
\hline
\end{tabular}

\section*{Greek Letters}

2nd \(G\)
2nd G＋letter

2nd G \(\quad\)＋letter
To access the Greek character set
To access lowercase Greek letters．Example：2nd G W displays \(\omega\)
To access uppercase Greek letters．Example：2nd G \(\dagger \mathrm{W}\) displays \(\Omega\)
If you press a key combination that does not access a Greek letter，you get the normal letter for that key．


\section*{Keystroke Differences}

There are certain differences in keystrokes using the TI-89 Titanium / Voyage \({ }^{\text {TM }} 200\) for various operations. The following table shows the keystrokes for major commands for the two calculators.
\begin{tabular}{|c|c|c|}
\hline FUNCTION & TI-89 Titanium & Voyage 200 \\
\hline \multicolumn{3}{|l|}{LETTERS} \\
\hline One lowercase letter (a-s, u, v, w) & alpha A-S, U-W & A-S, U-W \\
\hline One lowercase letter ( \(\mathrm{t}, \mathrm{x}, \mathrm{y}, \mathrm{z}\) ) & T, X, Y, Z & T, X, Y, Z \\
\hline Several lowercase letters & 2nd [a-lock] & \\
\hline End several lowercase letters & alpha & \\
\hline Several uppercase letters & (1) [a-lock] & 2nd [CAPS] \\
\hline End several uppercase letters & alpha & 2nd [CAPS] \\
\hline \multicolumn{3}{|l|}{FUNCTION KEYS} \\
\hline F6 & 2nd [F6] & F6 \\
\hline F7 & 2nd [F7] & F7 \\
\hline F8 & 2nd [ F 8 ] & F8 \\
\hline \multicolumn{3}{|l|}{NAVIGATION} \\
\hline Scroll tall objects up or down in history & T \(\odot, \pm \odot\) &  \\
\hline Move cursor far left or far right on entry line & 2nd (1), 2nd (1) & 2nd (1), 2nd (1) \\
\hline Diagonal movement & \begin{tabular}{l}
\(\odot\) and (1) \\
( \()\) and ( \((1)\) \\
\(\odot\) and (1)
and (1)
\end{tabular} & \begin{tabular}{l}
\(\Theta\) and (1) \\
\(\odot\) and (1) \\
\(\odot\) and (1) \\
\(\odot\) and (1)
\end{tabular} \\
\hline \multicolumn{3}{|l|}{FUNCTIONS} \\
\hline Display Home screen & HOME & - [CALC HOME] \\
\hline Cut & - [CUT] & \(\square \mathrm{x}\) \\
\hline Copy & \(\checkmark\) [COPY] & \(\square \mathrm{C}\) \\
\hline Paste & - [PASTE] & \(\square \mathrm{V}\) \\
\hline Catalog & CATALOG & 2nd [CATALOG] \\
\hline Display Units dialog box & 2nd [UNITS] & - [UNITS] \\
\hline Sin & 2nd [SIN] & SIN \\
\hline Cos & 2nd [cos] & COS \\
\hline Tan & 2nd [TAN] & TAN \\
\hline LN & 2nd [LN] & LN \\
\hline \(\mathrm{e}^{\mathrm{x}}\) & \(\checkmark[\mathrm{e} x]\) & 2nd [ \(\mathrm{e}^{x}\) ] \\
\hline EE & EEE & 2nd [EE] \\
\hline
\end{tabular}
\begin{tabular}{|c|c|c|}
\hline FUNCTION & TI-89 Titanium & Voyage 200 \\
\hline \multicolumn{3}{|l|}{SYMBOLS} \\
\hline - (Conversion triangle) & [nd [-] & 2nd [ \(\stackrel{\text { ] }}{ }\) \\
\hline - (Underscore) & - [-] & 2nd [-] \\
\hline \(\theta\) (Theta) & ๑[日] & \(\theta\) \\
\hline 1 ("With") & \(\square\) & 2nd [1] \\
\hline ' (Prime) & 2nd [ \({ }^{\text {] }}\) & 2nd [ \({ }^{\prime}\) ] \\
\hline \({ }^{\circ}\) (Degree) & [2nd [ \({ }^{\circ}\) ] & 2nd [ \({ }^{\circ}\) ] \\
\hline \(\angle\) (Angle) & 2nd [ \(\angle\) ] & 2nd [ \(\angle\) ] \\
\hline \(\Sigma\) (Sigma) & CATALOG \(\Sigma(\) & 2nd [ \(\Sigma\) ] \\
\hline \(\mathrm{x}^{-1}\) (Reciprocal) & CATALOG 1 -1 & 2nd [ \(x\)-1] \\
\hline Space & alphan [-] & Space bar \\
\hline \multicolumn{3}{|l|}{HIDDEN SHORTCUTS} \\
\hline Place data in sysdata variable & OT & - D \\
\hline Greek characters & \(\square \square\) alpha or \(\square^{\square}\) & \(\bullet\) G or \(\bullet\) G \(\dagger\) \\
\hline Keyboard map & - EE & - [KEY] \\
\hline Place data in Home screen history & 眞 & \(\bullet \mathrm{H}\) \\
\hline Grave (à, è, ì, ò, ù) & 2nd [CHAR] 5 & 2nd A a, e, i, o, u \\
\hline Cedilla (ç) & 2nd [CHAR] 56 & 2nd C c \\
\hline Acute (á, é, í, ó, ú, ý) & 2nd [CHAR] 5 & 2nd E a, e, i, o, u, y \\
\hline Tilde (ã, ñ, õ) & 2nd [CHAR] 56 & 2nd N a, n , o \\
\hline Caret ( \(\hat{\text { a }, ~ e ̂, ~ i ̂, ~ o ̂, ~ u ̂) ~}\) & 2nd [CHAR] 5 & 2nd \(\mathrm{O} \mathrm{a}, \mathrm{e}, \mathrm{i}, \mathrm{o}, \mathrm{u}\) \\
\hline Umlaut (ä, ë, ï, ö, ü, y ) & 2nd [CHAR] 5 & 2nd U a, e, i, o, u, y \\
\hline ? (Question mark) & 2nd [CHAR] 3 & 2nd Q \\
\hline \(\beta\) (Beta) & 2nd [CHAR] 56 & 2nd S \\
\hline \# (Indirection) & 2nd [CHAR] 3 & 2nd T \\
\hline \& (Append) & \(\square\) ® (times) & 2nd H \\
\hline @ (Arbitrary) & - STOD & 2nd R \\
\hline \# (Not equal to symbol) & - \(\square^{\square}\) & 2nd V \\
\hline ! (Factorial) & - & 2nd W \\
\hline Comment (Circle-C) & -1) & 2nd \(X\) © \\
\hline New & F1 3 & - N \\
\hline Open & F1 1 & -0 \\
\hline Save copy as & F1 2 & - 5 \\
\hline Format dialog box & - & - F \\
\hline
\end{tabular}

\section*{Index}
Symbols!, factorial69, 879
\#, indirection ..... 586, 881
\(\sqrt{ }()\), square root ..... 881
\%, percent ..... 877
\&, append ..... 586, 880
', minute notation ..... 883
', prime ..... 883
\(\Sigma()\), sum ..... 253, 881
f( ), integrate \(73,227,228,230,234,253\), ..... 254, 880
*, multiply\(\int f(x) d x\) (graph math tool)317, 320
+, add ..... 873
F1-F8 (function keys)
moving among toolbar menus ..... 50
selecting categories ..... 25, 28
selecting menus ..... 44
uses ..... 11
\({ }^{\circ}\), degree notation ..... 743, 882, ..... 883
-, negate ..... 877
-, subtract ..... 874
\(\angle\), angle ..... 882
.*, dot multiplication ..... 877
.+, dot addition ..... 876
\(\rightarrow\), store 581, 885 .^, dot power ..... 877
", second notation ..... 883 ..... 875
\(\neq\), /=, not equal ..... 588, 878
\(<\), less than ..... 588, 878
.-, dot subtraction ..... 876
./, dot division ..... 877
217
<<...>>, insufficient display memory
\(=\), equal ..... 588, 878
>, greater than ..... 588, 879
(-) (negation key) ..... 11
@, arbitrary integer ..... 263
\(\Delta\) list( ), list difference ..... 811
\(\Delta\) tbl, table increment ..... 441
\(\Delta \mathrm{tmpCnv}(\) ), temperature-range conversion ..... 272, 863
\(\Delta x\) window variable ..... 311
\(\Delta y\) window variable ..... 311
^, power ..... 876
_, underscore ..... 883
\(\leq,<=\), less than or equal ..... 588, 879
\(\square\) (subtraction key) ..... 11
\(\geq\), >=, greater than or equal ..... 588, 879
©, comment ..... 566, 886
|, with ..... 73, 74, 226, 235, 885
\(\infty\), infinity ..... 263
\(\square / \square[\mathrm{DEL}]\) (delete character) ..... 13
[0] (hand modifier key) status ..... 34
2nd (second modifier key)
description ..... 10
status ..... 34
[2nd [MEM] (MEMORY) ..... 14
2nd [EE] (exponent key) ..... 12
2nd [ \(\quad\) ] (measurement conversions) ..... 13
2nd [RCL] (recall) ..... 14
2nd [CATALOG] (Catalog) commands ..... 19
description ..... 18
exiting ..... 21
key command ..... 14
2nd [CUSTOM] (Custom)
description ..... 50
example ..... 51
key command ..... 13
2nd [CHAR] (Character) description ..... 45
entering special characters ..... 7
key command ..... 14
selecting characters ..... 7
2nd [QUIT]
calculator Home screen ..... 21
entering commands ..... 19
exiting the split-screen mode ..... 59
key command ..... 14
turning off the calculator ..... 4
, convert ..... 269, 884
Bin, display as binary ..... 665, 767
Cylind, display as cylindrical vector ..... 780
DD, display as decimal angle ..... 782
Dec, display as decimal integer 665, 782
DMS, display as degree/minute/second 788
Hex, display as hexadecimal ..... 665, 804
Polar, display as polar vector ..... 827
Rect, display as rectangular vector ..... 835
Sphere, display as spherical vector ..... 854
1 (shift modifier key)
description ..... 10
status ..... 34
\(\square\) (diamond modifier key) description ..... 10
status ..... 34
- S (SAVE COPY AS) example ..... 49
STOD (store) key ..... 14
\(\uparrow(1) \odot \ominus\) (cursor keys)
entering commands ..... 19
opening Apps ..... 25
selecting entry/answer pairs ..... 23
using the CHAR menu ..... 7
-, negate ..... 144
\(\Pi\) ( ), product ..... 253, 881
\(\theta\) max window variable ..... 329
\(\theta\) min window variable ..... 329
\(\theta\) step window variable ..... 330
r, radian ..... 882
\({ }^{\top}\), transpose ..... 858
\(\mathrm{x}^{-1}\), reciprocal ..... 884
Numerics
009AppB, page \(=574\) ..... 921
0b, binary indicator ..... 886
Oh, hexadecimal indicator ..... 886
10^( ), power of ten ..... 884
3D (three-dimensional) mode ..... 34
3D graphing ..... 361-395
animation ..... 90, 377
CONTOUR LEVELS ..... 93, 381
HIDDEN SURFACE ..... 93, 381
WIRE AND CONTOUR ..... 93, 381
WIRE FRAME ..... 93, 381
A
ABOUT screen ..... 60
abs( ), absolute value ..... 746, 764
absolute value, abs( ) ..... 746
accent marks
CHAR menu ..... 14
accented characters ..... 640
add, + ..... 873
Algebra menu ..... 242, 245
