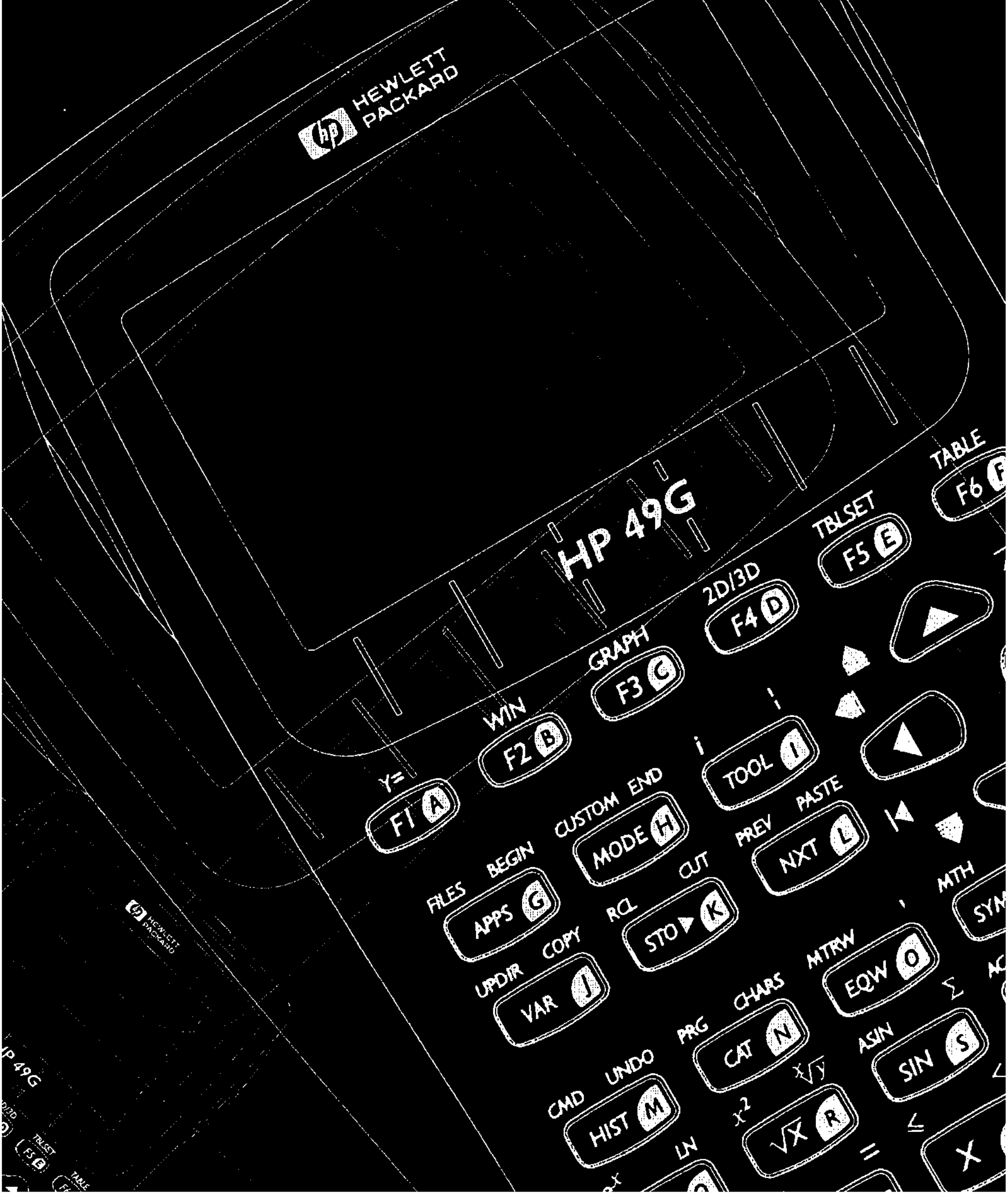
HEWLETT®
PACKARD

HP 49G

ADVANCED USER'S GUIDE

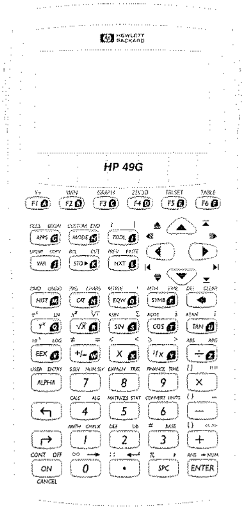


HP 49G

GRAPHING CALCULATOR

ADVANCED USER'S GUIDE

Version 1



HEWLETT PACKARD

Contents

Chapter 6: Unit objects 6-1

Chapter 7: Constants library 7-1

Chapter 8: Number bases 8-1

Chapter 9: Lists and Sequences 9-1

Chapter 10: Advanced plotting options 10-1

Chapter 13: Customization 13-1

Chapter 14: Computer Algebra Commands 14-1

Index I-1

Preface

This guide contains information on the advanced functionality of the HP 49G. It is a supplement to the *Pocket Guide* and *User's Guide* that is shipped with the HP 49G.

Material covered

This guide contains the following information:

- **Chapter 1, Entering commands** contains information on the different ways that you can enter and use commands.
- **Chapter 2, System flags** contains information on using the HP 49G's flags, and the commands to control them.
- **Chapter 3, Command line operations** illustrates how to use the command line editor.
- **Chapter 4, The stack** explains how to operate the calculator in RPN mode, and how to manipulate objects on the stack.
- **Chapter 5, Matrices and linear algebra** describes advanced matrix operations, and how to solve linear algebra problems using matrices.
- **Chapter 6, Unit objects** contains information on using the Units application to work with units in mathematical operations.
- **Chapter 7, Constants library** contains information on the library of physical constants that you can access and use in calculations.
- **Chapter 8, Number bases** contains information on working with binary, octal and hexadecimal number bases, and working with binary integer arithmetic.
- **Chapter 9, Lists and sequences** contains information on creating and working with lists and sequences.
- **Chapter 10, Advanced plotting options** provides additional information on creating and manipulating plots.
- **Chapter 11, Memory** contains information on storing and retrieving data, using library objects, and how the HP 49G manages memory.
- **Chapter 12, Date and time arithmetic** describes how to work with dates and times.
- **Chapter 13, Customization** describes how to customize the keyboard, and how to create menus.

- **Chapter 14, Computer algebra commands** describes each of the computer algebra commands that the calculator contains, and what each command does.

The HP 49G documentation set

The HP 49G documentation set is a mixture of hard copy documentation and documentation available from the HP Calculators web site.

You can find all of the documentation for the HP 49G on the HP Calculators web site, at the following address:

http://www.hp.com/calculators/graphing/49g_info.html

The HP 49G documentation set is made up of the following components:

- *User's Guide*, available with the calculator.
- *Pocket Guide*, available with the calculator.
- *Advanced Topics*, being chapters 1 to 13 of this manual, and available from the web site.
- *CAS Commands*, being chapter 14 of this manual, and available from the web site.
- *Other Commands A to F*, available from the web site.
- *Other Commands G to P*, available from the web site.
- *Other Commands Q to Z*, available from the web site.

NOTICE

This manual and any examples contained herein are provided as is and are subject to change without notice. Except to the extent prohibited by law, Hewlett-Packard Company makes no express or implied warranty of any kind with regard to this manual and specifically disclaims the implied warranties and conditions of merchantability and fitness for a particular purpose and Hewlett-Packard Company shall not be liable for any errors or for incidental or consequential damage in connection with the furnishing, performance or use of this manual and the examples herein.

Chapter 1

Entering commands

There are a number of ways you can enter a command:

- by pressing the key or keys for the command
- by selecting the command from a subject-specific menu
- by selecting the command from the command catalog
- by typing the command on the command line.

Keyboard entry

The most commonly needed commands can be entered directly from the keyboard by pressing one or two keys. For example, to enter the SIN command, press the SIN key; to enter the LOG command, press ↱ log. See chapter 1 of the *User's Guide* for an explanation of each key on the HP 49G keyboard.

Subject-specific menus

The HP 49G provides a number of menus that list just those commands relevant to a specific topic.

For example, all the programming commands are collected together on the programming menu. You access this menu by pressing ↰ PRG.

Other subject-specific menus include a:

- math menu (↰ MTH)
- symbolic solve menu (↰ S.SLV)
- exponential and logarithm menu (↰ EXP&LN)
- trigonometry menu (↱ TRIG)
- calculus menu (↰ CALC)
- algebra menu (↱ ALG)
- matrices menu (↰ MATRICES)
- statistics menu (↱ STAT)
- convert menu (↰ CONVERT)

- units menu (↱ UNITS)
- arithmetic menu (↰ ARITH)
- complex number menu (↱ CMPLX)
- base menu (↱ BASE).

These subject-specific menus have keys allocated to them on the keyboard.

There are many other subject-specific menus. For example, the math menu and the various Computer Algebra System sub-menus can be selected from the Applications menu:

1. Press APPS.

   The Applications menu is displayed.

2. Press ▲.

   The last page of the Applications menu is displayed.

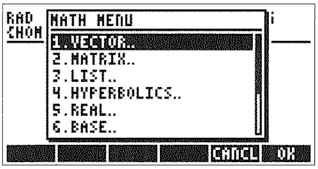
3. Highlight the menu you want—MATH or CAS—and press OK or ENTER.

   Both menus are made up of several sub-menus, each listing the commands of relevance to a particular area of mathematics.

The symbolic menu (displayed by pressing SYMB) provides a series of sub-menus each listing the more commonly used commands of relevance to a particular area of mathematics: algebra, arithmetic, calculus and so on.

Sub-menus

Some entries on a subject-specific menu are shortcuts to sub-menus. For example, pressing ↰ MTH displays the MATH menu. The first item on the MATH menu is VECTOR. The two periods after the word "VECTOR" indicates that there is a further menu. This further menu lists commands of relevance to vector mathematics (such as ABS, DOT, CROSS, etc).



Entering a command from a sub-menu

1. With the main menu displayed, highlight the name of the sub-menu.

   You can do this by:

   - pressing ▼ until the sub-menu name is highlighted or
   - pressing the number key that corresponds to the number of the sub-menu. For example, to highlight the REAL sub-menu in the above example you could press 5.)

2. Press OK or ENTER.

   The sub-menu is displayed.

3. Highlight the name of the command you want to enter.

   You can do this by:

   - pressing ▼ until the command name is highlighted or
   - pressing the number key that corresponds to the number of the command.

4. Press OK or ENTER.

If you are working in algebraic mode, the command you selected appears on the command line ready for you to specify arguments.

If you are working in RPN mode, the result of the command appears on the stack.

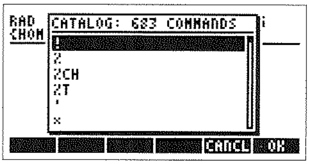
Entering a command from a function key menu

A subject-specific menu can take the form of a choose list (as described in the previous section) or a function key menu. A function key menu is one that appears across the bottom of the screen, with items chosen from the menu by pressing the corresponding function key: F1 to F6.) This is explained in chapter 2 of the *User's Guide*.

Command catalog

The HP 49G provides a single choose list that contains all its commands. This is called the *command catalog.*

Displaying the command catalog



1. Press (CAT).

   A choose list is displayed. This is the command catalog.

2. Locate the command that you want to enter. (See the next section.)

3. Press OK or (ENTER).

   If you are working in algebraic mode, the command you selected appears on the command line ready for you to specify arguments.

   If you are working in RPN mode, the result of the command appears on the stack.

Locating a command in the command catalog

There are two ways to locate a command in the command catalog:

- by scrolling
- by searching on the first character or first few characters of the command.

Scrolling through the catalog

To scroll through the command catalog, press the ▼ or ▲ keys. Keep doing so until the command you want is highlighted.

You can use the shift keys in conjunction with the arrow keys to scroll more quickly through the catalog. For example:

↱⤓ displays the last command in the catalog

↱⤒ displays the first command

↱⇟ displays the next page of commands

↱⇞ displays the previous page of commands.

The number of commands displayed per page is determined by flag –90. When set—which is the default—the commands are displayed in the mini-font. This gives you six commands per page. If flag –90 is cleared, the commands are displayed in your currently selected system font. If, for example, you have chosen System 8 as your current system font, you will see only four commands per page.

Searching for a command in the command catalog

The majority of commands can be selected from the command catalog by searching on the first character of the command's name.

1. Press the keys for the first character—or first few characters—of the command's name.

   For example, to find the HALFTAN command, type ALPHA H or ALPHA ALPHA HA.

   The first command with a name starting with the letter or letters you typed will be highlighted.

2. If the command you want is not highlighted, press ▼ until the command is highlighted.

Note that the first character of a command's name may be:

- an upper-case character (for example, SIN)
- a lower-case character (for example, rref), or
- a symbol (for example →TAG).

The type of the first character determines the best way to search for the command.

Upper case: if the command you want to find begins with an upper-case letter, you must press ALPHA before pressing the key for that letter. If you intend to search by typing the first few characters of the command name, you will need either to:

- hold the ALPHA key down while you key all the characters, or
- lock the alpha keyboard before you type the characters.

Lower case: if the command you want to find begins with a lower-case letter, you must press ALPHA ↰ before pressing the key for that letter. For example, pressing ALPHA ↰ R displays the rpm command.

Non-alphabetic characters: if the command you want to find begins with a non-alphabetic character, press the keys for that character. For example, to search for →TAG, type ↱ →. The first command with a name that starts with → is displayed. (In this example, you need to press ▼ until →TAG is highlighted.)

While the alpha keyboard is active, you can also search for a command that begins with a special character by typing that character and then one or more of the next alphabetical characters.

The keystrokes for non-alphabetic characters can be found in the Characters catalog. For example, the keystrokes for Σ are ↱ and S. Therefore, to search for the Σ+ command in the command catalog, you display the catalog and press ↱ S.

The Characters catalog—accessed by pressing ↱ CHARS—is explained in chapter 2 of the *User's Guide.*

Note that you can import additional commands from third-party command libraries. When a library is attached, the commands in the library will appear in the command catalog. If any such command begins with a character that is not available directly from the HP 49G keyboard, you will not be able to search for the command.

Typing commands

In addition to selecting a command from the command catalog or from a menu, you can also enter a command by typing its name on the command line.

For example, to factorize an expression while in algebraic mode, you could:

1. type ALPHA ALPHA FACTOR ↰ ()
2. enter the expression
3. press ENTER.

To factorize an expression while in RPN mode, you could:

1. enter the expression onto level 1 of the stack
2. type ALPHA ALPHA FACTOR
3. press ENTER.