algebra operations ..... 760
All category ..... 28
and (Boolean), and ..... 236, 669, 764
and picture, AndPic ..... 612
and, Boolean and ..... 236, 589, 669
AndPic, and picture ..... 765
Angle mode ..... 175, 291, 897
angle mode ..... 15
status ..... 34
angle( ), angle ..... 765
angle, \(\angle\) ..... 882
ans( ), last answer ..... 766
answer (last), ans( ) ..... 208
APD (Automatic Power Down) ..... 134
APD (Automatic Power Down) feature during calculation or program ..... 5
turning on after ..... 5
append, \& ..... 586, 880
APPLICATIONS menu ..... 169
APPLICATIONS menu ([बAPSS) ..... 45, 53
approx( ), approximate ..... 766
Approximate mode . .154, 176, 191, ..... 228
approximate, approx( ) ..... 243
Apps (calculator software applications) deleting ..... 61
icon highlighted, last open ..... 4
icons ..... 2
names ..... 25
opening ..... 25, 53
shortcuts ..... 30
switching ..... 58
Apps desktop
calculator Home screen and ..... 21
categories ..... 25, 29
clock ..... 37
date and time ..... 39
initial startup ..... 2
mode ..... 15, 35
parts of ..... 4
split-screen status ..... 33
turning off ..... 35
turning off the calculator ..... 4
arbitrary integer, @ . . . . . . . . . . . . . 263
Arc (graph math tool) 317, 323, 332, 339
arc length, arcLen( ) . . . . . . . . . . . . . 253
arccosine, \(\cos ^{-1}(\) ) . . . . . . . . . . . . . . . 773
archive variables, Archive . . . . . 581, 692
Archive, archive variables . 581, 691, 766
archiving variables . . . . . . . . . . . . . . 129
arcLen( ), arc length . . . . . . . . . . . . . 766
arcsine, \(\sin ^{-1}()\). . . . . . . . . . . . . . . . 849
arctangent, \(\tan ^{-1}()\), arctangent . . . . 859
assembly language . . . . . 625, 628, 793
augment( ), augment/concatenate ..726, 766
augment/concatenate, augment( ) . . 726
Auto mode . . . . . . . . 154, 176, 191, 229
AUTO mode status . . . . . . . . . . . . . . . 34
Automatic Power Down (APD) feature during calculation or program . . . . . 5
in OS download mode . . . . . . . . . 66
turning on after . . . . . . . . . . . . . . . . 5
automatic simplification . . . . . . . . . . . 230
automatic tables . . . . . . . . . . . . . . . . 444
auto-paste . . . . . . . . . . . . . 23, 200, 209
avgRC( ), average rate of change . . 767
axes (sequence), CUSTOM . . . . . . . 349
Axes graph format . . 302, 401, 420, 421
Axes settings
372, 380
B
backspace ( \(\boxed{\square}\) ) ..... 13
Base mode ..... 15, 176
BATT message ..... 191
batteries ..... 136, 191
precautions ..... 66
prolonging life ..... 5
replacing ..... 1, 63, 64
binary
display, Bin ..... 665, 767
indicator, Ob ..... 886
rotate, rotate( ) ..... 670
shift, shift( ) ..... 671
BldData, build data ..... 425, 581, 768
Boolean
and, and . . . . . . .236, 589, 669, 764
exclusive or, xor ..... 589, 669, 867
not, not ..... 589, 668, 821
or, or ..... 589, 669
Box Plot ..... 541
build
data, BldData ..... 425, 581, 768
table, Table ..... 858
web, Build Web ..... 349
Build Web, build web ..... 349, 351
build web, Build Web ..... 351
BUSY ..... 35
BUSY indicator ..... 192, 558
Busy/Pause status ..... 35
C
cables ..... 60, 62, 699, 712, 717
Calc menu ..... 252
calculator Home screen
2nd [QUIT] ..... 14
changing entry/answer pairs ..... 24
custom menu ..... 50
entering commands ..... 19
function keys ..... 11
key command ..... 14
toolbar menus ..... 44
turning off the calculator ..... 4
calculator software applications (Apps) 4
icons ..... 2
Calculator-Based Laboratory system connecting ..... 62
Calculator-Based Laboratory. See CBL
Calculator-Based Ranger See CBR
Calculator-Based Ranger systemconnecting62
calculus operations ..... 760
Catalog ([2nd [CATALOG]) commands ..... 19
description ..... 18
exiting ..... 21
key command ..... 14
CATALOG menu ..... 179
categories
All ..... 28
Apps desktop ..... 29
customizing ..... 29
English ..... 28
example of editing ..... 30
Graphing ..... 28
Math ..... 28
Organizr (organizer) ..... 28
Science ..... 28
selecting ..... 28
selecting empty ..... 29
SocialSt (social studies) ..... 28
CBL
get/return, Get ..... 799
programs ..... 617
send list variable, Send ..... 840
statistical data ..... 553, 554
CBL 2 system
activity ..... 740
connecting ..... 62
programs ..... 740
CBR
get/return, Get ..... 799
programs ..... 617
send list variable, Send ..... 840
statistical data ..... 553, 554
CBR system
connecting ..... 62
programs ..... 740
ceiling(), ceiling ..... 768
ceiling, ceiling( ) ..... 729
certificate . \(705,711,712,713,714,715\), ..... 716
Certificate revision (Cert. Rev.) ..... 222
cFactor( ), complex factor .244, 752, ..... 768
CHAR menu ([2nd [CHAR])45
entering special characters ..... 7
key command ..... 14
char( ), character string ..... 769
character string, char( ) ..... 586
characters
deleting ..... 13
Greek ..... \(7,14,45,641,642\)
international/accented ..... 7, 14, 45
math ..... 7, 14, 45
numeric code, ord( ) ..... 587
punctuation ..... 45
special \(7,14,45,639,640,642\)
string, char( ) ..... 586
symbols ..... 642
uppercase ..... 10
uppercase/lowercase ..... 142, 634
checkTmr( ), check timer ..... 769
circle
drawing ..... 478
graphing ..... 77, 80
circle, Circle ..... 615
Circle, draw circle ..... 615, 769
Circular definition error ..... 580
Clean Up menu ..... 177
clear
drawing, ClrDraw ..... 614
error, ClrErr ..... 620
graph, ClrGraph ..... 460, 611, 661
I/O, ClrIO ..... 560, 604
clipboard ..... 200, 637
Clock
dialog box ..... 37
operation ..... 36
turning off ..... 42
turning on ..... 44
ClockOff, turning clock off ..... 769
ClockOn, turning clock on ..... 770
ClrDraw, clear drawing ..... 614, 770
CIrErr, clear error ..... 620, 770
CIrGraph, clear graph ..... 611, 661, 770
ClrHome, clear home ..... 25, 771
ClrIO, clear I/O ..... 771
ClrIO, clear I/O ..... 560, 604, 771
cobweb plot. See web plots
colDim( ), matrix column dimension ..... 771
colNorm( ), matrix column norm ..... 771
combinations, \(\mathrm{nCr}(\) ) ..... 818
comDenom( ), common denominator ..... 771
command mark ..... 645
command scripts ..... 198, 645, 648
activity ..... 732
commands
Flash Apps ..... 18
Key ..... 7, 9
comment, © ..... 566, 886
common denominator, comDenom( ) 243,244, 251
complex
factor, cFactor( ) ..... 244, 752, 768
mode, Complex Format ..... 176, 897
modulus surface ..... 388
numbers ..... 68, 69
solve, cSolve( ) ..... 227, 776
tables ..... 448
zeros, cZeros( ) ..... 227, 244, 780
Complex Format mode ..... 176, 897
complex format mode ..... 15
Complex menu ..... 244
conj( ), complex conjugate ..... 772
connecting
TI ViewScreen overhead panel ..... 62
TI-Presenter video adapter ..... 62
Constant Memory ..... 134
Constant Memory feature ..... 14
constants ..... 265
predefined ..... 279
contour plots ..... 382, 386, 387
DrwCtour, draw contour ..... 386
contour-level graphing ..... 93, 381
contrast
adjusting ..... 2, 64
initial startup ..... 2
contrast, adjusting ..... 135
convert measurements ..... 13
convert time, timeCnv( ) ..... 861
convert, ..... 269, 884
Coordinates graph format ..... 301, 330
copy ..... 200, 637
copy variable, CopyVar ..... 581, 687
CopyVar, copy variable ..... 687, 772
cos( ), cosine ..... 773
\(\cos ^{-1}()\), arccosine ..... 773
cosh( ), hyperbolic cosine ..... 774
\(\cosh ^{-1}()\), hyperbolic arccosine ..... 774
cot( ), cotangent ..... 774
\(\cot ^{-1}()\), inverse cotangent ..... 775
coth( ), hyperbolic cotangent ..... 775
coth \(^{-1}()\), inverse hyperbolic cotangent 775
crossP( ), cross product ..... 775
csc( ), cosecant ..... 775
\(\csc ^{-1}()\), inverse cosecant ..... 776
\(\operatorname{csch}()\), hyperbolic cosecant ..... 776
csch \(^{-1}\) ( ),inverse hyperbolic cosecant ..... 776
cSolve( ), complex solve ..... 227, 776
cubic regression, CubicReg ..... 531
CubicReg, cubic regression ..... 531, 778
cumSum( ), cumulative sum ..... 519, 778
Current folder mode ..... 175
Current folder status ..... 34
Current mode ..... 15
cursor
3D graph ..... 368
deleting characters ..... 13
free-moving 305, 331, 338, 347, 367,405
hidden surface ..... 370
in the history area ..... 23
location following APD ..... 5
moving ..... 160
off the curve ..... 371
Selecting a command ..... 19
trace ..... 307
Viewing entries ..... 23
cursor keys \((\odot \odot \odot \odot)\)entering commands19
opening Apps ..... 25
selecting entry/answer pairs ..... 23
using the CHAR menu ..... 7
CustmOff, custom toolbar off ..... 218, 779
CustmOn, custom toolbar on ..... 218, 779
CUSTOM ([2nd [CUSTOM]) menu ..... 51
descripton ..... 50
key command ..... 13
CUSTOM axes (sequence) ..... 349
CUSTOM custom plots ..... 343, 420, 421
CUSTOM menu ..... 218
custom plots, CUSTOM . . 343, 420, 421custom toolbar See toolbar
Custom Units mode ..... 177
Custom, define toolbar ..... 605, 779
cut ..... 200, 637
cycle picture, CyclePic ..... 485, 613, 780
Cycle, cycle ..... 779
CyclePic, cycle picture ..... 613, 780
cylindrical vector display, Cylind ..... 780
cZeros( ), complex zeros ..... 227, 244, 780
D
\(d(\) ), first derivative \(73,234,252,254,782\)
darker/lighter ..... 135
data (new), NewData ..... 818
data filtering ..... 736
data plots ..... 106Data/Matrix Editor
cell width ..... 512
column header ..... 513, 516, 517, 518
filling ..... 510
list variable ..... 505, 507
new, NewData ..... 581
scrolling ..... 510
shift, shift( ) ..... 847
sorting columns ..... 520
statistical plots ..... 536
values ..... 508, 509
data/matrix editor ..... 454
date
reset ..... 44
setting ..... 36
dayOfWk( ), day of week ..... 782
DE (differential equation) mode ..... 34
decimal
angle display, DD ..... 782
integer display, Dec ..... 665, 782
define toolbar, Toolbar ..... 605, 863
Define, define . .343, 399, 432, 722, 783define, Define . 212, 258, 295, 336, 343,364, . 399, 432, 457, 464, 576, 581,611,722, 783
DEG (degree) mode ..... 34
degree notation, \({ }^{\circ}\) ..... 743, 882, 883
degree/minute/second display, 1 DMS 788
delete character ( \(\square / \square[\mathrm{DEL}])\) ..... 13
deleting
folder, DelFold ..... 581, 784
variable, DelVar 225, 257,581,584,784
deleting variables ..... 131
DelFold, delete folder ..... 581, 784
DelVar, delete variable ..... 581, 584, 784
denominator ..... 771
derivatives ..... 73
first derivative, d ( ) ..... 782
first derivative, \(d(\) ) 73, 234, 252, 254
numeric derivative, nDeriv( ) 253, 818
Derivatives (graph math tool) . .317,320,
332, ..... 339
deSolve( ), solution ..... 253, 430, 784
\(\operatorname{det}()\), matrix determinant ..... 786
diag( ), matrix diagonal ..... 786
dialog box
CLOCK ..... 37
edit categories ..... 30
menu indicator ..... 48
MODE ..... 14
to open Apps ..... 25
dialog box, define, Dialog ..... 605, 786
dialog boxes ..... 167
Dialog, define dialog box ..... 605, 786
Diamond modifier key ( \(\bullet\) ) description ..... 10
status ..... 34
differential equations
DIRFLD, direction field ..... 401, 409
first order ..... 411, 430
FLDOFF, field off ..... 401, 410, 437
graphing ..... 396-438
initial conditions ..... 407
second order ..... 413, 430
SLPFLD, slope field ..... 401,
solution methods ..... 400, 425
third order ..... 417
troubleshooting ..... 433
diftol window variable ..... 404
\(\operatorname{dim}()\), dimension ..... 787
dimension, \(\operatorname{dim}(\) ) ..... 586
direction field, DIRFLD ..... 401, 409, 436
DIRFLD, direction field ..... 401, 409, 436
Disp, display I/O screen . 118, 567, 604,
620, ..... 787
DispG, display graph ..... 604, 611, 787
DispHome, display Home screen ..... 604,

    787
display
graph, DispG ..... 604, 611, 787
Home screen, DispHome . .604, 787I/O screen, Disp 118, 567, 604, 620,787
table, DispTbl ..... 604, 611, 788
display as
binary, Bin ..... 665, 767
cylindrical vector, Cylind ..... 780
decimal angle, DD ..... 782
decimal integer, Dec ..... 665, 782
degree/minute/second, \(>\) DMS ..... 788
hexadecimal, Hex ..... 665, 804
polar vector, Polar ..... 827
rectangular vector, Rect ..... 835
spherical vector, Sphere ..... 854
Display Digits mode ..... 157, 175
display digits mode ..... 15
DispTbl, display table ..... 788
Distance (graph math tool) 317, 321, 332,339
divide, / ..... 875
domain constraints ..... 240
dot
addition, .+ ..... 876
division, ./ ..... 877
multiplication, .* ..... 877
power, .^ ..... 877
subtraction, .- ..... 876
dotP( ), dot product ..... 788
DrawFunc, draw function ..... 472, 616, 788
drawings and drawing
circle, Circle ..... 615, 769
circles ..... 478
clearing, CIrDraw ..... 614, 770
contour, DrwCtour ..... 616, 790
erasing ..... 477
freehand ..... 476
function, DrawFunc ..... 472, 616, 788
horizontal line, LineHorz ..... 616, 810
inverse, Drawlnv ..... 473, 616, 788
line, Line ..... 615, 810
lines ..... 477, 479
on a graph ..... 613
parametric, DrawParm 472, 616, 789
Pencil ..... 476
polar, DrawPol ..... 472, 616, 789
slope, DrawSlp ..... 480, 615, 789
tangent line, LineTan ..... 616, 810
vertical line, LineVert ..... 616, 810
Drawlnv, draw inverse ..... \(473,616,788\)
DrawParm, draw parametric . . 472, 616, ..... 789
DrawPol, draw polar ..... 472, 616, 789
DrawSlp, draw slope ..... 480, 615, 789
drop-down menu, DropDown ..... 606
DropDown, drop-down menu ..... 606, 789
DrwCtour, draw contour ..... 386, 616, 790
dtime window variable ..... 404
E
E, exponent ..... 12, 790
\(e\), natural log base ..... 263
\(e^{\wedge}(), e\) to a power ..... 790
editing ..... 160
eigVc( ), eigenvector ..... 791
eigVI( ), eigenvalue ..... 791
else if, Elself ..... 463, 591
Else, else ..... 805
else, Else ..... 591
Elself, else if ..... 591, 791
end
custom, EndCustm ..... 605, 779
dialog, EndDlog ..... 605, 786
for, EndFor ..... 567, 595, 798
function, EndFunc ..... 572, 799
if, Endlf ..... 567, 590, 805
loop, EndLoop ..... 598, 814
program, EndPrgm ..... 116, 576, 828
toolbar, EndTBar ..... 605, 863
try, EndTry ..... 620, 864
while, EndWhile ..... 597, 866
EndCustm, end custom ..... 605, 779
EndDlog, end dialog ..... 605, 786
EndFor, end for ..... 567, 595, 798
EndFunc, end function ..... 572, 799
Endlf, end if ..... 567, 590, 805
EndLoop, end loop ..... 598, 814
EndPrgm, end program ..... 116, 576, 828
EndTBar, end toolbar ..... 605, 863
EndTry, end try ..... 620, 864
EndWhile, end while ..... 597, 866
English category ..... 28
entry line
clearing the history area ..... 25
cursor rests on ..... 23
inserting commands ..... 19
recalling ..... 23
entry( ), entry ..... 792
entry, entry( ) ..... 207
entry/answer pairs ..... 24
status ..... 35
equal, = ..... 588, 878
equations, solving . . . 650, 654, 658, 659
error conditions after APD ..... 5
errors and troubleshooting
Circular definition ..... 580
clear error, ClrErr ..... 620, 770
Memory error ..... 697, 698
Out-of-memory ..... 260
pass error, PassErr ..... 620, 826
programs ..... 619
transmission ..... 705, 715
Estep window variable ..... 404
Euler method ..... 400, 425
evaluate polynomial, polyEval( ) ..... 827
EXACT mode status ..... 34
exact( ), exact ..... 792
Exact/Approx mode . 154, 176, 191, 226, 228, ..... 229
exact/approx mode ..... 15
example ..... 16
editing categories ..... 30
restoring the default custom menu ..... 51
selecting menu options ..... 46
turning off the clock ..... 43
turning on/off the custom menu ..... 51
using dialog boxes ..... 49
using the CHAR menu ..... 8
using the keyboard map ..... 9, 10
examples, previews, activities
3D graphing ..... 90, 729
additional graphing topics ..... 98
baseball ..... 743
CBL 2 program ..... 740
complex factors ..... 752
complex modulus surface ..... 388
complex numbers ..... 68
complex zeroes ..... 745
constants and measurement units ..... 75
converging web plots ..... 352
\(\cos (x)=\sin (x)\) activity ..... 727
cubic polynomial ..... 745
data filtering ..... 736
data/matrix editor ..... 104
decomposing a rational function ..... 734
derivatives ..... 73
differential equations ..... 94
diverging web plots ..... 354
expanding expressions ..... 70
factorial ..... 68
factoring polynomials ..... 246
Fibonacci sequence ..... 359
function graphing ..... 77, 80
graphing functions ..... 78
implicit plots ..... 393
integrals ..... 73
memory management ..... 126
number bases ..... 124
numeric solver ..... 121
oscillating web plots ..... 355
parametric graphing ..... 82, 743
path of a ball ..... 82
polar rose ..... 85
pole-corner problem ..... 721
population ..... 106
predator-prey model ..... 357, 421
prime factors ..... 68
programming ..... 115, 118, 621
Pythagorean theorem ..... 721
quadratic formula ..... 723
rational factors ..... 752
real factors ..... 752
reducing expressions ..... 71
sampling ..... 753
second-order differential equation 413, ..... 430
sequence graphing ..... 87
solving linear equations ..... 72, 247
split screen ..... 102, 743
standard annuity ..... 748
statistics ..... 106
symbolic manipulation ..... 73
tables ..... 100
text operations ..... 119
third-order differential equation ..... 417
time value of money ..... 750
trees and forest ..... 87
tutorial script with the text editor ..... 731
variable management ..... 126
exclusive or (Boolean), xor 589, 669, 867
exclusive or picture, XorPic ..... 613, 867
Exec, execute assembly language . 628,793