Every command found in the command catalog can be entered directly from the keyboard.

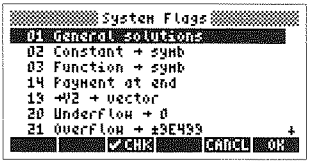
Some command names contain special characters (for example, → and Σ). The Characters catalog—accessed by pressing ↱ CHARS and explained in chapter 2 of the *User's Guide*—shows the keystrokes needed to enter special characters.

Chapter 2

System flags

System flags provide you with some control over how the HP 49G behaves and displays information. For example, by setting flag –60, you can lock the alpha keyboard by pressing (ALPHA) once rather than twice. Clearing flag –60 returns the mode to its default setting (where (ALPHA) must be pressed twice to lock the alpha keyboard).

Displaying system flags



```
System Flags
01 General solutions
02 Constant → symb
03 Function → symb
14 Payment at end
19 →V2 → vector
20 Underflow → 0
21 Overflow → ±9E499
```

1. Press (MODE) to display the Calculator Modes input form.
2. Press FLAGS.

   The System Flags list is displayed. A tick to the left of a flag number indicates that the flag is set. If the space to the left of a flag number is empty, the flag is clear. From this screen you can set or clear any flag in the list.

The System Flags list is not a list of *all* the system flags. As a general rule, flags that are likely to be changed often can be set or cleared on various input forms, where they are more readily accessible. Flags that will rarely be changed are listed in the System Flags list.

For example, flag –95 determines whether the calculator is in algebraic or RPN mode. As this is a flag that many users are likely to want to change from time to time, it is accessible from an input form: the Calculator Modes input form (described in chapter 2 of the *User's Guide*). Because it is accessible from an input form, flag –95 is not listed in the System Flags list. (There are some exceptions to this general rule, with some flags accessible both from an input form and from the System Flags list.)

All the system flags—not just those on the System Flags list—are listed and described in the *HP 49G Pocket Guide*.

Setting and clearing flags

Method 1

Use this method only if the flag you want to change is listed in the System Flags list.

1. With the System Flags list displayed, highlight the flag that you want to set or clear.

   You can do this by either:

   - pressing the ▼ or ▲ keys until the flag you want is highlighted or
   - typing the first digit of the number of the flag.

   See the previous section for instructions on displaying the flags.

2. Press CHK.

   If the flag was set before, it is cleared; if it was clear before, it is now set.

3. Press OK or ENTER to return to the Calculator Modes screen and OK or ENTER again to return to your default screen.

Method 2

If you know the number of the system flag you want to change, you can change the flag from the command line. The syntax you use depends on the operating mode you have chosen: algebraic or RPN.

In algebraic mode, the syntax is:

command(flag)

In RPN mode, enter the flag on the first level of the stack and then enter the appropriate command. The command for setting a flag is SF and the command for clearing a flag is CF. (The flag commands are described on page 2-4.)

For example, to set flag –40 in algebraic mode, you enter SF(–40) and press ENTER. In RPN mode, you enter –40 onto level 1 of the stack, enter SF onto the command line and press ENTER.

Similarly, to clear flag –40 in algebraic mode, you enter CF(–40) and press ENTER. In RPN mode, you enter –40 onto level 1 of the stack, enter CF onto the command line and press ENTER.

You can use this method to set and clear all flags, not just those listed on the System Flags list.

Setting or clearing several flags at once

In algebraic mode, the syntax is:

command({*list*})

In RPN mode, enter a list of the flags you want to set or clear on the first level of the stack and then enter the appropriate command.

For example, to set flags –19 and–40 in algebraic mode, you enter SF({–19,–40}) and press ENTER. In RPN mode, you enter {–19 –40} onto level 1 of the stack, enter SF onto the command line and press ENTER. (Note that in algebraic mode, you must enter a comma between the elements in a list. In RPN mode, you can enter a comma or a space.)

Similarly, to clear flags –19 and–40 in algebraic mode, you enter CF({–19,–40}) and press ENTER. In RPN mode, you enter {–19 –40} onto level 1 of the stack, enter CF onto the command line and press ENTER.

Method 3

The flags that you are likely to need to change can be changed from various input forms (as explained on page 2-1). To change one of these flags, display the appropriate input form, place the cursor in the flag field and press CHK. If the flag was set before, it is cleared; if it was clear before, it is now set. (A tick mark in the field indicates that the flag is set.)

Examples of flags that can be set from input forms include operating mode, number format, angle measure, coordinate system, beep, key click, keep last stack, textbook mode, and clock type. These, and numerous other flags, are accessible by pressing MODE.

Editing input forms is described in detail in chapter 2 of the *User's Guide.*

Flag commands

The flag commands are listed and explained in the table below. The flag commands enable you to set and clear flags as well as having a value returned that indicates the status of a specified flag: 1 if the flag is set, 0 if the flag is clear.

Command	Description
SF	Sets the specified flag.
CF	Clears the specified flag.
FS?	Returns true (1) if the specified flag is set and false (0) if the flag is clear.
FC?	Returns true (1) if the specified flag is clear and false (0) if the flag is set.
FS?C	Tests the specified flag, returns true (1) if it is set or false (0) if it is clear, and then clears it.
FC?C	Tests the specified flag, returns true (1) if it is clear or false (0) if it is set, and then clears it.

Table 2-1 Flag commands

User flags

You can also apply the commands listed in the table above to user flags. (User flags are mainly used in programming.)

User flags are positive numbers, while system flags are negative numbers. (In the System Flags list—discussed on page 2-1—the system flags are shown as positive numbers. However, when you are setting, clearing, and testing system flags, you must specify the flag as a negative number.)

You have access to 128 user flags. You can set, clear, and test them the same way as with system flags. For example, to clear flag 25 that your program has set, you issue the command CF(25).

Chapter 3

Command line operations

The command line is where you enter and edit commands and objects. The HP 49G provides numerous tools to assist you when working on the command line. Some of these tools have their own key; others can be selected from the command line editor's Tool menu.

Activating the command line

The way you activate the command line depends on whether you intend to create a new object or edit an existing object.

To create a new object, just start typing. The command line is activated as soon as you start. (While you can create most objects from the command line, many objects are best created using specific applications. For example, it will usually be easier to create an expression using Equation Writer, or a matrix using Matrix Writer.)

To edit an existing object on the command line, select the object from history or from the stack and:

- in algebraic mode, press ENTER
- in RPN mode, press EDIT.

Note that pressing EDIT in algebraic mode activates whatever application is best suited to editing the type of object you selected, not the command line editor. For example, if you selected an expression, pressing EDIT activates Equation Writer. To edit the object on the command line, you must press ENTER.

Positioning the cursor

When editing the object on the command line, you will almost always need to reposition the cursor.

Main methods: single-line command line

Press ◀ or ▶ to move the cursor left or right respectively.

To go directly to the last character on the command line, press ↰ ⫼▶ or ↱ ▶|.

To go directly to the first character on the command line, press ↰ ◀⫼ or ↱ |◀.

Main methods: multi-line command line

In multiline entries—for example, blocks of programming code—press ▲ or ▼ to move the cursor to the previous or next line respectively.

To go directly to the end of the last line, press ↱ ⤓.

To go directly to position x in the last line, where x is the number of characters the cursor is from the start of the current line, press ↰ ⩡.

To go directly to the start of the first line, press ↱ ⤒.

To go directly to position x in the first line, where x is the number of characters the cursor is from the start of the current line, press ↰ ⩠.

Helpful commands and sub-menus

Like all other HP 49G applications, the command line editor has its own Tool menu. This menu is displayed by pressing TOOL while the command line is active (that is, while the cursor is blinking). You then select a command by pressing the corresponding function key.

The commands that provide ways of positioning the cursor are:

←SKIP

Moves the cursor to the beginning of the current word (that is, to the beginning of the word in which the cursor is currently placed). If pressed again, the cursor moves to the beginning of the previous word if there is one.

SKIP→

Moves the cursor to the beginning of the next word (or to the end of the word if there is no next word).

→BEG

Moves the cursor to the beginning of the selection or, if nothing is selected, to the beginning of the command line. (In multiline entries, the beginning of the command line is the beginning of the first line of the entry.)

→END

Moves the cursor to the end of the selection or, if nothing is selected, to the beginning of the command line.

GOTO

The GOTO menu provides three useful commands:

Go to line: this command sends the cursor to the start of the line whose number you specify.

Go to position: this command sends the cursor to the position you specify (being the specified number of characters—including spaces—from the start of the command line).

See "Command line information" on page 3-12 for instructions on how to find the position of a character on the command line.

Labels: this displays a list of labels in the object on the command line. If you highlight a label in the list and press OK or ENTER, the cursor is sent to the start of that label.

A label is any string of characters that begins with an asterisk. Labels are particularly useful in programming code, where they can be used to name discrete blocks of code.

FIND The Find command is on the Search sub-menu. You can use this command to send the cursor to the character or character string you specify. See “Find” on page 3-8 for more information.

Selecting characters

The HP 49G provides a number of commands that work on *selected* text (such as copy and cut). To select characters, you mark the beginning of the selection and the end of the selection.

1. Position the cursor at the beginning of your selection.

   See “Positioning the cursor” on page 3-2 for a description of various ways to position your cursor.

2. Press ↱ BEGIN.
3. Position the cursor at the end of your selection.
4. Press ↱ END.

   Your selection is now highlighted.

Pressing ↱ END without first pressing ↱ BEGIN causes all characters to the left of the cursor to be selected. Therefore, if you want to select all characters to the left of the cursor, press ↱ END at step 2 above and ignore the remaining steps

Copy, cut, and paste

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

   This is explained on page 3-4 (and in chapter 3 of the *User’s Guide* if you are selecting parts of an expression in the Equation Writer).

2. Do one of the following:
   - to copy the characters, press ↱ COPY, or
   - to cut the characters, press ↱ CUT.
3. Position your cursor where you want to paste the characters you have copied or cut.

   See “Positioning the cursor” on page 3-2 for a description of ways to position your cursor.

4. Press ↱ PASTE.

Editing the command line

Deleting characters

The simplest way to delete a character is to position the cursor to the immediate right of the character and press ◄.

Other ways of deleting characters are provided by commands on the Tool menu for the command line editor. The menu is displayed by pressing TOOL when the command line is active. These commands are:

←DEL — Deletes the characters from the cursor to the beginning of the current word (that is, to the beginning of the word in which the cursor is currently placed). If pressed again, the entire previous word is deleted if there is one.

↱←DEL — Deletes all characters from the cursor to the start of the line.

DEL→ — Deletes the characters from the cursor to the beginning of the next word (or, if there is no next word, to the end of the current word).

↱DEL→ — Deletes all characters from the cursor to the end of the line.

DEL L — Deletes all characters on the current line.

REPLACE — There are various replace options on the Search menu. These options enable you to search for the characters you want to delete and replace them with nothing, thereby deleting those characters.

The replace options are discussed on page 3-8.

Inserting characters

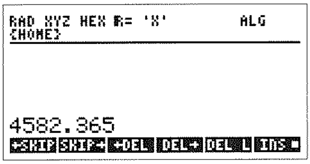
By default, any character you enter on the command line will be inserted between the characters on either side of the cursor. To replace characters rather than insert them, you de-activate insert mode. Each character you enter will then replace—that is, overwrite—the character directly below the cursor.

To de-activate (or activate) insert mode:

1. With the command line active, press TOOL.
2. If the INS command is not displayed, press NXT until it is.

   A small square to the right of the command's label—as in the example on the right—indicates that insert mode is active.

   The shape of the cursor also indicates whether you are in replace mode (cursor is a left-pointing arrow) or insert mode (cursor is a filled rectangle).
3. Press INS to switch insert mode on off or on.



```
RAD XYZ HEX R= 'X'          ALG
{HOME}

4582.365
```

Editing complex objects

In many cases you will find the methods outlined in the previous two sections well-suited to editing command line objects. However, when you need to edit a complex object, you might find it easier to isolate just that part of the object you want to edit. To do this, make sure that the Tool menu for the command line editor is displayed. (Press TOOL if it is not.)

1. Select the part of the object that you want to edit.

   See "Selecting characters" on page 3-4 for instructions.
2. If the EDIT command is not displayed, press NXT until it is.
3. Press EDIT.

   Your selection now appears on the command line by itself. The rest of the object has *not* been deleted; it is just not visible while you edit your selection.
4. Edit the entry on the command line.
5. Press ENTER.

   The object you started with is redisplayed, showing the changes you made at step 4 above.

Evaluating components of the command line

If there is a component of your object that could be evaluated—such as a mathematical expression—you can select the component and evaluate it. The result of the evaluation replaces the component.

To evaluate a component, make sure that the Tool menu for the command line editor is displayed. (Press TOOL if it is not.)

1. Select the component of the object that you want to evaluate.

   See "Selecting characters" on page 3-4 for instructions.

2. If the EXEC command is not displayed, press NXT until it is.
3. Press EXEC.

   The component is evaluated, if possible, and the result replaces the component.

Suspending editing

You can suspend editing the command line and return to it later, performing other operations in the meantime. To do this, make sure that the Tool menu is displayed. (Press TOOL if it is not.)

1. If the HALT command is not displayed, press NXT until it is.
2. Press HALT.

   Your default screen is redisplayed. You can now use the calculator to perform other operations.

   While you have an editing session suspended, the annunciator HLT appears in the status area of your default screen (providing that you have not hidden the status area).

3. Press ↰ CONT to return to the command line you earlier suspended.

   You will notice that the contents of the command line have not been affected by the operations you performed since suspending editing.

While you have one command line editing session suspended, you can suspend another. In fact, you can suspend any number of command line editing sessions.

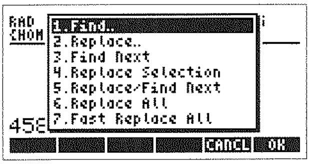
When you have more than one editing session suspended, pressing ↰ CONT returns you to the most recently suspended session. Pressing ↰ CONT again returns you to the second most recently suspended session, and so on.

Find and replace

The HP 49G provides a number of search and replace commands to help you edit multiline objects (such as arrays and programs). These are available from the Search menu (which is a sub-menu of the Tool menu).

1. With an object on the command line, press (TOOL).
2. Press (NXT) to display the second page of the Tool menu.
3. Press SEARCH.

   A choose list appears listing the find and replace options.



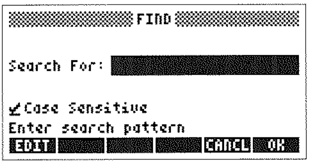
Find

This command searches through the object on the command line and highlights the first instance of a character or character string that matches your search string (that is, the character or character string you specify).

The Find command begins searching from the position of the cursor. Therefore, if your cursor is not at the start of the object, there is a risk that the Find command will not find your search string even though that string is in the object. It is important, then, that you position the cursor at the start of the object if you want to find all occurrences of your search string..

1. If necessary, position the cursor at the beginning of the object on the command line.
2. Display the Search menu, make sure that option 1—Find—is highlighted and press OK or (ENTER).

   The Find input form is displayed.



FIND

Search For:

✓Case Sensitive

Enter search pattern

3. Enter the character or character string that you want to find.

   The characters you type appear near the bottom of the input form.
4. Press OK or (ENTER) to place the characters you have typed into the Search For field.

   The cursor is now positioned in the Case Sensitive field. By default, the Find command performs a case sensitive search; that is, an upper-case character is considered a different character to its lower-case companion.

5. If you do not want a case sensitive search, press CHK.

   The tick in the Case Sensitive field is deleted. Your search will now look for both the upper-case and lower-case versions of the characters you entered into the Search For field.

6. Press OK or ENTER to begin the search.

   If your search string is found, it will be highlighted on the command line; otherwise a message is displayed informing you that your search string cannot be found.

Note that the Find command will look for your search string as both a whole word and as part of a longer word. For example, a search for A*B would be satisfied by the string D/A*B+2. You cannot set the HP 49G to perform only whole-word searches.

Find next

The object you are searching for might contain more than one instance of your search string. To find the next instance:

1. Display the Search menu.
2. Type 3 (or press ▼ until FIND NEXT is highlighted).
3. Press OK or ENTER.

   If your search string is found, it is highlighted on the command line; otherwise a message is displayed informing you that no further instances of your search string can be found.

Note that the HP 49G does not continue searching from the beginning of the object once it has reached the end.

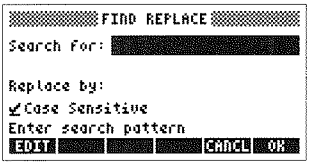
Replace

The Replace command searches through the object on the command line and highlights the first instance of a character or character string that matches your search string (that is, the character or character string you specify). You can then replace that string with another string.

The Replace command begins searching from the position of the cursor. Therefore, if your cursor is not at the start of the object, there is a risk that the Replace command will not find your search string even though that string is in the object. It is important, then, that you position the cursor at the start of the object if you want to find all occurrences of your search string.

1. If necessary, position the cursor at the beginning of the object on the command line.
2. Display the Search menu.
3. Type 2 (or press ▼ until REPLACE.. is highlighted).
4. Press OK or ENTER.

   The Find Replace input form is displayed.

   FIND REPLACE

   Search For:

   Replace by:

   ✓Case Sensitive

   Enter search pattern

5. Enter the character or character string that you want to replace.

   The characters you type appear near the bottom of the input form.
6. Press OK or ENTER to place the characters you have typed into the Search For field.
7. Enter the character or character string that you want to substitute for the search string.

   The characters you type appear near the bottom of the input form.
8. Press OK or ENTER to place the characters you have typed into the Replace By field.

   The cursor is now positioned in the Case Sensitive field. By default, the Replace command performs a case sensitive search; that is, an upper-case character is considered a different character to its lower-case companion.
9. If you do not want a case sensitive search, press CHK.

   The tick in the Case Sensitive field is deleted. Your search will now look for both the upper-case and lower-case versions of the characters you entered into the Search For field.

10. Press OK or ENTER to begin the search.

    If your search string is found, it will be highlighted on the command line; otherwise a message is displayed informing you that your search string cannot be found.

11. If you don't want to replace the current selection but want to continue looking for your search string, select FIND NEXT from the Search menu and repeat this procedure from step 10.

    If you want to replace the current selection with the specified substitute term, you have four options to choose from:

    - replace the current selection (option 4 on the Search menu)
    - replace the current selection and find the next instance of the search string (option 5 on the Search menu)
    - replace the current selection and all other instances of the search string, moving the cursor from instance to instance in the process (option 6 on the Search menu)
    - replace the current selection and all other instances of the search string without moving the cursor from instance to instance (option 7 on the Search menu). At the end of this replace operation, the cursor is positioned at the place of the last change.

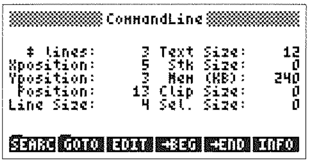
    Option 7—Fast replace—is significantly faster if the object your are searching for extends over many lines (as in the case of a large program).

12. Display the Search menu and choose the replace option you want.

13. Press OK or ENTER to begin the replace operation.

Command line information

The command line editor Tool menu also provides general information about the object on the command line and the position of the cursor. To see this information, press INFO. A screen entitled "Command Line" appears.



```
         CommandLine
  # Lines:      3 Text Size:    12
Xposition:      5  Stk Size:     0
Yposition:      3  Mem (KB):   240
 Position:     13 Clip Size:     0
Line Size:      4 Sel. Size:     0
```

The fields on this screen are:

# Lines	The number of lines over which the object extends.
X position	The number of characters from the start of the line to the position of the cursor.
Y position	The number of the line in which the cursor is placed.
Position	The number of the characters from the start of the object to the position of the cursor.
Line Size	The number of characters in the line on which the cursor is placed.
Text Size	The number of characters in the object.
Stk Size	The number of objects in history (or on the stack).
Mem (KB)	The amount of free memory.
Clip Size	The number of characters on the clipboard (that is, the number of characters you last copied or cut).
Sel. Size	The number of characters in the current selection.

Styles

You can set the style of a command line entry to bold, italic, underlined, or inverted, or to any combination of these characteristics. You can also choose a different font for the entry. To do this, make sure that the command line is active and the Tool menu for the command line editor is displayed. (Press (TOOL) if it is not.)

1. If the STYLES command is not displayed, press (NXT) until it is.
2. Press the function key that corresponds to the style you want the command line to have:
   - (F1) for BOLD
   - (F2) for ITALIC
   - (F3) for UNDERLINED
   - (F4) for INVERTED
   - (F5) to display the fonts list from which you can choose a different font.
3. Press (F6) for EDIT to return to the Tool menu for the command line editor.

Note that styles are retained when the object appears on history or on the stack only if that object is a string (that is, enclosed within inverted commas).

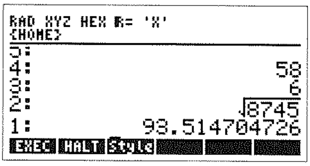
Chapter 4

The Stack

The HP 49G keeps a record of the objects you enter and the results of your operations. In algebraic mode this record is called *history*; in RPN mode it is called the *stack*.

Using the stack

Entries on the stack are numbered (as in the example at the right). An entry on the stack is referred to as being on a particular *level*. The level is the number of the line on which the entry appears. In the example at the right, 58 is on level 4, 6 is on level 3, $\sqrt{8745}$ is on level 2, and so on.



```
RAD XYZ HEX R= 'X'
{HOME}
5:
4:                 58
3:                  6
2:              √8745
1:        93.514704726
```

If you are working in RPN mode, you will use the stack to perform calculations. In doing so, you need to be aware of differences between the way calculations are performed and displayed in RPN mode and in algebraic mode. In RPN mode:

- A command that requires arguments—that is, a command that needs objects to act on—takes its arguments from the stack. Therefore, the arguments must be present before you execute the command: one argument per level and in the correct order. (There is one exception: when the command takes only *one* argument, you can execute the command with the argument on the command line, not on the stack.)
- A command's arguments are removed from the stack when the command is executed. The arguments are replaced by the result of the calculation.

In contrast, in algebraic mode you specify your arguments *after* you enter the command, and the command and its arguments are retained in history together with the result of the calculation.

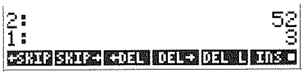
For example, to find the cube of 52, you need to specify two arguments: the number (52) and the index (3). In algebraic mode, you enter:

52 y^x 3 ENTER

In RPN mode, you enter:

52 ENTER 3 ENTER y^x

In other words, in RPN mode 52 and 3 are entered onto the stack before the command is entered: 52 must be on level 2 and 3 on level 1 before the command is executed.



```
2:                52
1:                 3
```

Strictly speaking, the last (or only) argument does not need to be on the stack before you execute a command in RPN mode. You can execute a command with the last (or only) argument still on the command line. Therefore, the second ENTER in the example immediately above can be omitted. However, any argument still on the command line when a command is executed will not be displayed on the stack if you undo the command (which you can do by pressing ↱ UNDO). Therefore, if you think you may need to undo a command and be able to see all the arguments, you should put *all* the arguments onto the stack before executing the command.

Example stack calculations

Using a one-argument command

1. If the argument is not already on level 1 of the stack, enter the argument onto the command line (and, optionally, onto the stack). If the argument is already on level 1 of the stack, go straight to step 2.
2. Execute the command.

Example: To calculate $\frac{1}{\sin 30}$

1. Enter 30 and press ENTER.
2. Press SIN.

   The result of sin 30 is now on level 1 of the stack. This result can be used as the argument of a further command without the result needing to be manually entered.
3. Press 1/x.

Note that if you get a symbolic answer when you wanted a numerical answer, press ↱ →NUM. The symbolic answer is evaluated.

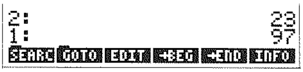
Using a multi-argument command

Method 1

1. Enter the arguments, pressing ENTER after each one.
2. Execute the command.

Example: To calculate 23 × 97

1. Enter 23 and press ENTER.
2. Enter 97 and press ENTER.
   23 is now on level 2 of the stack and 97 is on level 1.



```
2:                        23
1:                        97
```

3. Press ×.

In this example, the order in which you enter the arguments does not affect the answer. However, this is not always the case with two-argument commands. In the cube example on page 4-2, the result of entering the 3 before the 52 is the 3 raised to the power of 52, a very different result to 52 raised to the power of 3. Other examples where the order you enter the arguments is important include subtraction, division, and the percentage commands (%, %CH, and %T).

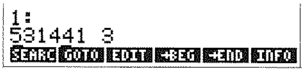
Method 2

In method 1 above, each argument is entered onto its own level of the stack before the command is executed. Another way is to enter all the elements onto the command line separating each with a space. Either:

- press ENTER to place the arguments onto the stack and then execute your command or
- execute your command with the arguments still on the command line.

Example: To calculate $\sqrt[3]{531441}$

1. Enter 531441 SPC 3
2. Press ENTER.



```
1:
531441 3
```

3. Press ↱ $\sqrt[x]{y}$.

Step 2 can be omitted if you will not want to undo the command and see the arguments. Pressing ↱ UNDO without having first placed the arguments on the stack deletes all record of the command: the result and the arguments. On the other hand, if you place the arguments on the stack before executing the command, pressing UNDO deletes the result but redisplays the arguments.

Multi-command calculations

Because the result of a calculation is retained on the stack, you can easily perform complex calculations by accumulating the results of sub-calculations on the stack and then treating these results as the arguments in a further calculation.

Example: To calculate $13^2 - (17 \times 19) + \frac{3}{7}$

1. Enter 13↰x^2.

   The result—169—appears on level 1 of the stack.
2. Enter 17 and press ENTER.
3. Enter 19 and press ENTER.
4. Press ×.

   The product of 17 and 19—323—appears on level 1, and the previous result—169—is now level 2.
5. Press −.

   The two previous results—169—and 323—are now treated as the arguments in a further operation. This operation replaces the arguments with the result of the operation, that is, the difference between the first result and the second.

Chapter 5

Matrices and linear algebra

The HP 49G has extensive capabilities for entering and manipulating arrays. An array object can be a vector or matrix.

Many of the matrix operations described in this chapter also apply to vectors. Wherever this is the case, the more general term *array* is used instead of *matrix*.

By default, Matrix Writer will interpret a one-row array as a vector rather than as a matrix. If you want a one-row array interpreted as an array, press VEC first.

For instructions on how to create an array—and how to open an array in Matrix Writer—see chapter 8 of the *User's Guide*.

Matrix Writer operations

The following summarizes the operations that are available in Matrix Writer.

To move around an array

Method 1: using arrow keys

Press the arrow keys to move the cursor from cell to cell.

You can press ↱ before an arrow key to move the cursor directly to an outermost row or column:

- ↱ ▶ moves the cursor to the last column
- ↱ ▲ moves the cursor to the first row
- ↱ ◀ moves the cursor to the first column
- ↱ ▼ moves the cursor to the last row.

Method 2: the GOTO command

1. Press GOTO.

   The Matrix Writer input form is displayed.

   Note that the GOTO command is on the second page of the Matrix Writer menu, so you may need to press NXT to display it.
2. Enter the row number of the cell you want to go to.
3. Press ENTER.
4. Enter the column number of the cell you want to go to.
5. Press ENTER.
6. Press OK or ENTER.

   The matrix is redisplayed and the cursor is now in the cell whose row–column coordinates you specified.

To edit an array

1. Move the cursor to a cell to be edited.

   See the previous section for instructions on cursor movement.
2. Press EDIT.

   The contents of the cell are copied to the command line.

If you want to completely change the contents of the cell, you can do so without first pressing EDIT. Just start typing the new contents. What you type appears on the command line.

3. Make your changes.
4. Press ENTER to move your changes from the command line to the cell.
5. Repeat from step 1 if you want to change other cells.
6. Press ENTER to close Matrix Writer and place the changed array on the command line (or press CANCEL to close Matrix Writer and discard your changes).

   If you discarded your changes, ignore the next step.
7. Press ENTER again to save your changes.

To make the cells narrower or wider

- Press ←WID to make the cells narrower.

  More columns are displayed.
- Press WID→ to make the cells wider.

  Fewer columns are displayed.

Note that these commands modify the width of *all* columns, not just the column with the highlighted cell.

To control how the cursor moves after an entry

By default, the cursor moves to the adjacent cell in the next *column* after you place an object in a cell. You can change this setting in one of two ways:

- To make the cursor move to the adjacent cell in the next *row* after entry, press GO↓.

  A small box (■) appears beside the command on the menu.
- To prevent the cursor from advancing at all after entry, press GO→ and GO↓ until neither command shows a box (■) beside it on the menu.

The setting you choose remains the mode of cursor movement until you change the setting again.

To return to the default setting, press GO→ until a box (■) appears beside it on the menu.

If you want to change the way the cursor moves through an array, make your change *before* entering objects into the array. Once you start entering objects, the way the cursor moves cannot be changed.

Manipulating columns and rows

To insert a column

1. Move the cursor to the column where you want the new column to appear.
2. Press +COL. A column of zeros is inserted.

   Note that the +COL command is on the second page of the menu, so you may need to press NXT to display it.