execute assembly language, Exec ..... 628,
793
execute program, Prgm ..... \(116,576,828\)
Exit, exit ..... 793
exprlist( ), expression to list ..... 793
expand( ), expand ..... 70, 725, 745, 793
expand, expand( ) 70, 243, 246, 725, 745
exponent key (2nd [EE]) ..... 12
exponent, E ..... 12, 790
Exponential Format mode ..... 158, 176, 897
exponential Format mode ..... 15
exponential regession, ExpReg 531, 795
expr( ), string to expression 586, 604, 794
ExpReg, exponential regession 531, 795
expressions ..... 21, 147, 148, 149, 160
expanding ..... 70
expression to list, \(\exp (\) list( ) ..... 793
reducing ..... 71
string to expression, expr( ) 586, 604,794
Extract menu ..... 244
eye \(\phi\) z-axis window variable . . .364, 373,
374
eye \(\theta\) x-axis window variable ..... 364,373,
374
eye \(\psi\) rotation window variable . 364,373 ,375
F
factor( ), factor 70, 725, 752, 795
factor, factor( ) ..... 70, 227, 243, 246, 725,
752
factorial,! ..... 69, 879
factoring ..... 246
activity ..... 752
false message ..... 262
family of curves ..... 464
Fibonacci sequence ..... 359
field off, FLDOFF ..... 401, 410, ..... 437
field picture, fldpic ..... 405
Fill, matrix fill ..... 796
Flash applications ..... 169, 183, 260, 673,
674, ..... 677
deleting ..... 705
FLASH APPLICATIONS ( \(\bullet\) APPS)
accessing Apps not listed ..... 54
description ..... 45
key command ..... 13
Flash, upgrading operating system ..... 711,
712
FLDOFF, field off ..... 401, 410, 437
fldpic, field picture ..... 405
fldres window variable ..... 404
floor( ), floor ..... 729, 796
floor, floor( ) ..... 729
fMax( ), function maximum ..... 797
fMin( ), function minimum ..... 797
FnOff, function off ..... 611, 797
FnOn, function on ..... 611, 797
folders ..... 175
delete, DelFold ..... 581, 784
get/return, getFold( ) ..... 801
locking/unlocking ..... 687
new, NewFold ..... 581, 819
pasting name ..... 689, 690
renaming ..... 683, 687
setting, setFold( ) ..... 601, 842
transmitting ..... 702, 703
VARLINK 677, 678, 679, 682, 683,684, .......................... 685
For, for ..... 798
for, For ..... 567, 595, 798
format string, format( ) 586, 604, 611
format( ), format string ..... 798
FORMATS dialog box 93, 94, 301, 379,
381, ..... 383, 641
fpart( ), function part ..... 799
fractions ..... 243, 251, 734, 829
free-moving cursor 305, 331, 338, 347,
367, ..... 405
Frobenius norm, norm( ) ..... 821
full-screen mode
2nd [QUIT] ..... 14
Apps desktop ..... 33
changing from split-screen ..... 59
displaying Apps in ..... 58
FUNC (function) mode ..... 34
Func, program function ..... 572, 799
function keys (ㅌ1-F8)
moving among toolbar menus ..... 50
selecting categories ..... 25, 28
selecting menus ..... 44
functions ..... 18, 147
delayed simplification ..... 233
graphing ..... 288
maximum, fMax() ..... 227, 253, 797
minimum, \(\mathrm{fMin}(\) ) ..... 227, 253, 797
multistatement ..... 462
off, FnOff ..... 297, 611, 797
on, FnOn ..... 297, 611, 797
part, fPart( ) ..... 799
program function, Func ..... 572, 799
user-defined 184, 211, 258, 364, 460,
\(462,570,572,783\)
G
Garbage collection message . . 693,694,
695, ..... 696
\(\operatorname{gcd}()\), greatest common divisor ..... 799
get time zone, GetTmZn( ) ..... 803
Get, get/return CBL/CBR value 553,618,799
get/return
calculator, GetCalc ..... 618, 708, 799
CBL/CBR value, Get . .553, 618, 799configuration, getConfg( ) . .601, 800denominator, getDenom( ) .244, 801folder, getFold( ) . . . . 581, 601, 801key, getKey( ) . . . . . . . . . . .603, 801mode, getMode( ) . . . . . . . . 601, 802
number, getNum( ) ..... 244, 802
type, getType( ) ..... 224, 581, 803
units, getUnits( ) ..... 601, 803
GetCalc, get/return calculator . 618, 708,799
getConfg(), get/return configuration 601,800
getDate( ), get date ..... 800
getDenom( ), get/return denominator ..... 801
getDtFmt( ), get date format ..... 801
getDtStr( ), get date string ..... 801
getFold( ), get/return folder ..... 581, 601
getKey( ), get/return key ..... 603, 801
getMode( ), get/return mode ..... 601, 802
getNum( ), get/return number ..... 802
getTime( ), get time ..... 802
getTmFmt( ), get time format ..... 802
getTmStr( ), get time string ..... 802
getTmZn( ), get time zone ..... 803
getType( ), get/return type ..... 581, 803
getUnits( ), get/return units ..... 601, 803
global variables ..... 584
go to, Goto ..... 578, 593, 600
Goto, go to ..... 804
graph
mode ..... 15, 34
number mode ..... 34
Graph 2 mode ..... 176
Graph mode 175, 191, 290, 328, 335,342, . . . . . . . . . . . . . . . . . 363, 398
graph mode status ..... 34
graph number mode status ..... 34
Graph Order graph format ..... 302, 400
Graph, graph ..... 804
graph, Graph ..... 295, 458, 465, 612
Graph<->Table, table-graph ..... 441
graphing category ..... 28
graphs
number of ..... 57, 58
graphs and graphing
\(\int f(x) d x\) ..... 317, 320
3D ..... 361-395
animation ..... 485
Arc ..... 317, 323, 332, 339
clearing, ClrGraph 460, 611, 661, 770
contour plots ..... 382, 386, 387
coordinates ..... 78, 305
custom axes ..... 349
custom plots ..... 343, 420, 421
Derivatives ..... 317, 320, 332, 339
differential equations ..... 396-438
Distance ..... 317, 321, 332, 339
drawing 474, 476, 477, 478, 479, 480,
481, 613
family of curves ..... 464
formats ..... 301, 330, 400
functions ..... 288
functions off, FnOff ..... 611, 797
functions on, FnOn ..... 611, 797
graph databases ..... 488
graph, Graph ..... 458, 612, 804
Home screen ..... 456, 458
implicit plots ..... 390, 393
independent variable ..... 456
Inflection ..... 317, 321
Intersection ..... 317, 319
inverse functions ..... 473
line styles ..297, 329, 337, 344, 364
math functions ..... 317
matrix data ..... 454
Maximum ..... 317, 319
Minimum ..... 78, 317, 319
modes 175, 191, 290, 328, 335, 342,363, 398
native independent variable ..... 456
nested functions ..... 461
operations ..... 760
overview . . 288, 326, 333, 340, 361, 396
panning ..... 309
parametric ..... 333
pausing ..... 304
pictures ..... 482, 484
piecewise functions ..... 460
polar ..... 326
programs ..... 611
QuickCenter ..... 310
recall graph database, RcIGDB .613,835
selecting functions ..... 295, 336, 344,399
sequence ..... 340-360
setting, setGraph( ) . . 601, 612, 842
Shade ..... 318, 323
shading, Shade 616, 846
simultaneous graphs ..... 466
split screen ..... 467, 470, 491
store graph database, StoGDB .613,855
style, Style ..... 612, 856
Tangent ..... 317, 322, 332, 339
text ..... 481
time plots ..... 343, 349, 420, 421
trace, Trace 307, 612, 730, 740, 742,745, 863
tracing .78, 307, 310, 332, 339, 348,367, ...................... . 406two-graph mode .467, 468, 469, 491Value . 317, 318, 332, 339, 348, 368,406
viewing window . 299, 329, 337, 345,365
web plots ..... 343, 349, 350
window variables 299, 329, 337, 345,365
\(Y=\) editor \(77,80,291,329,335,342\), 363, ..... 398, 456
Zero ..... 317, 319
zoom . 311, 331, 339, 348, 367, 612
zoom factors ..... 312, 314
zoom Memory ..... 312, 315
greater than or equal, \(\geq\), >= ..... 588, 879
greater than, > ..... 588, 879
greatest common divisor, gcd( ) ..... 799
Greek characters ..... 7, 641, 642
Grid graph format ..... 302
HHand modifier key (图)status34
Hardware version ..... 222
hexadecimal display, \(>\mathrm{Hex}\) 665, 804
indicator, Oh 886
hidden surface \(93,370,381\) highlighting
characters when editing 10
to view full name of App ..... 3
highlighting text ..... 636
Histogram ..... 542
History area
status ..... 35
history area ..... 195, 196, 648
History indicator ..... 24
Home icon ..... 21
Home screen ..... 193
Home screen. See calculator home screenhyperbolic
arccosine \(\cosh ^{-1}()\) ..... 774
arcsine, \(\sinh ^{-1}()\) ..... 850
arctangent, tanh \({ }^{-1}\) () ..... 860
cosecant, csch( ) ..... 776
cosine, cosh( ) ..... 774
cotangent, coth( ) ..... 775
secant, sech( ) ..... 840
sine, \(\sinh (\) ) ..... 850
tangent, tanh() ..... 859
I
ID list ..... 716, 718
ID number ..... 221, 711, 712, 716, 718
identity matrix, identity( ) ..... 805
identity( ), identity matrix ..... 805
If, if ..... 805
if, If ..... 463, 567, 590, 591
imag( ), imaginary part ..... 806
implicit plots ..... 390, 393
implied multiplication ..... 148, 336
Independent AUTO/ASK, independent auto/ask ..... 444, 449
independent auto/ask, Independent AUTO/ASK ..... 442, 444, 449
indirection, \# ..... 586, 881
infinity, \(\infty\) ..... 263
Inflection (graph math tool) ..... 317, 321
initial conditions ..... 407
initial startup ..... 2
input string, InputSt . .585, 603, 708, 807
Input, input ..... 806