You can also add a column to an array without using Matrix Writer. See "To insert one or more new columns into an array" on page 5-11.

To add a column to the right of the last column of data

1. Move the cursor to the right of the last column of data.

   Pressing ↱ ▶ will move the cursor directly to the last column of data. You then press ▶ to move to the next column.
2. Enter an object.
3. Press ENTER to move your object to the highlighted cell.

   The rest of the column fills with zeros and your array now includes this new column.

To delete a column

1. Move the cursor to the column you want to delete.
2. Press –COL.

   Note that the –COL command is on the second page of the menu, so you may need to press NXT to display it.

To insert a row

1. Move the cursor to the row where you want the new row to appear.
2. Press +ROW. A row of zeros is inserted.

   Note that the +ROW command is on the second page of the menu, so you may need to press NXT to display it.

You can also add a row to an array without using Matrix Writer. See "To insert one or more new rows into a matrix" on page 5-11.

To add a row below the bottom row of data

1. Move the cursor to the row below the last row of data.

   Pressing ↱ ▼ will move the cursor directly to the last row of data. You then press ▼ to move to the next row.

2. Enter an object.
3. Press ENTER to move your object to the highlighted cell.

   The rest of the row fills with zeros and your array now includes this new row.

To delete a row

1. Move the cursor to the row you want to delete.
2. Press –ROW.

To delete the contents of a selection of cells

1. Move the cursor to the first cell in the group of cells you want to delete.
2. Press ↱ BEGIN to select BEGIN.
3. Move the cursor to the last cell in the group of cells you want to delete.
4. Press ↱ END to select END.

   The cells between your beginning and end cell are now highlighted.

5. Press DEL to delete the contents of the highlighted cells.

   The DEL command does *not* leave the selected cells empty. Instead, the contents of a cell are replaced with a zero.

Summary of Matrix Writer operations

Key	Description
EDIT	Places the contents of the current cell on the command line for editing.
VEC	For one-row arrays, toggles between vector entry and matrix entry. If this command is selected, one-row arrays are entered onto the command line as vectors (example: [1 2 3]); if it's not selected, one-row arrays are entered as matrices (example: [[1 2 3]]).
←WID	Decreases the width of all cells.
WID→	Increases the width of all cells.
GO→	Sets left-to-right entry mode. The cursor moves to the next *column* after data entry.
GO↓	Sets top-to-bottom entry mode. The cursor moves to the next *row* after data entry.
+ROW	Inserts a row of zeros at the current cursor position.
–ROW	Deletes the current row.
+COL	Inserts a column of zeros at the current cursor position.
–COL	Deletes the current column.
→STK	Copies the current cell to history (level 1 of the stack).
GOTO	Move the cursor to a specified cell.
DEL	Replace the contents of selected cells with zeros.

Advanced matrix operations

Creating special matrices

To create an array filled with a given constant

1. Select the Constant Array command.

   ↰ MATRICES CREATE CON
2. For the first argument of the command, enter either:
   - a list containing the dimensions of the desired constant array: { *rows*, *columns* } or
   - an existing array.
3. For the second argument, enter the constant that you want in the array.
4. Press ENTER.

   The result is an array of the dimensions you entered (or of the dimensions of the specified array) filled with the specified constant.

To create an identity matrix

1. Select the Identity Matrix command.

   ↰ MATRICES CREATE IDN
2. Enter either:
   - a number representing the number of rows and columns you want in the identity matrix or
   - an existing array.
3. Press ENTER.

   The result is an identity matrix of the specified dimensions (that is, a square matrix with zero for all elements except the diagonal elements, which are all 1).

To create an array filled with random integers

1. Select the Random Matrix command.

   ↰ MATRICES CREATE RANM

2. Enter either:
   - a list containing the dimensions of the desired random matrix: { *rows*, *columns* } or
   - an existing array.

3. Press ENTER.

   The result is a random array of the specified dimensions (or of the dimensions of the specified array). The elements are integers within the range –9 to 9.

Assembling matrices

To assemble a matrix by rows from a series of vectors

1. Select the Rows-to-Matrix command.

   ↰ MATRICES CREATE ROW ROW→

2. Enter each vector in the order you want them to appear in the matrix. Enter the row 1 vector first, then the row 2 vector, and so on, separating each vector with a comma.
3. Enter the number of rows in the desired matrix.
4. Press ENTER.

   The result is a matrix composed of the vectors you entered.

To assemble a matrix by columns from a series of vectors

1. Select the Columns-to-Matrix command.

   ↰ MATRICES CREATE COLUMN COL→

2. Enter each vector in the order you want them to appear in the matrix. Enter the column 1 vector first, then the column 2 vector, and so on, separating each vector with a comma.
3. Enter the number of columns in the desired matrix.
4. Press ENTER.

   The result is a matrix composed of the vectors you entered.

To assemble a matrix with a particular diagonal from a vector

1. Select the Vector-to-Matrix Diagonal command.

   ↰ MATRICES CREATE DIAG→

2. Enter the vector containing the diagonal elements.
3. Enter either:
   - a list containing the dimensions of the desired matrix: {*rows columns*} or
   - a real number representing the number of rows and columns in the desired square matrix.
4. Press ENTER.

   The result is a matrix of the desired dimensions using the elements of the vector as the diagonal elements of the matrix. If the vector contains more diagonal elements than are needed to create the matrix, the extra elements are discarded. If the vector does not contain enough elements to complete the matrix, the undefined diagonal elements are set to zero.

To assemble a matrix from a sequence of elements

1. Select the Stack-to-Array command.

   ↰ PRG TYPE →ARRY

2. Enter the elements in *row-major order.*

   Row-major order begins with the first element (the element in row 1 and column 1). The next element is the next one in the *row*. If there are no more elements in the row, the next element is the first element in the next row, and so on.

3. Enter a list containing the dimensions of the desired matrix: {*rows, columns*}.
4. Press ENTER to assemble the matrix.

For example, →ARRY(1, 2, 3, 4, {2, 2}) results in $\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix}$

Disassembling matrices

To disassemble a matrix into its elements

1. Select the Object-to-Stack command.

   ↰ PRG TYPE OBJ→

2. Enter or select the matrix you want to disassemble.
3. Press ENTER.

   The matrix is disassembled in row-major order. A list indicating the dimensions of the matrix is also returned.

To disassemble a matrix into row vectors

1. Select the Matrix-to-Rows command.

   ↰ MATRICES CREATE ROW →ROW

2. Enter or select the matrix you want to disassemble.
3. Press ENTER.

   The result is a list of row vectors (first row to last), and the number of rows in the matrix.

To disassemble a matrix into column vectors

1. Select the Matrix-to-Columns command.

   ↰ MATRICES CREATE COLUMN →COL

2. Enter or select the matrix you want to disassemble.
3. Press ENTER.

   The result is a list of column vectors (first column to last), and the number of columns in the matrix.

To extract the vector of diagonals from a matrix

1. Select the Matrix-Diagonal-to-Array command.

   ↰ MATRICES CREATE →DIAG

2. Enter or select the matrix.
3. Press ENTER.

   The result is a vector whose elements were the diagonal elements of the matrix.

Inserting rows and columns

To insert one or more new rows into a matrix

1. Select the Insert Row command.

   ↰ (MATRICES) CREATE ROW ROW+
2. Enter or select the array you want to modify.
3. Enter the vector or matrix that you want to insert.

   An inserted array must have the same number of columns as the array into which it is being inserted.
4. Enter the row number you want the first (or only) inserted row to be.
5. Press (ENTER).

   The rows beneath, and including, the row specified at step 4 are pushed down to make room for the inserted rows.

You can also add rows to an array using Matrix Writer. See "To insert a row" on page 5-4.

To insert one or more new columns into an array

1. Select the Insert Column command.

   ↰ (MATRICES) CREATE COLUMN COL+
2. Enter or select the array you want to modify.
3. Enter the vector or matrix that you want to insert.

   An inserted array must have the same number of rows as the array into which it is being inserted.
4. Enter the column number you want the first (or only) inserted column to be.
5. Press (ENTER).
6. The columns to the right of, and including, the column specified at step 4 are pushed to the right to make room for the inserted columns.

You can also add columns to an array using Matrix Writer. See "To insert a column" on page 5-4.

Extracting rows and columns

To extract a particular row from an array

1. Select the Delete Row command.

   ↰ MATRICES CREATE ROW ROW–
2. Enter or select the array with the row you want to extract.
3. Enter the number of the row you want to extract.
4. Press ENTER.

   The result is the array without the extracted row, and the extracted row as a vector.

To extract a particular column from an array

1. Select the Delete Column command.

   ↰ MATRICES CREATE COLUMN COL–
2. Enter or select the matrix with the column you want to extract.
3. Enter the number of the column you want to extract.
4. Press ENTER.

   The result is the matrix without the extracted column, and the extracted column as a vector.

Swapping rows and columns

To swap two rows in an array

1. Select the Row Swap command.

   ↰ MATRICES CREATE ROW RSWP
2. Enter or select the array with the rows you want to swap.
3. Enter the number of the one of the rows you want to swap.
4. Enter the number of the other row you want to swap.
5. Press ENTER.

   The result is the array with the two specified rows swapped.

To swap two columns in an array

1. Select the Column Swap command.

   ↰ MATRICES CREATE COLUMN CSWP
2. Enter or select the matrix with the columns you want to swap.
3. Enter the number of the one of the columns you want to swap.
4. Enter the number of the other column you want to swap.
5. Press ENTER.

   The result is the array with the two specified columns swapped.

Extracting and replacing elements of matrices

To extract the element at a specified position

1. Select the Get Element command.

   ↰ MATRICES CREATE GET
2. Enter or select the array with the element you want to extract.
3. Enter either:
   - a list containing the row number and column number of the element you want to extract, or
   - the position number—that is, row-major number—of the element you want to extract.
4. Press ENTER.

   The result is the extracted element.

To replace an element at a specified position

1. Select the Put Element command.

   ↰ MATRICES CREATE PUT
2. Enter or select the array with the element you want to replace.
3. Enter either:
   - a list containing the row number and column number of the element you want to replace, or
   - the position number—that is, row-major number—of the element you want to replace.
4. Press ENTER.

   The result is a modified array.

Characterizing matrices

Matrix calculations are often sensitive to special characteristics of the matrices used. The HP 49G has a number of commands that return characteristics of matrices. Note that some commands are only defined for *square* matrices, some for any rectangular matrix.

Commands for characterizing matrices

Keys	Description
↰ (MATRICES) OPERATIONS SIZE	Returns the dimensions of the array (that is, the number of rows and number of columns).
↰ (MATRICES) OPERATIONS ABS	Returns the Frobenius norm of a matrix and the Euclidean length of a vector: the square root of the sums of the squares of the absolute values of the elements.
↰ (MATRICES) OPERATIONS SNRM	Returns the spectral norm of a matrix. The spectral norm of a matrix is equal to the largest singular value of the matrix. Same as ABS for a vector.
↰ (MATRICES) OPERATIONS RNRM	Returns the row norm of a matrix. The row norm of a matrix is the maximum value (over all rows) of the sums of the absolute values of all elements in a row. The row norm of a vector is the maximum absolute value of its elements.
↰ (MATRICES) OPERATIONS CNRM	Returns the column norm of a matrix. The column norm of a matrix is the maximum value (over all columns) of the sums of the absolute values of all elements in a column. The column norm of a vector is the sum of the absolute values of its elements.

Keys (Continued)	Description
↰ MATRICES OPERATIONS SRAD	Returns the spectral radius of a square matrix. The spectral radius is the absolute value of the largest eigenvalue of the matrix.
↰ MATRICES OPERATIONS COND	Returns the column-norm condition number of a square matrix. The condition number is defined to be the product of the column norm of a square matrix and the column norm of its inverse.
↰ MATRICES OPERATIONS RANK	Returns an estimate of the rank of a matrix. The rank of a matrix is equal to the number of non-zero singular values of the matrix. If flag –54 is clear (default), RANK treats any computed singular value less than 10^{-14} times the size of the largest computed singular value as zero. If flag –54 is set, RANK counts all non-zero singular values no matter what their size.
↰ MATRICES OPERATIONS DET	Returns the determinant of a square matrix. DET checks flag –54, and will refine its computed value only if –54 is clear (default).
↰ MATRICES OPERATIONS TRACE	Returns the trace of a square matrix. The trace of a matrix is equal to the sum of the diagonal elements and also equal to the sum of the eigenvalues of the matrix.

Transforming matrices

To transpose a matrix

1. Select the appropriate Transpose Matrix command:
   - ↰ MTH MATRIX MAKE TRN (if you want the conjugate transpose of a complex matrix), or
   - ↰ MATRICES OPERATIONS TRAN (if you want transposition without conjugation).
2. Enter or select the array you want to transpose.
3. Press ENTER to transpose the matrix.

   The first row of the original matrix is now the first column, the original second row is now the second column and so on.

To invert a square matrix

1. Press 1/x.
2. Enter or select the array you want to invert.
3. Press ENTER to transpose the matrix.

To change the dimensions of an array

1. Select the Redimension Array command.

   ↰ MATRICES CREATE RDM
2. Enter or select the array that you want to redimension.
3. Enter a list containing the new dimensions of the array: { *row, column* }.
4. Press ENTER.

   Elements from the original array are placed *in row-major order* into the newly-dimensioned array. If there are fewer elements in the new array than in the original one, the excess elements are dropped. If there are more elements in the new array than in the original one, the missing elements are filled with zeros (or (0,0) if the array is complex).

For example, RDM($\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$, {3,4}) results in: $\begin{bmatrix} 1 & 2 & 3 & 4 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$

More matrix arithmetic

Simple matrix arithmetic is covered in chapter 8 of the HP 49G *User's Guide.* This section covers some of the other arithmetic options.

To change the sign of each element in a matrix

1. Press (+/-)(( )).
2. With the cursor between the parentheses, enter or select the matrix.
3. Press (ENTER).

To multiply a matrix and vector

1. Enter or select the matrix.
2. Press (X).
3. Enter or select the vector.

   The number of elements in the vector must equal the number of columns in the matrix.
4. Press (ENTER).

To divide an array by a square matrix

1. Enter or select the array.
2. Press (÷).
3. Enter the square matrix.

   The number of rows in the matrix must equal the number of rows in the array.
4. Press (ENTER).

Transforming complex matrices

To combine two real matrices into a complex matrix

1. Select the Real-to-Complex command.

   ↰ MTH COMPLEX R→C

2. Enter or select the real matrix that will become the real part of the complex matrix.
3. Enter or select the real matrix that will become the imaginary part of the complex matrix.

   This matrix must have the same dimensions as the matrix entered at step 2.

4. Press ENTER.

   The two real matrices are combined to form a complex matrix.

To split a complex matrix into two real matrices

1. Select the Complex-to-Real command.

   ↰ MTH COMPLEX C→R

2. Enter or select the complex matrix that you want to split.
3. Press ENTER.

   The result is two real matrices formed from the complex matrix.

To conjugate each element of a complex matrix

1. Select the Conjugate function.

   ↱ COMPLEX CONJ

2. Enter or select the complex matrix that you want to conjugate.
3. Press ENTER.

To extract the matrix of real parts from a complex matrix

1. Select the Real Part function.

   ↱ COMPLEX RE

2. Enter or select the complex matrix whose real components you want to extract.
3. Press ENTER.

   The result is a matrix comprising just the real components of the complex matrix.

To extract the matrix of imaginary parts from a complex matrix

1. Select the Imaginary Part function.

   ↱ COMPLEX IM

2. Enter or select the complex matrix whose imaginary components you want to extract.

3. Press ENTER.

   The result is a matrix comprising just the imaginary components of the complex matrix.

Linear algebra topics

The use of matrix functions to solve systems of linear equations is covered in chapter 8 of the HP 49G *User's Guide.* This section covers other important linear algebra commands.

Eigenvalues and eigenvectors

A square ($n \times n$) matrix **A** is said to have an *eigenvalue* λ and a corresponding *eigenvector* **x** if $\mathbf{Ax} = \lambda\mathbf{x}$.

Eigenvalues are the roots of the *characteristic equation*—$\det(\mathbf{A} - \lambda\mathbf{I}) = 0$—which is a polynomial of degree n. Thus, **A** has n eigenvalues, although they are not always distinct. Each eigenvalue has a corresponding eigenvector.

The HP 49G allows you to compute either the eigenvalues only (a faster computation) or both the eigenvalues and their corresponding eigenvectors.

To compute the eigenvalues for a square matrix

1. Select the Eigenvalues command.

   ↰ MATRICES EIGENVECTOR EGVL

2. Enter or select the square ($n \times n$) matrix whose eigenvalues you want to calculate.

3. Press ENTER.

   The result is a vector of n eigenvalues.

To compute the eigenvalues and eigenvectors for a square matrix

1. Select the Eigenvalues and Eigenvectors command.

   ↰ MATRICES EIGENVECTOR EGV
2. Enter or select the square ($n \times n$) matrix whose eigenvalues and eigenvectors you want to calculate.
3. Press ENTER.

   The result is an $n \times n$ matrix of eigenvectors and an n-element vector of eigenvalues.

To compute the singular values of a matrix

1. Select the Singular Values command.

   ↰ MATRICES FACTORIZATION SVL
2. Enter or select the matrix.
3. Press ENTER.

   The result is a vector—of length MIN(m,n)—of the singular values of the matrix. The values are returned in non-increasing order.

To decompose or factor a matrix

The HP 49G offers a set of matrix decomposition and factorization tools that you can use either alone or in program routines to solve specialized problems. These tools are explained in the following table:

Keys	Description
↰ MATRICES FACTORIZATION LU	**Crout LU Decomposition.** This procedure is used in the process of solving an exactly-determined system of linear equations, inverting a matrix, and computing the determinant of a square matrix. It factors the square matrix (**A**) into a lower-triangular matrix **L**, an upper-triangular matrix **U** containing ones on its diagonal, and a permutation matrix **P** such that **PA** = **LU**.
↰ MATRICES FACTORIZATION LQ	**LQ Factorization.** This command factors an $m \times n$ matrix **A** into an $m \times n$ lower trapezoidal matrix **L**, an $n \times n$ orthogonal matrix **Q**, and an $m \times m$ permutation matrix **P** such that **PA** = **LQ**.
↰ MATRICES FACTORIZATION QR	**QR Factorization.** This command factors an $m \times n$ matrix **A** into an $m \times m$ orthogonal matrix **Q**, an $m \times n$ upper trapezoidal matrix **R**, and an $n \times n$ permutation matrix **P** such that **AP** = **QR**.
↰ MATRICES FACTORIZATION SCHUR	**Schur Decomposition.** This command factors a square matrix **A** into an orthogonal matrix **Q** and an upper-triangular matrix (or, if **A** is real-valued, the upper quasi-triangular matrix) **U** such that $\mathbf{A} = \mathbf{QUQ}^{\mathbf{T}}$ (where $\mathbf{Q}^{\mathbf{T}}$ is the transpose of matrix **Q**).

Keys (Continued)	Description
↰ MATRICES FACTORIZATION SVD	**Singular Value Decomposition.** This command factors an $m \times n$ matrix **A** into an $m \times m$ orthogonal matrix **U**, an $n \times n$ orthogonal matrix **V**, and a vector **S** of the singular values of **A** such that **A** = **US‘V** (where **S‘** is the $m \times n$ matrix formed by using the elements of **S** as its diagonal elements).

Chapter 6

Unit objects

The Units application contains a catalog of 127 units that you can combine with real numbers to create *unit objects*. It also provides you with tools to manipulate unit objects.

The Units application enables you to:

- convert units—for example, you can convert the unit object 10_ft to 120_in or 3.048_m
- factor units—for example, you can factor 20_W with respect to 1_N and return 20_N*m/s
- calculate with units—for example, you can add 10_ft/s to 10_mph and return 16.82_mph.

Overview of the Units application

The calculator's units are based on the 7 base units of the International System of Units (SI): *m* (meter), *kg* (kilogram), *s* (second), *A* (ampere), *K* (kelvin), *cd* (candela), and *mol* (mole). The HP 49G makes use of two additional base units: *r* (radian) and *sr* (steradian). The remaining 118 units are compound units, that is, units derived from the 9 base units.

For example, in (inch) is defined as .0254m, and Fdy (Faraday) is defined as 96487 A*s. (See appendix B of the *User's Guide* for a complete listing of the built-in units and their SI values.)

Units are grouped into the following categories:

length	area	volume	time
speed	mass	force	energy
power	pressure	temperature	electric current
angle	light	radium	viscosity

These categories can be chosen from the Units menu (displayed by pressing ↱ UNITS).

Unit objects

A unit object has two parts: a *real number* and a *unit expression* (a single unit or multiplicative combination of units). The two parts are linked by the underscore character _. For example, 2_in (2 inches) and 8.303_gal/h (8.303 US gallons per hour) are unit objects. Like other object types, a unit object can be placed on the stack, stored in a variable, and used in algebraic expressions and programs.

When you perform a *unit conversion*, the HP 49G replaces the old unit expression with the new unit expression you specify, and automatically multiplies the number by the appropriate conversion factor.

Operators in unit objects follow this order of precedence:

1. ( )
2. ^
3. * and /

For example, 7_m/s^2 is 7 meters per second squared, and 7_(m/s)^2 is 7 square meters per second squared.

To create a unit object

1. Key in the number part of the unit object.
2. Press ↱ _.
3. Press ↱ UNITS to display the Units menu.
4. Press ▲ or ▼ until the category of the unit is highlighted.
5. Press OK or ENTER.
6. Press ▲ or ▼ until the unit is highlighted.
7. Press OK or ENTER.
8. For compound units, repeat from steps 3 for each additional unit in the unit expression, preceding each with ×, ÷, or y^x as required.

If you know the unit abbreviation, you can type it directly on the command line instead of selecting it from the Units menu. Note that unit names are case-sensitive; for example, Hz (hertz) must be typed with an uppercase H and a lowercase z.

In addition to the method outlined above, you can create a unit object in RPN mode by assembling it from components on the stack.

To assemble a unit object from the stack

1. Place the number part of the unit object on level 2 of the stack and the unit expression on level 1.

   The unit expression must be in the format *n_unit* where *n* is any real number and *unit* is the unit abbreviation (lb, in, m, etc.).

2. Execute the →UNIT command.

   You can execute the →UNIT command by:

   - pressing (↱) (UNITS) TOOLS →UNIT
   - selecting it from the commands catalog ((CAT) →UNIT) or
   - pressing (↱) (→) (ALPHA) (ALPHA) UNIT (ENTER).

For example, entering 12 on level 2, 1_ft on level 1 and executing the →UNIT command returns 12_ft.

Unit prefixes

You can also insert a *unit prefix* in front of a unit to indicate a power of ten. The following table lists the available prefixes. (To key in μ, press (ALPHA) (↱) M.)

Unit Prefixes

Prefix	Name	Exponent	Prefix	Name	Exponent
Y	yotta	+24	d	deci	–1
Z	zetta	+21	c	cent	–2
E	exa	+18	m	milli	–3
P	peta	+15	μ	micro	–6
T	tera	+12	n	nano	–9
G	giga	+9	p	pico	–12
M	mega	+6	f	femto	–15
k or K	kilo	+3	a	atto	–18
h or H	hecto	+2	z	zepto	–21
D	deka	+1	y	yocto	–24

Converting units

The HP 49G provides two commands for converting unit objects from one unit of measurement to another:

- CONVERT
- UBASE.

The CONVERT command—which requires two arguments—can be used to convert one type of unit to *any* other similar unit. The UBASE command—which requires only one argument—is used to convert a unit object to its equivalent SI base unit: feet to metres, knots to metres per second, and so on.

The following examples illustrate how to convert units if you are working in algebraic mode. The steps to follow if you are working in RPN mode are given at the end of the procedure.

To convert from one unit to another

1. Press ↱ UNITS TOOLS CONVERT.
2. Enter the unit object with the units that you want to convert.

   Example: 12_m
3. Press ↱ , .
4. Enter a unit object with the units you want the first unit object converted to.

   Example: 1_ft

   It does not matter what value you give to the number part of this unit object.
5. Press ENTER.

   Example result: 39.3700787402_ft

In RPN mode: follow steps 2, 4, and 1.

If you are converting an angle measurement to units of the same type as your current angle measurement setting, you can omit the unit expression at step 4 above.

For example, if your current angle measurement setting is radians, you can convert 1000_arcmin to radians by entering CONVERT(1000_arcmin, 1) and pressing ENTER.

To convert units to SI base units

The CONVERT command discussed in the previous section can be used to convert to *any* specified units, SI units or otherwise. If you want to convert a unit object to its equivalent SI base units, the UBASE command is quicker, as it requires just one argument.

1. Press ↱ UNITS TOOLS UBASE.
2. Enter the unit object with the units that you want to convert.

   See "To create a unit object" on page 6-2 for instructions on creating unit objects.

   Example: 365_ft
3. Press ENTER.

   Example result: 111.252_m

In RPN mode: follow steps 2 and 1.

You can also use the UBASE command to convert an angle measurement unit to radians.

Calculating with units

The HP 49G lets you execute many arithmetic operations with unit objects, just as you would with real numbers:

- addition and subtraction (dimensionally consistent units only)
- multiplication and division
- inversion
- raising to a power
- percentage calculations (dimensionally consistent units only)
- comparisons of values (dimensionally consistent units only)
- trigonometric operations (planar angular units only)

Where possible, units are automatically converted during the calculation. In such cases, the result is expressed in the units of the last unit object entered.

Note that some operations require certain unit objects to be enclosed in parentheses if you are working in algebraic mode. This is illustrated in the examples given in the next section.

The trigonometric operations SIN, COS, and TAN, operate only on unit objects with *planar angular* units: radians (r), degrees (°), grads (grad), arc-minutes (arcmin), or arc-seconds (arcs).

Temperature units require special attention: see "Working with temperature units" on page 6-8.

Sample unit calculations

The following examples assume that you are working in algebraic mode.

Subtraction. To subtract 39in from 4ft:

1. Enter 4_ft.
2. Press −.
3. Enter 39_in.
4. Press ENTER.

   The answer is 9_in. Note that the answer takes the units of the last entered unit object. If you had subtracted 4ft from 39in, the answer would be in feet.

In RPN mode: follow steps 1, 3, and 2.

Unit multiplication and division. To multiply 50ft by 45ft and divide the result by 3 days:

1. Enter (50_ft).

   Note that when you are in algebraic mode and multiplying or dividing unit objects, or raising a unit object to a power, the unit object must be entered in parentheses. (The one exception is the last unit object you specify.)
2. Press ×.
3. Enter (45_ft).
4. Press ÷.
5. Enter 3_d.

   Note that the last unit object does not need to be entered in parentheses.
6. Press ENTER.

   The answer is 750_ft^2/d.

In RPN mode: follow steps 1, 3, 2, 5, and 4. Parentheses are not required around unit objects in RPN mode.

Powers. To cube 2ft/s.

1. Enter (2_ft/s).

   Note that when you are in algebraic mode and raising a unit object to a power, the unit object must be entered in parentheses.
2. Press (y^x).
3. Enter 3.
4. Press (ENTER).

   The answer is 8_ft^3/s^3

In RPN mode: follow steps 1, 3, and 2. Parentheses are not required around unit objects in RPN mode.

Percentages. To find what percentage of 4.2cm^3 is 1in^3:

1. Press (↰) (MTH) REAL %T.
2. Enter 1_in^3.
3. Press (↱)(,).
4. Enter 4.2_cm^3.
5. Press (ENTER).

   The answer is 25.6299725198.

In RPN mode: follow steps 2, 4, and 1.

Factoring unit expressions

The UFACT command factors one unit within a unit object, returning a unit object whose unit expression consists of the factored unit and the remaining SI base units.

To factor units within a unit expression

1. Press ↱ UNITS TOOLS UFACT
2. Enter the unit object with the original units.
   Example: 74_pdl.
3. Press ↱ , .
4. Enter any number (such as 1) and attach the units you want to factor out.
   Example: 1_kg.
5. Press ENTER.
   Answer: 10.2308666238_kgm/s^2.

In RPN mode: follow steps 2, 4, and 1.

Working with temperature units

The HP 49G lets you work with temperature units in the same way that you work with other units. However, you need to be alert to the difference between temperature *level* and temperature *difference*. For example, a temperature *level* of 0°C means "the freezing point of water", but a temperature *difference* of 0°C means "no change."

When °C or °F represents a temperature *level*, the temperature is a unit with an additive constant: 0°C = 273.15 K, and 0°F = 459.67°R. But when °C or °F represents a temperature *difference*, the temperature is a unit with no additive constant: 1°C = 1 K, and 1°F = 1°R.

Converting temperature units

Conversions between the four temperature scales (K, °C, °F, and °R) involve additive constants as well as multiplicative factors. The additive constants are *included* in a conversion when the temperature units reflect actual temperature *levels*, and are *ignored* when the temperature units reflect temperature *differences*.