input, Input ..... 603, 612
InputSt, input string . .585, 603, 708, 807
insert mode (2nd [INS]) ..... 14
inString( ), within string ..... 586, 807
instructions ..... 147
calculator Home screen ..... 21
Catalog ..... 18
insufficient display memory, <<...>> ..... 217
int( ), integer ..... 807
intDiv( ), integer divide ..... 667, 807
integer divide, intDiv( ) ..... 667
integer part, iPart( ) ..... 88, 807
integer, int( ) ..... 807
integrate, \(\int() 73,227,228,230,234,253\),254, 880
international/accented characters ..... 7
Intersection (graph math tool) . . 317, 319 Labels graph format ..... 302
inverse cosecant, \(\csc ^{-1}\) ( ) ..... 776
Language mode ..... 177
inverse cotangent, \(\cot ^{-1}()\) ..... 775
inverse hyperbolic
cosecant, \(\operatorname{csch}^{-1}()\) ..... 776
cotangent, coth \(^{-1}\) () ..... 775
secant, \(\operatorname{sech}^{-1}()\) ..... 840
inverse, \(\mathrm{x}^{-1}\) ..... 884
iPart( ), integer part ..... 807
isClkOn( ), is clock on ..... 808
isPrime( ), prime test ..... 808
Item, menu item ..... 606, 608, 808
K
key commands
keyboard map ..... 9
special characters ..... 7
keyboard ..... 136
2nd (second) key ..... 138
\(\dagger\) (shift) key ..... 138
- (diamond) key ..... 138
alpha (alpha) key ..... 138
key codes ..... 603
map ..... \(7,8,640,641\)
shortcuts ..... 640, 641
keys
function ..... 11
modifier ..... 10
language mode
changing mode setting ..... 16
viewing ..... 15
last answer ..... 141, 152, 205, 208
last entry ..... 141, 205, 206
Lbl, label ..... 808
lcm, least common multiple ..... 808
Leading Cursor graph format ..... 302
least common multiple, Icm ..... 808
left( ), left ..... 809
left, left( ) ..... 244, 586
left-right split screen setting ..... 54
setting initial Apps ..... 56
status ..... 32
less than or equal, \(\neq\), ..... 879
less than or equal, \(\leq,<=\) ..... 588
less than, < ..... 588, 878
lighter/darker ..... 135
limit( ), limit ..... 809
limit, limit( ) ..... 234, 253, 255
Line, draw line ..... 615, 810
linear regression, LinReg ..... 532, 811
LineHorz, draw horizontal line ..... 616, 810
LineTan, draw tangent line ..... 616, 810
LineVert, draw vertical line ..... 616, 810
Link transmission table ..... 720
linking and transmitting ..... 840, 841
L
label, Lbl ..... 578, 593, 600
calculator to calculator 617, 699, 701, 702, 708, 710, 711
cancelling ..... 704
errors ..... 705, 714, 715
Flash applications ..... 702, 706, 707
folders ..... 702, 703, 705
get/return CBL/CBR value, Get .553, 618, ..... 799
program ..... 617, 708
send chat, SendChat ..... 708
send list variable, Send ..... 619, 840
send to calculator, SendCalc ..... 618,708
variables ..... 702, 703
LinReg, linear regression ..... 532, 811
list difference, \(\Delta\) list( ) ..... 811
list to matrix, listrmat( ) ..... 811
listrmat( ), list to matrix ..... 517, 811
lists
augment/concatenate, augment( ) ..... 766
cross product, crossP( ) ..... 775
cumulative sum, cumSum( ) ..... 778 ..... 778
difference, slist( ) ..... 811
dot product, dotP( ) ..... 788
expression to list, exp>list( ) ..... 793
list to matrix, list>mat( ) ..... 811
matrix to list, mat>list( ) ..... 815
maximum, \(\max (\) ) ..... 815
mid-string, mid( ) ..... 816
minimum, \(\min (\) ) ..... 816
new data, NewData ..... 581, 818
new, newList( ) ..... 819
operations ..... 760
product, product( ) ..... 828
sort ascending, SortA ..... 854
sort descending, SortD ..... 854
summation, sum( ) ..... 837, 857
table variables ..... 451
variables ..... 501
\(\ln (\) ), natural logarithm ..... 811
LnReg, logarithmic regression . . 532, 812local variable, Local . .574, 579, 581, 582
Local, local variable ..... 812
lock variable, Lock ..... 581
Lock, lock variable ..... 812
locked/archived variable status ..... 35
log( ), logarithm ..... 813
logarithm, log( ) ..... 813
logarithmic regression, LnReg ..... 532, 812
logarithms ..... 811, 813
logistic regression, Logistic ..... 532
Logistic, logistic regression ..... 532, 813
Loop, loop ..... 814
loop, Loop ..... 598
LU, matrix lower-upper decomposition ..... 814
M
matrlist( ), matrix to list ..... 815
math category ..... 28
MATH menu ..... 317
MATH menu (2nd [MATH]) ..... 45
math operations ..... 21, 761
matrices
augment/concatenate, augment( ) 726, ..... 766
column dimension, colDim( ) ..... 771
column norm, colNorm( ) ..... 771
copying ..... 522
cumulative sum, cumSum( ) ..... 778
data from a graph ..... 454
determinant, det( ) ..... 786
diagonal, diag( ) ..... 786
dimension, \(\operatorname{dim}()\) ..... 787
dot addition, .+ ..... 876
dot division, ./ ..... 877
dot multiplication, .( ..... 877
dot power, .^ ..... 877
dot subtraction, .- ..... 876
eigenvalue, eigVI( ) ..... 791
eigenvector, eigVc( ) ..... 791
filling, Fill ..... 796
identity, identity( ) ..... 805
list to matrix, list>mat( ) ..... 811
locking ..... 515
lower-upper decomposition, LU ..... 814
matrix to list, mat>list( ) ..... 815
maximum, \(\max ()\) ..... 815
minimum, \(\min ()\) ..... 816
new data, NewData ..... 581, 818
new, newMat( ) ..... 819
operations ..... 761
pretty print ..... 504
product, product( ) ..... 828
QR factorization, QR ..... 832
random, randMat( ) ..... 726, ..... 834
reduced row echelon form, rref( ) 249, ..... 839
row addition, rowAdd( ) ..... 838
row dimension, rowDim( ) ..... 838
row echelon form, ref( ) ..... 836
row multiplication and addition, mRowAdd( ) ..... 817
row norm, rowNorm( ) ..... 839
row operation, mRow( ) ..... 817
row swap, rowSwap( ) ..... 839
submatrix, subMat( ) ..... 857
summation, sum( ) ..... 837, 857
transpose, \({ }^{\top}\) ..... 858
matrix to list, mat>list( ) ..... 815
max( ), maximum ..... 815
Maximum (graph math tool) ..... 317, 319
mean( ), mean ..... 815
measurement conversions (2nd [ \(\downarrow\) ]) ..... 13
median( ), median ..... 815
medium-medium line regression, MedMed532, 816
MedMed, medium-medium line regression532, 816
memory
archiving, Archive ..... 581, 692, 766
checking ..... 673, 674
insufficient display memory, <<...>>217
resetting 673, 674
unarchive, Unarchiv . . 582, 692, 864VARLINK screen \(675,677,678,679\),682, 683, 684, 685, 691
MEMORY (2nd [MEM]) ..... 14
Memory (zoom) ..... 312, 315
Memory error ..... 697
menu item, Item 606, 608, 808
Menus
APPLICATIONS ( \(\overline{\text { APPS }})\) ..... 45, 53
menus ..... 163
Algebra ..... 242, 245
APPLICATIONS ..... 169
Calc ..... 252
canceling ..... 50
CATALOG ..... 179
CHAR ..... 7, 14, 45
Clean Up ..... 177
Complex ..... 244
CUSTOM ..... 218
custom ..... 607, 610
CUSTOM (2nd [custom]) ..... 13, 50, 51
Extract ..... 244
FLASH APPLICATIONS ( \(\bullet\) APPS)13,45, 54
MATH ..... 317
options ..... 10
selecting options ..... 45
submenu options ..... 47
toolbar 163, 218
Trig ..... 244
using ..... 163
messages
BATT ..... 191
false ..... 262
Garbage collection . . 693, 694, 695, 696
insufficient display memory, <<...>>217
true ..... 262
undef (undefined) ..... 264
mid( ), mid-string ..... 816
mid-string, mid( ) ..... 587
\(\min ()\), minimum ..... 816
Minimum (graph math tool) .78, 317, 319
minute notation, ..... 883
\(\bmod ()\), modulo ..... 817
modes ..... 173
3D (three-dimensional) ..... 34
Angle \(15,34,175,291\), ..... 897
APPROX ..... 34
Approximate ..... 154, 176, 191
Apps desktop ..... 15
AUTO ..... 34
Auto ..... 154, 176, 191, 229
Base ..... 15, 176
Complex Format ..... 176, 897
complex format ..... 15
current ..... 15
Current folder ..... 175
custom units ..... 15
DE (differential equation) ..... 34
DEG (degree) ..... 34
Display Digits ..... 157, 175
display digits ..... 15
EXACT ..... 34
Exact/Approx ..... 154, 176, 191, 226,
228, ..... 229
exact/approx ..... 15
Exponential Format . . 158, 176, 897
exponential format ..... 15
fullscreen ..... \(14,27,33,56,58,59\)
FUNC (function) ..... 34
get/return, getMode( ) . . . 601, 802
Graph 175, 191, 290, 328, 335, 342,
363, ..... 398
graph ..... 15
Graph 2 ..... 176
graph number ..... 34
graph type ..... 34
grayed out ..... 15
insert (2nd [INS]) ..... 14
Language ..... 177
language ..... 15, 16
Number of Graphs ..... 176
overwrite ([2nd [INS]) ..... 14
PAR (parametric) ..... 34
POL (polar) ..... 34
Pretty Print ..... 15, 154, 176
RAD (radian) ..... 34
SEQ (sequence) ..... 34
setting in programs ..... 600
setting, setMode( ) ..... 601, 612, 843
settings ..... 14
Split App ..... 176
Split Screen ..... 176
split screen 3, 15, 27, 32, 34, 54, 56,58,59
Unit System ..... 177
unit system ..... 15
Vector Format ..... 176
vector format ..... 15
modifier keys (2nd \(\rightarrow\) 图) ..... 10
status ..... 34