- **Pure temperature units (levels).** If both unit expressions consist of a single, unprefixed temperature unit with no exponent, the CONVERT command performs an *absolute* temperature scale conversion which includes the additive constants.
- **Combined temperature units (differences).** If either unit expression includes a prefix, an exponent, or any unit other than a temperature unit, the CONVERT command performs a *relative* temperature unit conversion which ignores the additive constants.

The two examples below illustrate these differences.

Example 1: Convert 25°C to °F.

1. Press ↱ UNITS TOOLS CONVERT.
2. Enter 25_°C.
3. Press ↱ ,.
4. Enter 1_°F.

   It does not matter what value you give to the number part of this unit object.
5. Press ENTER.

   The answer is 77_°F.

In RPN mode: follow steps 2, 4, and 1.

Example 2: Convert 25 °C/min to °F/min.

Note that unlike in the first example, in this example a *relative* temperature conversion will be performed.

1. Press ↱ UNITS TOOLS CONVERT.
2. Enter 25_°C/min.

   Note that °C/min is not available in the units catalog and so must be created. You can select °C from the units catalog, and then press ÷ and type “min” to complete the compound unit. (Note that *min* must be entered in lower case.)
3. Press ↱ , .
4. Enter 1_°F/min.

   It does not matter what value you give to the number part of this unit object.
5. Press ENTER.

   The answer is 45_°F/min.

In RPN mode: follow steps 2, 4, and 1.

Chapter 7

Constants library

The constants library contains a collection of commonly used physical constants and quantities. You can use them in equations and programs.

The following table lists the constants in the order they appear in the library.

Abbrev-iation	Description	Value (SI)
NA	Avogadro's number	6.0221367E23 gmol^{-1}
k	Boltzmann constant	1.380658E–23 J/K
Vm	Molar volume	22.4141 l/gmol
R	Universal gas constant	8.31451 J/(gmol·K)
StdT	Standard temperature	273.15 K
StdP	Standard pressure	101.325 kPa
σ	Stefan-Boltzmann constant	5.67051E–8 W/(m^{2}·K^{4})
c	Speed of light in vacuum	299792458 m/s
ε0	Permittivity of vacuum	8.85418781761E–12 F/m
μ0	Permeability of vacuum	1.25663706144E–6 H/m
g	Acceleration of gravity	9.80665 m/s^{2}
G	Gravitational constant	6.67259E–11 m^{3}/(s^{2}·kg)
h	Planck's constant	6.6260755E–34 J·s
hbar	Dirac's constant	1.05457266E–34 J·s
q	Electron charge	1.60217733E–19 C
me	Electron rest mas	9.1093897E–31 kg
qme	q/me	175881962000 C/kg

Abbrev-iation	Description	Value (SI)
mp	Proton rest mass	1.6726231E–27 kg
mpme	mp/me	1836.152701
α	Fine structure constant	0.00729735308
ϕ	Magnetic flux quantum	2.06783461E–15 Wb
F	Faraday constant	96485.309 C/gmol
R∞	Rydberg constant	10973731.534 m^{-1}
a0	Bohr radius	0.0529177249 nm
μB	Bohr magneton	9.2740154E–24 J/T
μN	Nuclear magneton	5.0507866E–27 J/T
λ0	Photon wavelength (ch/e)	1239.8425 nm
f0	Photon frequency (e/h)	2.4179883E14 Hz
λc	Compton wavelength	0.00242631058 nm
rad	1 radian	1 radian
twoπ	2π radians	6.28318530718 radians
angl	∡ in trig mode	180°
c3	Wien displacement constant	0.002897756 m·K
kq	k/q	0.00008617386 J/(K·C)
ε0q	ε0/q	55263469.6 F/(m·C)
qε0	q*εo	1.4185979E–30 F·C/m
εsi	Dielectric constant	11.9
εox	SiO_2 dielectric constant	3.9
I0	Reference intensity	0.000000000001 W/m^2

To view the constants library

1. Press APPS CONSTANTS LIB.
2. Press ENTER.

In RPN mode: follow step 1 only.

To view the value and units of a particular constant

1. Highlight the constant whose value you want to see.

   You can either press ▼ or ▲ until the constant is highlighted, or press ALPHA followed by the first character of the constant's abbreviation. For example, to quickly find the acceleration of gravity, enter ALPHA ↰ G. (The left-shift key is needed because the abbreviation for the deceleration of gravity is a lower-case *g*.)
2. Press VALUE.

   The numerical value of the constant is displayed.
3. If the units are not displayed and you wish to see them, press UNITS.
4. If SI units are displayed and you prefer imperial units, press ENGL. (Alternatively, if imperial units are displayed and you prefer to see SI units, press SI.)

To copy a constant to the stack or history

1. Highlight the constant whose value you want to copy.

   You can either press ▼ or ▲ until the constant is highlighted, or press ALPHA followed by the first character of the constant's abbreviation.
2. If you want to copy the value and its units, display the units. (If the units are not displayed, press UNITS.)
3. Press →STK.

Once copied to the stack or history, you can include the value or unit object in calculations.

To return to the stack or history, press QUIT or CANCEL.

To include a constant in an algebraic expression

You can include a constant in an expression you are creating on the command line or in Equation Writer.

1. At the position in the expression where you want to include a constant, type CONST.
2. Press ↰().
3. With the cursor between the parentheses, type the abbreviation for the constant.

   For example: 4.56*CONST(g).

When the expression is evaluated, the constant is also evaluated. (In the above example, the answer, in SI units, is 44.718824_m/s^2.)

Chapter 8

Number bases

Entering and displaying binary integers

You can enter and display integers in one of four forms:

- decimal (base 10)
- hexadecimal (base 16)
- octal (base 8) or
- binary (base 2).

You use the pound symbol (#) together with a suffix to indicate the base of a number. The suffixes are:

d (decimal)

h (hexadecimal)

o (octal) and

b (binary).

For example, # 182d, # B6h, # 266o, and # 10110110b are all ways of representing 182. In this guide, numbers represented in this way are called *binary integers* regardless of their base.

The current base setting determines which base is used to display integers.

On the HP 49G, binary integers can take from 1 to 64 bits, depending on the current *wordsize.*

To set the base

To set the base, select the appropriate command from the command catalog (CAT):

- DEC (to set the base to decimal)
- HEX (to set the base to hexadecimal)
- OCT (to set the base to octal)
- BIN (to set the base to binary).

The settings for flags –11 and –12 determine the current base. The DEC, HEX, OCT, and BIN commands control the settings of these flags:

		–11	
		Set	Clear
–12	Set	Hex	Bin
	Clear	Oct	Dec

An annunciator on your default screen indicates the current base setting.

To set the wordsize

1. Press ↱ BASE STWS.
2. Key in the new wordsize (from 1 to 64).
3. Press ENTER.

In RPN mode: follow steps 2 and 1.

If a binary integer argument exceeds the current wordsize, the excess leading bits are dropped before the command is executed. If necessary, results are also truncated.

To recall the current wordsize

1. Press ↱ BASE RCWS.
2. Press ENTER.

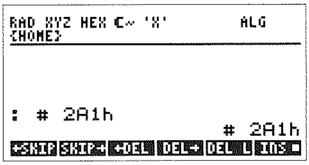
In RPN mode: follow step 1 only.

To enter a binary integer

1. Press ↰ #.
2. Enter the value of the binary integer.
3. Enter the suffix that indicates the base you want to use: d, h, o, or b. Your entry must be a lower-case character.

Step 3 is optional if the base you want to use is the same as the current base setting.

4. Press ENTER.



```
RAD XYZ HEX C~ 'X'          ALG
{HOME}

: # 2A1h
                         # 2A1h
```

Binary integers are displayed on the HP 49G with a space after the # sign. You do not need to enter a space when creating a binary integer.

If you specify a base other than the current base setting, the HP 49G converts the binary integer you entered to an integer to the base of the current setting. If you want to see the binary integer you entered, press ⬑ CMD.

For example, if your current base setting is hexadecimal and you enter # 1101b, the calculator displays your entry as # Dh. To see your entry, press ⬑ CMD. Your entry is at the top of the list of last four executed commands.

Binary integer arithmetic

To add, subtract, multiply, or divide two binary integers

1. Enter the first binary integer.
2. Press the key for the operation you want to perform: + − × or ÷.
3. Enter the second binary integer.
4. Press ENTER.

In RPN mode: follow steps 1, 3, and 2.

The two binary integers do not have to be to the same base.

The answer is expressed in the current base setting, and the two arguments are converted to that setting.

Note that any remainder from a division is lost and the answer is truncated to be an integer.

To find the negative of a binary integer

1. Press CAT NEG.
2. Enter the binary integer.
3. Press ENTER.

In RPN mode: follow steps 2 and 1.

The negative of a binary number is its *two's complement* (all bits inverted and 1 added).

To convert a binary integer to a different number base

1. Press CAT *n*, where *n* is the command that represents the base you want to convert to: DEC, BIN, HEX, or OCT.
2. Enter the binary integer.
3. Press ENTER.

For example, to convert # 1101b to hexadecimal, enter HEX(# 1101b) and press ENTER.

In RPN mode: follow steps 2 and 1.

Note that converting a binary integer to a different number base also changes the base setting to the base that you converted the integer to.

To convert a binary integer to a real number

1. Press ↱ BASE B→R.
2. Enter the binary integer.

   For example, # 3Ah.
3. Press ENTER.

   The answer is 58.

In RPN mode: follow steps 2 and 1.

To convert a real number to a binary integer

1. Set the base to the type that you want to convert to. See "To set the base" on page 8-1.
2. Press ↱ BASE R→B.
3. Enter the real number.
4. Press ENTER.

In RPN mode: follow steps 2 and 1.

If necessary, the real number is rounded to an integer before converting. Negative real numbers are converted to # 0 and real numbers $\geq 1.84467440738 \times 10^{19}$ are converted to the largest binary integer (# FFFFFFFFFFFFFFFFH, for example).

Using Boolean operators

The HP 49G provides a number of commands that enable you to perform Boolean operations and comparisons on binary integers. These commands—available by pressing ↱ BASE LOGIC—are illustrated in the following table. The input syntax shown assumes that you are in algebraic mode.

Commands	Examples	
	Input	Output
AND Logical bit-by-bit AND of two arguments. Compares corresponding bits and returns true (1) if both bits are 1.	# 1100b AND # 1010b	# 1000b
NOT Returns the one's complement of the argument. Each bit in the result is the complement of the corresponding bit in the argument.	NOT (#111b)	#111111110000b[a]
OR Logical bit-by-bit OR of two arguments. Compares corresponding bits and returns true (1) if either bit is 1.	# 1100b OR # 1010b	# 1110b
XOR Logical bit-by-bit exclusive OR of two arguments. Compares corresponding bits and returns true (1) if only one of the bits is 1.	# 1101b XOR # 1011b	# 110b

[a] This answer assumes that the wordsize has been set to 12.

Manipulating bits and bytes

The following commands enable you to manipulate binary integers one bit or one byte at a time. The commands are available by pressing ↱ BASE BIT OR ↱ BASE BYTE. Unless otherwise stated, each example assumes the wordsize is set to 24.

Commands	Example	
	Input	Output
ASR Arithmetic Shift Right. Performs 1 bit arithmetic right shift. The most significant bit is regenerated.	# 1100010b	# 110001b
RL Rotate Left. Binary integer rotates left one bit. (The example assumes that the wordsize is 4.)	# 1100b	# 1001b
RLB Rotate Left Byte. Binary integer rotates left one byte.	# FFFFh	# FFFF00h
RR Rotate Right. Binary integer rotates right one bit. (The example assumes that the wordsize is 4.)	# 1101b	# 1110b
RRB Rotate Right Byte. Binary integer rotates right one byte.	# A0B0C0h	C0A0B0h
SL Shift Left. Binary integer shifts left one bit.	# 1101b	# 11010b
SLB Shift Left Byte. Binary integer shifts left one byte.	# A0B0h	# A0B000h
SR Shift Right. Binary integer shifts right one bit.	# 11011b	# 1101b
SRB Shift Right Byte. Binary integer shifts right one byte.	# A0B0C0h	# A0B0h

Chapter 9

Lists and Sequences

Creating Lists

To enter a list from the keyboard

1. Press {}. The braces indicate the beginning and end of a list.
2. Enter the elements of the list, separating each with a comma (↱ ,).
3. Press ENTER.

   Note that the list is displayed without commas.

To assemble a list from a set of stack objects

If you are working in RPN mode, you can assemble a list from a contiguous set of objects already on the stack.

1. Enter the objects on the stack (or place them at the bottom of the stack: level 1, level 2, and so on).
2. Enter the number of objects you want in the list on the first level of the stack.
3. Execute the →LIST command to convert the objects to a list.

   The TO LIST command can be executed by pressing ↰ PRG LIST →LIST.

To append a new object to the beginning of a list

1. Enter or select the new object.
2. Press +.
3. Enter or select the list.
4. Press ENTER.

In RPN mode: follow steps 1, 3, and 2.

To append a new object to the end of a list

1. Enter or select the list.
2. Press (+).
3. Enter or select the new object.
4. Press (ENTER).

In RPN mode: follow steps 1, 3, and 2.

List Processing

To apply a one-argument command to each element in a list

The order in which you enter the command and the list depends on the type of function it is: *prefix* or *postfix*.

A **prefix function** is one whose name or abbreviation usually comes before its arguments; for example, SIN(x) and SQ(x, y).

A **postfix function** is one whose name or abbreviation usually comes after its arguments; for example, 6!.

The following example shows how to the apply a one-argument prefix command to a list if you are working in algebraic mode. The table below the example gives the order of the steps for postfix functions, and for when you are working in RPN mode.

1. Enter or select the command.

   For example, SQ().
2. Enter or select the list.

   For example, SQ({3, 4, 5}).
3. Press (ENTER).

   The answer is a new list, with each element the square of each element in the original list: {9, 16, 25}.

If you are executing a postfix function, or working in RPN mode, the steps you should take are given in the following table.

	Prefix	Postfix
Algebraic	1, 2, 3	2, 1, 3
RPN	2, 3, 1	2, 1

Another example: To find the factorial of 3, 4, and 5 while working in algebraic mode.

1. Enter or select the list: {3, 4, 5}.
2. Select the factorial command: ↰ MTH PROBABILITY !
3. Press ENTER.

   The answer is {6, 24, 120}.

To add corresponding elements of two lists

1. Enter or select the first list.

   For example, {4, 7, 9}.
2. Select the ADD command.

   ↰ MTH LIST ADD
3. Enter the second list.

   For example, {2, –4, –8}.

   The number of elements in the second list must be the same as the number of elements in the first list.
4. Press ENTER.

   The answer is {6, 3, 1}.

In RPN mode: follow steps 1, 3, and 2.

To concatenate two lists

To concatenate two lists is to form a list made up of the elements of both lists. The order of the elements in each sub-list is preserved.

1. Enter the list whose elements will form the first part of the concatenated list.

   For example, {1, 2, 3}
2. Press +.
3. Enter the list whose elements will form the latter part of the concatenated list.

   For example, {4, 5, 6}.
4. Press ENTER.

   The answer is {1, 2, 3, 4, 5, 6}.

In RPN mode: follow steps 1, 3, and 2.

To subtract, multiply, or divide corresponding elements of two lists

1. Enter the first list.
2. Press −, ×, or ÷.
3. Enter the second list.

   The number of elements in the second list must equal the number of elements in the first list.
4. Press ENTER.

In RPN mode: follow steps 1, 3, and 2.

For example, {2, 4, 7} × {3, 5, –4} yields {6, 20, –28}.

Applying a function or program to a list

The DOLIST command enables you to run programs or execute functions on *groups* of lists.

To run a program or execute a function on lists

When you are operating on a number of lists—especially long lists—it may be easier and clearer if you are in RPN mode (as in the case of the following example).

1. Enter the lists.
2. Enter the number of lists to be operated on. This number is the number of elements in each list operated on in each iteration of the function or program.
3. Enter a program or function.
4. Execute DOLIST.

Example: Apply the function defined by $a_n + (b_n \times c_n)$ to three lists, a, b, and c.

1. Enter the lists.

   For example, {1, 2, 3, 4}, {4, 5, 6, 7}, and {7, 8, 9, 11}.
2. Enter the number of lists to be operated on (for example, 3).
3. Enter the function.

   « * + »
4. Execute the DOLIST command.

   ↰ PRG LIST PROCEDURES DOLIST

   The answer is {29, 42, 57, 81}.

To apply a procedure sequentially to elements of a list

The DOSUBS command enables you to apply a program or function to the elements in the list one after the other. In RPN mode, the procedure is:

1. Enter the list.
2. Enter the frame index. This is the number of elements affected by each iteration of the function. For example, entering 3 would take 3 elements from the list and use them as arguments for the function.
3. Enter a program or function.
4. Execute DOSUBS.

Example: Find the 2-element moving average of {2, 4, 8, 16, 50}.

1. Enter the list.
2. Enter the frame index.

   In this example, the number is 2, as you want to find the average of *two* numbers each time.
3. Enter the program.

   « + 2 / »
4. Execute the DOSUB command.

   ↰ PRG LIST PROCEDURES DOSUBS.

   The answer is {3, 6, 12, 33}.

To execute a function on every element of a list

The STREAM command enables you to apply a function recursively to every element in a list. The command takes the first two elements, executes a function, takes the result and the next element, executes the function again, and so on, until there are no more arguments for the function to work on.

In RPN mode, the steps are:

1. Enter the list.

   For example, { 1 2 3 4 5 }.
2. Enter a program or function that takes two arguments and returns one result.

   For example, « * ».
3. Execute STREAM.

   ↰ PRG LIST PROCEDURES STREAM.

   In this example, the function multiplies all the elements in the list. The answer is 120.

You can also use the product list command to calculate the product of all the elements in a list. See "To find the product of the elements in a finite list" on page 9-10.

List Manipulations

The following functions provide ways to manipulate the elements of a list:

Function	Description
↰ MTH LIST SORT	Sorts the elements in a list in ascending order. In RPN mode, the list must be on level 1.
↰ MTH LIST REVLIST	Reverses the order of the elements in a list. In RPN mode, the list must be on level 1.
↰ PRG LIST ELEMENTS HEAD	Returns the first element in the list. In RPN mode, the list must be on level 1.
↰ PRG LIST ELEMENTS TAIL	Returns a list of all the elements in the list except the first element. In RPN mode, the list must be on level 1.
↰ PRG LIST ELEMENTS GET	Returns the element in the list (argument 1/level 2) corresponding to a specified position (argument 2/level 1). For example, GET({1, 4, 7, 8}, 3) returns 7, since 7 is the third element in the list.
↰ PRG LIST ELEMENTS GETI	Similar to the GET command (see above) but also returns the specified position number incremented by 1 (and the original list). For example, GETI({1, 4, 7, 8}, 3) returns {1, 4, 7, 8}, 4 (the specified position number + 1), and 7 (the third element in the list).
↰ PRG LIST ELEMENTS PUT	Replaces an element at a particular position (argument 2/level 2) of a list (argument 1/level 3), with a new element (argument 3/level 1). For example, PUT({1, 2, 3}, 2, 5) returns {1, 5, 3}.
↰ PRG LIST ELEMENTS PUTI	Similar to the PUT command (see above) but also returns the specified position number incremented by 1. For example, PUTI({1, 2, 3}, 2, 5) returns {{1, 5, 3}, 3}.
↰ PRG LIST ELEMENTS SIZE	Returns the number of elements in a list. In RPN mode, the list must be on level 1.

Function (Continued)	Description
↰ PRG LIST ELEMENTS POS	Returns the position of the first occurrence of an element (argument 2/level 1) in a specified list (argument 1/level 2). For example: POS({2, 4, 6, 1, 2, 3, 4}, 4) returns 2 (since the first occurrence of 4 is at position 2 in the list.
↰ PRG LIST OBJ→	RPN command to disassemble a list into its elements and return (to level 1) the number of elements that were in the list. Each element is placed on a separate level of the stack.
↰ PRG LIST SUB	Creates a sub-list of elements from a list (argument 1/level 3) starting with the element at one position (argument 2/level 2) and ending with the element at another position (argument 3/level 1). For example, SUB({1, 2, 3, 4, 5}, 2, 4} returns {2, 3, 4}.
↰ PRG LIST REPL	Replaces the elements of a list (argument 1/level 3) with the elements from another list (argument 3/level 1) starting at a specified position (argument 2/level 2). For example, REPL({6, 7, 8, 2}, 2, {1, 2, 3} returns {6, 1, 2,3 }.

Sequences

Sequence commands automate the generation of a list from the repeated execution of a function or program.

To generate a sequence

In algebraic mode:

1. Specify the sequential calculation command.
   ↰ PRG LIST PROCEDURES SEQ.
2. Enter the function or program (or its name).
3. Enter the index variable name.
4. Enter the initial value for the variable.
5. Enter the final value for the variable.
6. Enter the step size of the increment.
   The number of elements generated is the integer portion of ((*final* – *initial*)/*step*) + 1.
7. Press ENTER.

In RPN mode: follow steps 2, 3, 4, 5, 6, and 1.

Example: Generate a list of the squares of the integers from 15 to 19.

1. Press ↰ PRG LIST PROCEDURES SEQ.
2. Enter ↱ ' ↰ x^2 X ▶ ▶ and press ↱ ,.
   Note that tick marks are needed to delimit the algebraic object.
3. Enter X and press ↱ ,.
   x is the variable that will be evaluated at each iteration.
4. Enter 15 and press ↱ ,.
   15 is the initial value, that is, the value used in the first iteration.
5. Enter 19 and press ↱ ,.
   19 is the final value, that is, the value used in the last iteration.
6. Enter 1.
   1 is the step value, that is, by how much the initial value will be incremented after each iteration.
7. Press ENTER.

The result is {225, 256, 289, 324, 361}.

If you had entered 2 for the step value at step 6, then every second integer in the specified range would have been included in the iterations, and the result would have been {225, 289, 361}.

To find the sum of the elements in a finite list

1. Select the list sum command.

   ↰ MTH LIST SLIST
2. Enter or select the list.

   For example, {2, 4, 7}.
3. Press ENTER.

   The answer is 13.

In RPN mode: follow steps 2 and 1.

To find the product of the elements in a finite list

1. Select the list product command.

   ↰ MTH LIST PLIST
2. Enter or select the list.

   For example, {2, 3, 4}.
3. Press ENTER.

   The answer is 24.

In RPN mode: follow steps 2 and 1.

To find the set of first differences of a finite sequence

The first differences for the list $\{ x_1\ x_2 \ldots x_n \}$ is defined as $\{ x_2 - x_1 \ldots x_n - x_{n-1} \}$.

1. Select the list differnces command.

   ↰ MTH LIST DLIST
2. Enter or select the list.

   For example, {2, 5, 11, 13, 33}.
3. Press ENTER.

   The answer is {3, 6, 2, 20}.

In RPN mode: follow steps 2 and 1.

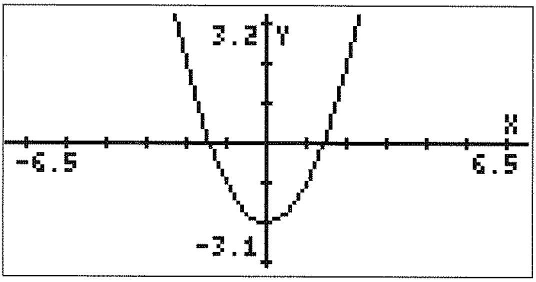
Chapter 10

Advanced plotting options

Labelling and relocating the axes

To label the coordinate axes with the variable names

The names of the independent and dependent variables, and the coordinates (in user-units) of the largest and smallest *displayed* values for each variable, can be added to the plot after it has been drawn. The figure below shows labels added to the plot of $y = x^2 - 2$ (assuming that you have used the default settings).



To label the axes:

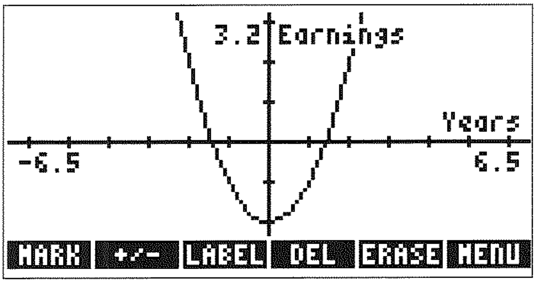
1. Press EDIT.
2. Press NXT to display the second page of the function-key menu.
3. Press LABEL.

   You may need to hide the menu to see the lowest label on the vertical axis. You hide the menu by pressing ⊕ or ⊖, and redisplay it by pressing ⊕ or ⊖.

To label the axes with user-defined labels

1. Select the AXES command from the command catalog (CAT).
2. Enter a list containing the horizontal and vertical axis labels as strings: { "*h-label*" "*v-label*" }.

   For example: AXES({"Years", "Earnings"}).
3. Press ENTER to store the labels.
4. Press ↰ GRAPH to display the plot again.
5. Press EDIT.
6. Press NXT to display the second page of the function-key menu.
7. Press LABEL.

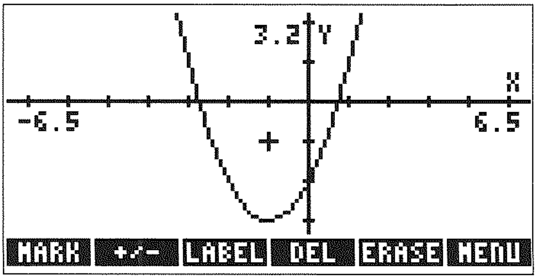


To have the axes intersect at a point other than (0,0)

1. Select the AXES command from the command catalog (CAT).
2. Enter a complex number representing the new point of intersection.

   For example: AXES((1,1)).
3. Press ENTER to store the new point of intersection.
4. Press either ↰ Y=, ↰ WIN, or ↰ 2D/3D.
5. Press ERASE.
6. Press DRAW.

The following diagram is of the same plot as in the previous illustration, but the axes now meet where x =1 and y =1.



Plotting programs

You can plot a program if it takes nothing from the stack, uses the independent variable in the program, and returns exactly one untagged number to the stack.

Examples

- **Real result.** Equivalent to the expressions $f(x)$ (function plots) and $r(\theta)$ (polar plots). For example, the program:

  ```
  « IF 'X<0' THEN '3*X^3-45X^2+350' ELSE 1000 END »
  ```

  plots $f(x) = 3x^3 - 45x^2 + 350$ if $x<0$ and $f(x) = 1000$ if $x \geq 0$.

  Store the program in *EQ*, select autoscaling, and draw the plot.

- **Complex result.** Equivalent to $(x(t), y(t))$ (parametric plots). For example, the program:

  ```
  « 't^2-2' →NUM 't^3-2t+1' →NUM R→C »
  ```

  plots the parametric equations $x = t^2 - 2$ and $x = t^3 - 2t + 1$.

  Store the program in *EQ*, make *t* the independent variable, select autoscaling, and draw the plot.

Plotting range vs. display range

The *plotting range* is the range of the independent variable (or variables) over which the current equation is evaluated. If you don't specify the plotting range, the HP 49G uses the x-axis display range (specified by XRNG or by H-VIEW) as the plotting range. However, you can specify a plotting range that is different from the x-axis display range:

- For polar and parametric plots, the independent variable isn't related to the x-axis variable; therefore you specify the plotting range to control the range of the independent variable.
- For truth and conic plots, you can shorten plotting time by specifying plotting ranges that are smaller than the x- and y-axis display ranges. These plot types require you to specify the *dependent* variable. You can specify a plotting range different from the y-axis display range.

You can make *PICT* larger than its default size (131 by 64 pixels) and either keep the same x and y scale factors (which extends the display range), or keep the same display range (which extends the scale and appears to stretch the plot.

To check the current size of PICT

1. Press ↰ RCL to select the Recall command.
2. Type PICT.
3. Press ENTER.

   The message "Graphic *width* × *height*" is returned, indicating the current dimensions of PICT.

In RPN mode: follow steps 2 and 1.

To change the size of PICT

To keep the same scaling:

1. Press ↰ PRG PICT PDM to select the PICT DIMENSION command.
2. Enter a complex number to indicate the coordinates, in user units, of one corner of PICT.
3. Press ↱ ,.

4. Enter a complex number to indicate the coordinates, in user units, of the corner of PICT diagonally opposite the corner specified at step 2 above.

   For example, PDM((–6,–6), (6,9)).

5. Press ENTER.

Press ↰ GRAPH to see the results of the re dimensioning of PICT.

In RPN mode: follow steps 2, 4, and 1.

To keep the same display ranges:

1. Press ↰ PRG PICT PDM to select the PICT DIMENSION command.
2. Enter a binary integer to indicate the horizontal size of PICT in pixels.
3. Press ↱ ,.
4. Enter a binary integer to indicate the vertical size of PICT in pixels.