move variable, MoveVar ..... 581
MoveVar, move variable ..... 817
mRow( ), matrix row operation ..... 817
mRowAdd( ), matrix row multiplication and addition ..... 817
multiply, * ..... 874
multistatement functions ..... 462
N
natural log base, e ..... 263
natural logarithm, \(\ln (\) ) ..... 811
ncontour window variable ..... 365
\(\mathrm{nCr}(\) ), combinations ..... 818
ncurves window variable ..... 403
nDeriv( ), numeric derivative ..... 253, 818
negate, - ..... 144, 877
negation key ([_]) ..... 11
negative numbers ..... 11
new
data, NewData ..... 554, 581, 818
folder, NewFold ..... 581, 819
list, newList( ) ..... 819
matrix, newMat( ) ..... 819
picture, NewPic ..... 582, 613, 819
plot, NewPlot ..... 541, 612, 820
problem, NewProb ..... 178, 820
NewData, new data \(503,516,554,581\),818
NewFold, new folder ..... 581, 819
newList( ), new list ..... 819
newMat( ), new matrix ..... 819
NewPic, new picture ..... 582, 613, 819
NewPlot, new plot ..... 541, 612, 819
NewProb, new problem ..... 178, 820
nInt( ), numeric integral ..... 253, 820
nmax window variable ..... 345
nmin window variable ..... 345
norm( ), Frobenius norm ..... 821
not (Boolean), not ..... 668, 821
not equal, \(\neq\), \(=\) ..... 588, 878
not, Boolean not ..... 589, 668, 821
\(n \operatorname{Pr}()\), permutations ..... 821
nSolve( ), numeric solution ..... 243, 822
number bases ..... 664
Boolean operations ..... 668
conversions ..... 665
math operations ..... 666
Number of Graphs mode ..... 176
numbers
irrational ..... 227, 228
negative ..... 144
rational ..... 227, 228, 229
numeric
derivative, nDeriv( ) ..... 253, 818
integral, nlnt( ) ..... 253, 820
solution, nSolve( ) ..... 243, 822
numeric keypad ..... 11
numeric solver ..... 650, 654, 658, 659
equations ..... 650, 652, 653
graphing ..... 659, 660, 661
split screens ..... 660, 662
variables ..... 654, 658
0
on/off ..... 133
OneVar, one-variable statistic ..... 822
operating system ..... 713, 714
operating system (OS) downloading ..... 66
Operating System (OS) version ..... 222
operating system, upgrading ..... 711, 712
operators ..... 147
or (Boolean), or ..... 669
or, Boolean or ..... 589, 669, 823
ord( ), numeric character code ..... 587, 823
Organizr (organizer) category ..... 28
OS ..... 711, 712
OS (Operating System) version ..... 222
Out-of-memory error ..... 260
Output, output ..... 823
output, Output ..... 604, 611
overwrite mode (2nd [INS]) ..... 14
P
\(\mathrm{P} \boldsymbol{R x}(\) ) , rectangular \(x\) coordinate ..... 824
P>Ry(), rectangular y coordinate ..... 824
panning ..... 309
PAR (parametric) mode ..... 34
parallelepiped activity ..... 729
parametric graphing ..... 333
parentheses, brackets, and braces ..... 148
part( ), part ..... 824
pass error, PassErr ..... 620
PassErr, pass error ..... 826
paste ..... 200, 637
PAUSE ..... 35
PAUSE indicator ..... 192
Pause, pause ..... 826
pause, Pause ..... 604, 620
percent, \% ..... 877
permutations, \(n \operatorname{Pr}(\) ) ..... 821
pictures ..... 482, 484
and, AndPic ..... 612, 765
cycle, CyclePic ..... 613, 780
deleting ..... 485
exclusive or, XorPic ..... 613, 867
new, NewPic ..... 582, 613, 819
recall, RcIPic ..... 613, 835
replace, RplcPic ..... 613, 839
storing, StoPic ..... 613, 856
piecewise functions ..... 460
pixel
change, PxIChg ..... 615, 830
circle, PxiCrcl ..... 615, 830
horizontal line, PxIHorz ..... 616, 830
line, PxILine ..... 481, 615, 830
off, PxIOff ..... 615, 831
on, PxiOn ..... 481, 615, 831
test, pxITest( ) ..... 615, 831
text, PxIText ..... 615, 831
vertical line, PxIVert ..... 616, 831
plots
clearing ..... 539
data ..... 106
new, NewPlot ..... 541, 612, 820
off, PlotsOff ..... 297, 612, 827
on, PlotsOn ..... 297, 612, 827
selecting ..... 538, 545
tracing ..... 547
viewing window ..... 546
\(\mathrm{Y}=\) Editor ..... 544
PlotsOff, plots off ..... 297, 827
PlotsOn, plots on ..... 297, 827
plotStep window variable ..... 345
plotStrt window variable ..... 345
point
change, PtChg ..... 615, 829
off, PtOff ..... 615, 829
on, PtOn ..... 615, 829
test, ptTest( ) ..... 615, 829
text, PtText ..... 615, 830
POL (polar) mode ..... 34
polar
coordinate, R>P日( ) ..... 834
coordinate, R>Pr() ..... 834
graphing ..... 326
vector display, Polar ..... 827
polyEval( ), evaluate polynomial ..... 827
polynomials ..... 246, 254
activity ..... 745
evaluate, polyEval( ) ..... 827
random, randPoly( ) ..... 834
popup menu, PopUp ..... 603
PopUp, popup menu ..... 603, 827
power of ten, 10^() ..... 884
power regression, PowerReg ..... 532, 828
power, ^ ..... 876
PowerReg, power regression ..... 532, 828
pretty print ..... 78, 154, 194
Pretty Print mode ..... 15, 154, 176
Prgm, execute program 116, 576, ..... 828
prime number test, isPrime( ) ..... 808
prime numbers ..... 70
prime, ..... 883
problems (new), NewProb ..... 178, 820
product ID ..... 221
Product ID (identifier) ..... 222
product( ), product ..... 828
product, П() ..... 253, 881
Program Editor ..... 26
programs and programming 18,557-628
arguments ..... 569
assembly language ..... 625, 628
branching ..... 567, 590, 593
calling another program ..... 576
CBL ..... 617
CBL 2 system ..... 740
CBR ..... 617
CBR system ..... 740
clear error, ClrErr ..... 620, 770
clear graph, ClrGraph .460, 611, ..... 770
clear home, ClrHome ..... 771
clear I/O, ClrIO ..... 560, 604, 771
clear table, ClrTable ..... 771
comment, © ..... 566, 886
conditional tests ..... 587
copying ..... 564
custom toolbar off, CustmOff ..... 218,
605, ..... 779
custom toolbar on, CustmOn ..... 218,
605, ..... 779
debugging ..... 620
define dialog box Dialog ..... 605, 786
define toolbar, Custom ..... 605, 779
define toolbar, Toolbar ..... 605, 863
define, Define . . .576, 611, 722, ..... 783
deleting ..... 564
display graph, DispG . .604, 611, 787display Home screen, DispHome604,787
display I/O screen, Disp ..... 118,567,
604, ..... 620, 787
display table, DispTbl .604, 611, 788
drop-down menu, DropDown ..... 606,789else if, Elself . . . . . . . . 463, 591, 791
else, Else ..... 591, 805
end custom, EndCustm ..... 605, 779
end dialog, EndDlog ..... 605, 786
end for, EndFor ..... 567, 595, 798
end function, EndFunc ..... 572, 799
end if, Endlf 567, 590, 591, 805
end loop, EndLoop ..... 598, 814
end program, EndPrgm 116, ..... 576, 828
end toolbar, EndTBar ..... 605, 863
end try, EndTry ..... 620, 864
end while, EndWhile ..... 597, 866
entering ..... 561, 564
execute assembly language, Exec 628, ..... 793
execute program, Prgm ..... 116,576,828
exit, Exit ..... 793
for, For ..... 567, 595, 798
format string, format( ) 604, 611, 798function, Func . . . . . . . . . . . 572, 799
functions ..... 561, 570, 572
get/return configuration, getConfg( )601,800
get/return folder, getFold( ) 601, 801get/return from calculator, GetCalc618,708, 799
get/return key, getKey( ) ..... 603, 801
get/return units, getUnits( ) 803
go to, Goto ..... 578, 593, 600, 804
graphs ..... 611
if, If ..... \(463,567,590,591,805\)
input ..... 559, 567, 603
input, Input ..... 603, 612, 806
label, Lbl ..... 578, 593, 600, 808
local, Local .574, 579, 581, 582, 812
loop, Loop ..... 598, 814
looping ..... 567, 595, 597
menu item, Item ..... 606, 608, 808
menus ..... 606, 610
multicommand lines ..... 565
operations ..... 762
output ..... 559, 567, 604
output, Output ..... 604, 611, 823
pass error, PassErr ..... 620, 826
passing values ..... 569
pause, Pause ..... 604, 620, 826
popup menu, PopUp ..... 603, 827
prompt, Prompt( ) ..... 603, 828
request, Request 604, 606, 836
return, Return ..... 574, 577, 837
running ..... 557
stop, Stop ..... 564, 855
stopping ..... 558
subroutines ..... 576
tables ..... 611
text, Text ..... 605, 606, 861
Then, Then ..... 590, 591, 805
title, Title ..... 606, 862
try, Try ..... 620, 864
variables ..... 578
while, While ..... 597, 866
Prompt( ), prompt ..... 603, 828
prompt, Prompt( ) ..... 603
proper fraction, propFrac ..... 243, 251, 734
propFrac, proper fraction ..... 243, 251, 734,
829
PtChg, point change ..... 615, 829
PtOff, point off ..... 615, 829
PtOn, point on ..... 615, 829
ptTest( ), point test ..... 615, 829
PtText, point text ..... 615, 830
PxIChg, pixel change ..... 615, 830
PxICrcl, pixel circle ..... 615, 830
PxlHorz, pixel horizontal line ..... 616, 830
PxILine, pixel line ..... 481, 615, 830
PxIOff, pixel off ..... 615, 831
PxiOn, pixel on ..... 481, 615, 831
pxITest( ), pixel test ..... 615, 831
PxIText, pixel text ..... 615, 831
PxIVert, pixel vertical line ..... 616, 831
Q
QR factorization, QR ..... 832
QR, QR factorization ..... 832
quadratic regression, QuadReg ..... 532, 832
QuadReg, quadratic regression ..... 532, 832
quartic regression, QuartReg ..... 533, 833
QuartReg, quartic regression ..... 533, 833
QuickCenter ..... 310
Quit (2nd [QUIT]) ..... 14
R
r , radian ..... 882
\(R>P \theta()\), polar coordinate ..... 833
\(\operatorname{R}>\operatorname{Pr}(\) ), polar coordinate ..... 834
RAD (radian) mode ..... 34
radian, \({ }^{\text {r }}\) ..... 882
rand( ), random number ..... 834
randMat( ), random matrix ..... 726, 834
randNorm( ), random norm ..... 834
random
matrix, randMat( ) ..... 726, 834
norm, randNorm( ) ..... 834
number seed, RandSeed ..... 726, 834
number, rand() ..... 834
polynomial, randPoly( ) ..... 834
randPoly( ), random polynomial ..... 834
RandSeed, random number seed ..... 726,
834
rational functions activity ..... 734
RclGDB, recall graph database 490,613, 835
RclPic, recall picture ..... 613, 835
real( ), real ..... 835
recall
graph database, RcIGDB . 490,613, 835picture, RcIPic613, 835
Recall (2nd [RCL]) ..... 14
reciprocal, \(\mathrm{x}^{-1}\) ..... 884
rectangular \(x\) coordinate, \(P>R x()\) ..... 824
rectangular y coordinate, P>Ry() ..... 824
rectangular-vector display, Rect ..... 835
reduced row echelon form, rref( ) ..... 249,
727
ref( ), row echelon form ..... 836
regressions ..... 811
cubic, CubicReg ..... 531, 778
exponential, ExpReg ..... 531, 795
linear regression, LinReg ..... 532, 811
logarithmic, LnReglogistic, Logistic
532, 813
medium-medium line, MedMed .532,816
power regression, PowerReg ..... 532,
828
quadratic formula activity ..... 723
quadratic, QuadReg ..... 532, 832
quartic, QuartReg ..... 533, 833
selecting ..... 531
sinusoidal, SinReg ..... 533, 851
remain( ), remainder ..... 836
Rename, rename ..... 836
rename, Rename ..... 582
replace picture, RplcPic ..... 613, 839
Request, request ..... 836
request, Request ..... 604, 606
results ..... 21
return See get/return
Return, return ..... 837
return, Return
right( ), right ..... 837
right, right( ) ..... 244, 587
rotate( ), rotate ..... 670, 837
rotate, rotate( ) ..... 587, 670
round( ), round ..... 838
row echelon form, ref( ) ..... 836
rowAdd( ), matrix row addition ..... 838
rowDim( ), matrix row dimension ..... 838
rowNorm( ), matrix row norm ..... 839
rowSwap( ), matrix row swap ..... 839
RplcPic, replace picture ..... 613, 839
rref( ), reduced row echelon form ..... 249,
727, ..... 839
Runge-Kutta method ..... 400, 421
S
sampling activity ..... 753
SAVE COPY AS ( \(\quad\) S) example ..... 49
Scatter plots ..... 540
scientific notation ..... 12, 145
scripts ..... 198, 645, 648
activity ..... 732
tutorial ..... 732
scrolling ..... 23, 216, 448
sec( ), secant ..... 840
\(\sec ^{-1}()\), inverse secant ..... 840
secant, sec( ), ..... 840
sech( ), hyperbolic secant ..... 840
\(\operatorname{sech}^{-1}()\),inverse hyperbolic secant ..... 840
Second modifier key (2nd)
description ..... 10
status ..... 34
second notation, ..... 883
selecting categories ..... 28
send chat, SendChat ..... 708, 841
send list variable, Send 619, 840
send to calculator, SendCalc ..... 618,708,
840
Send, send list variable ..... 619, 840
SendCalc, send to calculator ..... 618, 708, 840
SendChat, send chat ..... 708, 841
SEQ (sequence) mode ..... 34
seq( ), sequence ..... 841
sequence graphing ..... 340-360
serial number ..... 221
set
folder, setFold( ) ..... 601, 842
graph, setGraph( ) ..... 601, 612
mode, setMode( ) ..... 601, 612
table, setTable( ) ..... 444, 601
units, setUnits( ) ..... 601
Set factors (zoom) ..... 312, 314
setDate( ), set date ..... 842
setDtFmt( ), set date format ..... 842
setFold( ), set folder ..... 601, 842
setGraph( ), set graph ..... 601, 612, 842
setMode( ), set mode ..... 601, 612, 843
setTable( ), set table ..... 444, 601, 844
setTime( ), set time ..... 844
setTmFmt( ), set time format ..... 844
setTmZn( ), set time zone ..... 845
setUnits( ), set units ..... 601, 845
Shade (graph math tool) ..... 318, 323
Shade, shade ..... 616, 846
shade, Shade ..... 616
Shift modifier key ( \(\quad\) ) description ..... 10
status ..... 34
shift( ), shift ..... 518, 587, 671, 847
shift, shift( ) ..... 587, 671
show statistical results, ShowStat . . 533,848
ShowStat, show statistical results ..... 533,
848
sign( ), sign ..... 848
simplification delayed ..... 233
rules ..... 230
stopping ..... 233
simult( ), simultaneous equations ..... 848
simultaneous equations, simult( ) ..... 249
\(\sin ()\), sine ..... 849
\(\sin ^{-1}()\), arcsine ..... 849
sinh( ), hyperbolic sine ..... 850
\(\sinh ^{-1}()\), hyperbolic arcsine ..... 850
SinReg, sinusoidal regression ..... 533, 851
sinusoidal regression, SinReg ..... 533, 851
slope field, SLPFLD 401, 409, ..... 434
SLPFLD, slope field 401, 409, ..... 434
Smart Graph ..... 304
SocialSt (social studies) category ..... 28
software version ..... 221
Solution Method graph format ..... 400
solution, deSolve( ) ..... 253, 430, 784
solve( ), solve 72, 74, 432, 851
solve, solve( ) 72, 74, 227, 228, 230, 234,
240, 243, 247, 432
solving linear equations ..... 72, 247
SortA, sort ascending ..... 854
SortD, sort descending ..... 854
special characters ..... 639, 640, 642
spherical vector display, (Sphere ..... 854
Split App mode ..... 176
split screen ..... 467, 470, 647, 660, 662
entry line ..... 497, 499
exiting ..... 494
setting ..... 491
switch, switch( ) ..... 601, 857
switching ..... 497
Split Screen mode ..... 176
split-screen mode
active graph ..... 34
exiting ..... 59
Number of graphs ..... 57
returning from within an App ..... 27
selecting active App ..... 58
setting ..... 54
setting initial Apps ..... 56
specifying Apps displayed ..... 56
Split 1 App ..... 56
Split 2 App ..... 56
status ..... 32
status and open Apps ..... 3
viewing ..... 15
square root, \(\sqrt{ }()\) ..... 881
standard annuity activity ..... 748
standard deviation, stdDev( ) ..... 855
start timer, startTmr( ) ..... 854
startTmr( ), start timer ..... 854
statistics ..... 525-556
Box Plot ..... 541
Calculation Type ..... 526, 531
categories ..... 550
Category ..... 526, 528
combinations, \(\mathrm{nCr}(\) ) ..... 818
factorial, ! ..... 69, 879
Freq ..... 526, 528
frequency ..... 548
Histogram plots ..... 542
mean, mean( ) ..... 815
median, median() ..... 815
new plot, NewPlot ..... 541, 820
operations ..... 763
overview ..... 525
permutations, \(n \operatorname{Pr}(\) ) ..... 821
plots .536, 538, 539, 543, 545
plots off, PlotsOff 297, 612, ..... 827
plots on, PlotsOn ..... 297, 612, ..... 827
random norm, randNorm( ) ..... 834
random number seed, RandSeed 726, ..... 834
random number, rand( ) ..... 834
Scatter plots ..... 540
show results, ShowStat ..... 533, 848
standard deviation, stdDev( ) ..... 855
two-variable results, TwoVar ..... 531,
864
variables ..... 533
variance, variance( ) ..... 865
xyline plots ..... 540
statistics. See also regressions status
battery low ..... 64
on Apps desktop ..... 3
split-screen ..... 32
status line ..... 189, 291
command parameters ..... 20
history information ..... 24
stdDev( ), standard deviation ..... 855
StoGDB, store graph database 490,613 , 855
Stop, stop ..... 855
stop, Stop ..... 564
StoPic, store picture ..... 613, 856
stopping a calculation ..... 153
Store (STO』) key ..... 14
storing
graph database, StoGDB ..... 613, 855
picture, StoPic ..... 613, 856
symbol, \(\rightarrow\) ..... 581, 885
string( ), expression to string ..... 587, 856
strings
append, \& ..... 586, 880
character code, ord() ..... 587
character string, char( ) . . . 586, 769
dimension, \(\operatorname{dim}(\) ) . . . . . . . . . . . . 586
expression to string, string ( ) . . 587, 856
format, format( ) .586, 604, 611, 798
indirection, \# . . . . . . . . . . . 586, 881
inputting, InputSt . . . . .585, 603, 708
left, left( ) . . . . . . . . . . . . . . . 586, 809
mid-string, mid( ) . . . . . . . . 587, 816
operations . . . . . . . . . . 585, 586, 763
right, right( ) . . . . . . . . . . . . . 587, 837
rotate, rotate( ) . . . . . . . . . . 587, 837
shift, shift( ) . . . . . . . . . . . . 587, 847
string to expression, expr( ) 586,604, 794
within, InString . . . . . . . . . 586, 807
Style, style . . . . . . . . . . . . . . . . . . . . . 856
style, Style . . . . . . . . . . . . . . . . . 299, 612
subMat( ), submatrix . . . . . . . . . . . . . 857
submenus . . . . . . . . . . . . . . . . . . . . 165
substitutions .............235, 238, 240
subtract, - . . . . . . . . . . . . . . . . . . . . . 874
subtraction key ( \(\square\) ) . . . . . . . . . . . . . . . 11
sum( ), summation . . . . . . . . . . 837, 857
sum, \(\Sigma(\) ) . . . . . . . . . . . . . . . . . . . 253, 881
switch( ), switch . . . . . . . . . . . . . . . . . 857
switch, switch( ) . . . . . . . . . . . . . . . . . 601
symbolic manipulation . . . . . . . . . . . . 256
system data, sysData . . . . . . . .454, 455
T
T, transpose ..... 858
to window variable ..... 402
TABLE SETUP, table setup ..... 440
Table, build table ..... 858
table-graph, Graph<->Table ..... 441
tables ..... 441
\(\Delta \mathrm{tbl}\)
444
automatic ..... 446, 452
clearing, CIrTable ..... 771
complex numbers ..... 448
differential equations ..... 438
displaying, DispTbl ..... 604, 611, 788
functions ..... 448
generating with sequence ..... 359
graphing, Graph<->Table ..... 441
incrementing, \(\Delta\) tbl ..... 441
Independent AUTO/ASK ..... 442, 444,
449
manual ..... 449
overview ..... 439
programs ..... 611
setTable( ) ..... 444
setting, setTable( ) ..... 601, 844
setup ..... 444
setup, TABLE SETUP ..... 440
starting, tblStart ..... 441
tbIStart ..... 441
tan( ), tangent ..... 859
\(\tan ^{-1}()\), arctangent ..... 859
Tangent (graph math tool) 317, 322, 332,339
tangent, \(\tan ()\) ..... 859
tanh( ), hyperbolic tangent ..... 859
tanh \(^{-1}()\), hyperbolic arctangent ..... 860
Taylor polynomial, taylor( ) 253, 256, ..... 860
taylor( ), Taylor polynomial ..... 860 ..... 860
tblStart, table start ..... 441
tCollect( ), trigonometric collection ..... 861
temperature conversion, \(\operatorname{tmpCnv}()\) ..... 272,
862
temperature-range conversion, \(\Delta t m p C n v()\) ..... 272
temperature-range conversion, \(\Delta t m p C n v()\) ..... 863
tExpand( ), trigonometric expansion ..... 861
text editing ..... 629, 639, ..... 645
cut, copy, paste ..... 200, 637
find ..... 638
highlighting ..... 636
Text, text ..... 861
text, Text ..... 605, 606
Then, Then ..... 590, 591, 805
three-dimensional graphing ..... 361-395
animation ..... 90, 377
CONTOUR LEVELS ..... 93, 381
HIDDEN SURFACE ..... 93, 381
WIRE AND CONTOUR ..... 93, 381
WIRE FRAME ..... 93, 381
TI Connectivity Cable .60, 699, 712, 717TI ViewScreen overhead panel
connecting ..... 62
TI Connect software ..... 60, 712
time
reset ..... 44
setting ..... 36
time plots, TIME 343, 349, 420, ..... 421
time value of money activity ..... 750
TIME, time plots ..... 343, 420, ..... 421
timeCnv( ), convert time ..... 861
TI-Presenter video adapter connecting ..... 62
Title, title ..... 862
tmax window variable ..... 337, 402
tmin window variable ..... 337
\(\operatorname{tmpCnv}()\), temperature conversion ..... 272,
862
toolbar
define, Custom ..... 605, 779
off, CustmOff ..... 218, 779
on, CustmOn ..... 218, 779
Toolbar menus
calculator Home screen ..... 44
moving among ..... 50
replaced by custom menu ..... 50
selecting math operations ..... 11
Toolbar, toolbar ..... 605, 863
top-bottom split screen
54
setting
56
setting initial Apps32
tplot window variable ..... 403
Trace, trace ..... 730, 740, 742, 745, 863
trace, Trace 307, 612, 730, 740, 742, 745tracing 78, 307, 310, 332, 339, 348, 367,406
transmitting. See linking and transmittingtranspose, \({ }^{\top}\)858
Trig menu ..... 244
trigonometric collection, tCollect( ) ..... 244
expansion, tExpand() ..... 244
trigonometric collection, tCollect() ..... 861
trigonometric expansion, tExpand() ..... 861
true message ..... 262
Try, try ..... 864
try, Try ..... 620
tstep window variable ..... 337, 402
turning clock off, ClockOff ..... 769
turning clock on, ClockOn ..... 770
turning off ..... 4
after APD ..... 5
following inactivity ..... 5
turning on initial startup ..... 2
TwoVar, two-variable results ..... 531
two-variable results, TwoVar ..... 531, 864
typing
file name ..... 26
to scroll through Catalog ..... 18
UUnarchiv, unarchive variables .582, 692,864unarchive variables, Unarchiv .582, 692,864
undef (undefined) message ..... 264
underscore, ..... 883
Unit ID (identifier) ..... 222
Unit System mode ..... 177
unit System mode ..... 15
unit vector, unitV( ) ..... 865
units
converting ..... 269
defaults ..... 273, 279
displaying ..... 273
get/return, getUnits( ) ..... 803
measurement ..... 265
modes ..... 177
setting, setUnits( ) ..... 601, 845
user-defined ..... 276
unit-to-unit cable ..... 66
connecting ..... 62
unitV( ), unit vector ..... 865
Unlock, unlock ..... 865
unlock, Unlock ..... 582
upgrading operating system (OS) ..... 711,
712
user-defined functions ..... 184, 211, 258,
364, \(460,462,570,572,783\)
user-defined units ..... 276
v
Value (graph math tool) . . 317, 318, 339,
368, ..... 406
variables ..... 35, 185, 187
archiving and unarchiving ..... 691
archiving, Archive ..... 581, 692, 766
clearing ..... 663
copy, CopyVar ..... 581, 687, 772
copying ..... 687
data ..... 502
defined ..... 223, 655
delayed simplification ..... 233
delete, DelVar ..... 225, 257, 581, 584,
784
deleting ..... 705
in applications ..... 689, 690
local, Local .574, 579, 581, 582, ..... 812
locking, Lock ..... 581
locking/unlocking ..... 192, 687
matrix ..... 504
moving, MoveVar ..... 581
overriding ..... 226
pasting name ..... 689, 690
recall ..... 14
referring to App files ..... 25
renaming ..... 683
statistical ..... 529, 533
store ..... 14
text ..... 198
transmitting ..... 699, 701, 703
unarchive, Unarchiv ..... 582, 692, 864
undefined ..... 223, 655
unknown, solving for ..... 654, 658
unlocking, Unlock ..... 582
VARLINK . 675, 677, 678, 679, 682,683, ............. 684, 685, 691variance( ), variance865
Vector Format mode ..... 176
vector format mode ..... 15
vectors
cross product, crossP( ) ..... 775
cylindrical vector display, Cylind ..... 780
dot product, dotP( ) ..... 788
unit, unitV( ) ..... 865
Vector Format mode ..... 176
viewing angle ..... 372
viewing orbit ..... 377
Wweb plots
convergence ..... 352
divergence ..... 354
oscillation ..... 355
WEB ..... 343, 349, 350
WEB, web plots ..... 343, 349, 350
when( ), when ..... 865
when, when( ) ..... 460
While, while ..... 866
while, While ..... 597
Window Editor ..... 53
window variables
diftol ..... 404
dtime ..... 404
Estep ..... 404
eye \(\phi\) (z axis) ..... 364, 373, 375
eye日 (x axis) ..... 364, 373, 375
eye\% (rotation) 364, 373, ..... 375
fldres ..... 404
ncontour ..... 365
ncurves ..... 403
nmax ..... 345
nmin ..... 345
plotStep ..... 345
plotStrt ..... 345
\(\theta \max\) ..... 329
\(\theta\) min ..... 329
\(\theta\) step ..... 330
t0 ..... 402
tmax ..... 337, 402
tplot ..... 403
tstep ..... 337, 402
xgrid ..... 365
xmax 299, 330, 337, 345, 365, ..... 403
xmin . 299, 330, 337, 345, 365, ..... 403
xres ..... 300
xscl ..... 300, 330, 338, 345, 403
ygrid ..... 365
ymax . 299, 330, 337, 345, 365, ..... 403
ymin . 299, 330, 337, 345, 365, 403
zmax ..... 365
zmin ..... 365
wire-and-contour graphing ..... 93, 381
wire-frame graphing ..... 93, 381
with, 73, 74, 226, 235, 885
within string, inString( ) 586, 807
X
xgrid window variable ..... 365
xmax window variable ..... 299, 330, 337,
345 , ..... 365, 403
xmin window variable 299, 330, 337, 345,365, 403
xor, Boolean exclusive or . 589, 669, 867
XorPic, exclusive or picture ..... 613, 867
xres window variable ..... 300
xscl window variable \(300,330,338,345\),403
xyline plots ..... 540
Y\(Y=\) editor 77, 80, 291, 329, 335, 342, 363,398, 456
ygrid window variable ..... 365
ymax window variable ..... 299, 330, 345,
365 , ..... 403
ymin window variable 299, 330, 337, 345,365, 403
yscl window variable 300, 330, 338, 345,403
Z
Zero (graph math tool) ..... 317, 319
zeroes
activity ..... 745
zeroes, zeroes() ..... 227, 243, 250
zeroes, zeros() ..... 723, 868
zeros( ), zeroes ..... 723, 868
zmax window variable ..... 365
zmin window variable ..... 365
zoom
box, ZoomBox ..... 311, 313, 869
data, ZoomData ..... 312, 870
decimal, ZoomDec ..... 311, 870
factors ..... 312, 314
fit, ZoomFit ..... 312, 871
in, Zoomln ..... 311, 314, 871
integer, Zoomlnt ..... 312, 871
Memory ..... 312, 315
out, ZoomOut ..... 311, 314, 872
previous, ZoomPrev ..... 316, 872
recall, ZoomRcl ..... 316, 872
square, ZoomSqr ..... 311, 872
standard, ZoomStd ..... 312, 873
store, ZoomSto ..... 315, 316, 873
trig, ZoomTrig ..... 312, 873
Zoom menu ..... 311
ZoomBox, zoom box ..... 869
ZoomData, zoom data ..... 870
ZoomDec, zoom decimal ..... 870
ZoomFit, zoom fit ..... 871
Zoomln, zoom in ..... 871
Zoomlnt, zoom integer ..... 871
ZoomOut, zoom out ..... 872
ZoomPrev, zoom previous ..... 872
ZoomRcl, zoom recall ..... 872
ZoomSqr, zoom square ..... 872
ZoomStd, zoom standard ..... 873
ZoomSto, zoom store ..... 873
ZoomTrig, zoom trig ..... 873```