   For example, PDM(#105h, #3Fh).

5. Press ENTER.

Press ↰ GRAPH to see the results of the redimensioning of PICT.

In RPN mode: follow steps 2, 4, and 1.

The result of the PDIM command depends on the type of coordinates—user-units or pixels—although both forms change the size of *PICT.*

To use computed values for plotting or display ranges

1. On the Plot Window input form, highlight the range field whose value you wish to compute.
2. Press NXT CALC.
3. Perform the desired calculation.
4. If it is not already a real number, convert the result to a real number by pressing ↱ →NUM.
5. Press OK to return to the Plot Window input form. The result of the calculation will be in the field you highlighted at step 1.

Saving and Restoring Plots

A plot consists of several components:

- The plot picture (that is, a graphic object).
- The current equation or equations (stored in the reserved variable *EQ*).
- The current plot parameters (stored in the reserved variable *PPAR* and, in the case of three-dimensional plot types, *VPAR*).
- Flag settings that determine plotting and display options.

You have the option to save any or all of these plot components in a variable so that you can retrieve them later. Here are two useful approaches:

- Save just the plot picture in a variable. This is a simple procedure (see below) but each plot picture uses about one kilobyte of memory.
- Save the current *EQ*, *PPAR*, *VPAR* (if necessary), and flag settings in a list. See “To save a reconstructable version of the current plot” on page 10-7. The plot can be reconstructed by restoring each of these values contained. (See “To reconstruct a plot from its stored version” on page 10-8.)

To save the current plot picture in a variable

1. While viewing the plot, press STO▶.

   A copy of the plot is placed in history.
2. Press CANCEL until you return to history.
3. Press STO▶.
4. Enter a name for the plot.
5. Press ENTER.

In RPN mode: follow steps 1, 2, 4, and 3.

To view a plot picture stored in a variable

1. Press VAR.
2. Press the function key corresponding to the variable containing the plot picture.

   You may have to press NXT a number of times to display the variable you want. You may also have to change directories if the variable is not in the current directory.
3. Press ▼ to display the plot.

To save a reconstructable version of the current plot

1. After drawing the plot, press CANCEL to return to your default screen.
2. Press ↰ { }.
3. Press VAR.
4. Press EQ.
5. Press ↱ ,.
6. Press PPAR.
7. Press ↱ ,.
8. If your plot was three-dimensional, press VPAR and ↱ ,.
9. Press CAT RCLF.
10. Press ▶ until the cursor is outside the list.
11. Press STO▶.
12. Enter a name for the list.
13. Press ENTER.

To reconstruct a plot from its stored version

This procedure is best done in RPN mode.

1. Press VAR.
2. Press the function key associated with the variable that contains the stored version of the plot's components.
3. Press ↰ PRG TYPE OBJ→ to disassemble the list and put the components onto the stack.
4. Press ◀ to delete the object on level 1. This is the number of items in the original list and is not needed in the procedure. The new level-1 object is the flag settings current at the time you stored the plot components variable.
5. Press CAT STOF to reset the flag settings. Note that your current flag settings will be lost.
6. If the plot is a three-dimensional plot, press ↱ ', type VPAR and press STO▸ to reset *VPAR* to its earlier values.
7. press ↱ ', type PPAR and press STO▸ to reset *PPAR* to its earlier values.
8. press ↱ ', type EQ and press STO▸ to reset *EQ* to its earlier value.
9. Press ↰ Y= ERASE and DRAW to redraw the plot.

Chapter 11

Memory

This section describes the memory structure of the HP 49G. It describes how to create backup objects of data that you want to save, and how to use libraries to add functionality to the calculator.

How memory is structured

The HP 49G contains a total of 2.5 Mb of memory. Of this memory:

- 1 Mb is used to hold the operating system.
- 1.5 Mb is used for performing the operations that you specify, and for storing data that you want to keep. The HP 49G needs a minimum of approximately 200 bytes free in order to perform operations.

The memory of the HP49G is divided into 4 areas:

- The Home directory
- Port 0, labelled IRAM in File Manager

  The Home directory and port 0 share the same area of memory. That is, the more data you store in the HOME directory, the less data you can store in port 0 and vice versa. The total size of objects in these two areas cannot exceed 241 Kb.
- Port 1, labelled ERAM in File Manager

  Port 1 can contain up to 255KB of data.
- Port 2, labelled FLASH in File Manager

  Port 2 can contain up to 1085KB of data.

The Home directory, port 0 and port 1 are part of RAM. Leaving the calculator without batteries can cause RAM to loose its data. When changing batteries, ensure that you do not leave the calculator without batteries for longer than two minutes.

The data in port 2 is stored in the Flash ROM. This data does not depend on batteries. It is preserved even if you leave the calculator without batteries for a long time.

Unlike the HOME directory, port memory cannot be subdivided into directories. A port can only contain two types of objects:

- backup objects
- library objects.

Accessing port contents

In order to access the contents of the variables stored in the ports you can use File Manager. For details on how to use File Manager, refer to the *User's Guide.*

Alternatively, you can access the contents of libraries and ports, by pressing ↱ LIB. A function key menu of the attached libraries and the available ports is displayed.

- To view the contents of a port, press the function key that corresponds to the port. The contents are displayed as a function key menu.
- To access the functions in an attached library, press the function key that corresponds to the library. The functions it contains are displayed as a function key menu. To run a function, press the function key that corresponds to the function.

Backup objects

You generally use backup objects to save data that you want to keep for a long time. Backup objects are special copies of objects from your HOME directory. Note the following points regarding backup objects:

- Backup objects can only exist in port memory.
- You cannot modify the contents of a backup object.
- You can store either a single calculator object as a backup object or an entire directory. You cannot create a backup object for selected objects in a directory.

The HP 49G monitors the integrity of backup objects. When you create a backup object in a port, the HP 49G calculates a CRC value (cyclic redundancy check) based on the binary data that the object contains. A CRC value is a checksum calculated from the data. This value is stored with the object.

When you restore a backup object, the calculator re-calculates the CRC value and compares it to the original value. If the CRC values are not the same, the calculator warns you that the data is corrupt.

You can use File Manager to copy and delete backup objects in a similar fashion to normal calculator objects. In addition, there are specific commands for manipulating backup objects.

Backing up and restoring HOME

You can back up and restore the contents of the entire HOME directory in a backup object. This includes all variables, and any key assignments and alarms that you have created.

Backing up

To backup the HOME directory in RPN mode, enter the following:

```
:Port_Number: Backup_Name ARCHIVE
```

To backup the HOME directory in algebraic mode, enter the following:

```
ARCHIVE(:Port_Number: Backup_Name)
```

Restoring

To restore the HOME directory in RPN mode, enter the following:

```
:Port_Number: Backup_Name RESTORE
```

For example, to backup the home directory as MYHOME in port 2, use the following command in algebraic mode:

```
ARCHIVE(:2: MYHOME)
```

To restore the HOME directory in algebraic mode, enter the following:

```
RESTORE(:Port_Number:Backup_Name)
```

To restore the HOME directory backup from the previous example, use the following command:

```
RESTORE(:2:MYHOME)
```

When you restore a HOME directory backup, the backup directory overwrites the current HOME directory and its contents. Any data not backed up is lost.
After you restore a HOME directory, the calculator restarts. The contents of history or the stack is lost.

Storing and deleting backup objects

There are three ways to create a backup object:

- Use File Manager to copy the object to a port. With this method, the backup object has the same name as the original object.
- Use the STO command to copy the object to a port, and assign it a name. See *Command Reference Part D* for details of the STO command.
- Use the ARCHIVE command to create a backup of the HOME directory, and assign a name to the backup object. See *Command Reference, Part B* for details of the ARCHIVE command.

There are two ways to delete a backup object from a port:

- Use File Manager to delete the object as you would a normal calculator object.
- Use the PURGE command
  - In RPN mode, use the following syntax:
    `:Port_Number: Backup_Name PURGE`
  - In algebraic mode, use the following syntax:
    `PURGE(:Port_Number: Backup_Name)`

    For example, to delete a backup object in port 1 named "MYBACKUP":
    `PURGE(:1: MYBACKUP)`

You can restore a backup object by using File Manager to copy it to the Home directory. When you restore a backup object, the HP 49G checks the object's integrity by calculating the CRC value and comparing it to the one stored with the backup object. An error message appears if the data has been corrupted.

Using data in backup objects

You cannot modify the contents of backup objects. You can, however, use the contents in calculator operations. For example, if your backup object contains a program, you can run the program. If your backup object contains a value assigned to a variable, you can use the variable in calculator operations.

To view a value saved in a backup object, or to run a program stored in a backup object, you can use File Manager as you would with a normal object.

You can also execute an object from the command line as follows:

- In RPN mode:
  - To evaluate a backup object, enter:
    ```
    :Port_Number: Backup_Name variable_name EVAL
    ```
  - To recall a backup object to the command line, enter:
    ```
    :Port_Number: Backup_Name variable_name RCL
    ```
- In algebraic mode:
  - To evaluate a backup object, enter:
    ```
    EVAL(:Port_Number: Backup_Name variable_name)
    ```
  - To recall a backup object to the command line, enter:
    ```
    RCL(:Port_Number: Backup_Name variable_name
    ```

You can also use ↱ LIB as described in "Accessing port contents" on page 11-2.

If you do not remember in which port a backup object is stored, you can use the & symbol in place of the port number. This causes the HP49G to search all the ports, and then the HOME directory.

You can also refer to a port by tags:

"E" and "ERAM" refer to port 1

"F" and "FROM" refer to port 2

Library objects

A library is a collection of objects that extend the calculator's functionality. You can execute objects in a library, but you can neither view, nor edit them. You can obtain libraries from various web sites.

Installing and attaching a library

To install a library, perform the following:

1. Copy the library to your HOME directory.
   - from a PC, use the Connectivity Kit.
   - From another calculator, use the calculator-to-calculator connection cable.
2. Install the library in a port.

   To install the library in a port, use the following commands. In the commands, *n* is the number of the port in which you want to store your library, and `Library_variable` is the name of the variable that contains the library. This is the variable that you created when you copied the library to HOME.

   – In RPN mode, with `Library_variable` on stack level 1:

   ```
   port_number STO
   ```

   – In algebraic mode:

   ```
   STO(Library_variable, port_number)
   ```

3. Attach the library. A library generally contains a special routine to attach it. This routine runs when the calculator re-boots. To reboot, press simultaneously ON and F3, then release them.

   You can use the ATTACH command to attach a library without re-booting. See *Command Reference Part B* for details on the ATTACH command.

Some libraries need to be stored in port 0. Check the documentation for a library for more information.

A library has a library number. The calculator uses this number to refer to the library. In File Manager, the library's number appears as "L*xxxx*", where *xxxx* is the library number.

Deleting a library

To delete a library from a port, use the following commands. In the commands, `port_number` is the number of the port you stored your object in and `lib_number` is the library number of the library you want to purge.

- In RPN mode:
  `:port_number: lib_number PURGE`
- In algebraic mode:
  `PURGE(:port_number: lib_number)`

How the HP 49G manages memory

The following section explains how the calculator manages memory in the various ports.

Port 0

Port 0 and the HOME directory share the same memory space. This means that objects stored in port 0 decrease the amount of user memory available, and objects stored in user memory decrease the amount of memory available in port 0.

Port 1

Port 1 contains 255 Kb of storage space. Because of the way port 1 memory is structured, you cannot store objects larger than 128 Kb. This is because port 1 memory is in fact 2 discrete areas of 128 Kb. Objects must be stored in either one of the two areas.

Because of this, when storing large objects it may appear that you have enough memory in port 1 to store your object, but the calculator will not allow you to store the object if it is bigger than the free space in either area.

Port 2

Port 2 is part of the Flash ROM. As with port 1, it is not possible to store objects larger than 128KB. Flash ROM is organized as 8 areas of 128KB each and one area of 64KB.

The method that the system uses to manage Flash ROM can sometimes affect operations in port 2. When it erases data, the system must erase 128Kb areas at a time. It cannot erase single objects. When you delete an object from flash ROM, the object is simply flagged as deleted. It still occupies memory space.

Occasionally when you want to store an object in Flash ROM, the system needs to remove files flagged as deleted. It does this by copying all the non-flagged object to RAM, erasing the entire Flash ROM, and then moving the non-tagged objects back to flash ROM.

Depending on how much data you have stored in flash ROM, up to 128KB of RAM might be needed for this operation. If there is not enough RAM, an error occurs, and the following message is displayed:

```
Ins. Mem. Need x bytes.
```

In this event, to store your object, you need to free up at least *x* bytes of RAM.

To list a port's contents, and find free memory

You can use the PVARS command to list the contents of a port. The PVARS command returns a list containing the port number and name of each object in the port, and the amount of free memory in the port.

Chapter 12

Date and time arithmetic

Date and time formats

The following table illustrates the date and time formats available on the HP 49G. The time and date illustrated is 4:31:04 PM on March 21, 2001.

Clock Display	Format	Number Format
Date:		
03/21/2001	Month/day/year format	3.212001
21.03.2001	Day.month.year format	21.032001
Time:		
04:31:04P	12-hour format	16.3104
16:31:04	24-hour format	16.3104

Table 12-1: Date and time formats

To set the format of the date and time

1. Press ↱ TIME.
2. Press ▼▼ to highlight the SET TIME, DATE... function and then press OK.
3. The SET TIME AND DATE input form is displayed.
4. Press the appropriate arrow keys to highlight the format field you want to change.

   The format fields are the two fields at the far right of the screen.
5. Change the format.

   You can do this by pressing +/- until the format you want is displayed, or by pressing CHOOS and selecting a format from a list of options.
6. If you want to change another format, repeat from step 4.
7. When you have finished, press OK.

   The input form closes and the date and time is displayed in the formats you set.

You can also use this procedure to set the date and time.

You can also set the date by executing the →DATE command, and set the time by executing the →TIME command.

Date and time tools

Numerous tools for working with dates and times are available from the Time menu.

Displaying the Time menu

There are two ways to access this menu:

- press ↱ TIME TOOLS
- hold down ↱ while pressing TIME.

To copy the date to the stack or history

1. Display the Time menu (see above).
2. Press DATE.
3. Press ENTER.

   The date is displayed in its *number format* (see the table on page page 12-1).

In RPN mode: follow steps 1 and 2.

To copy the time to the stack or history

1. Display the Time menu (see above).
2. Press TIME.
3. Press ENTER.

   The time is displayed in its *number format* (see the table on page page 12-1).

In RPN mode: follow steps 1 and 2.

Calculating with dates

To add days to a given date

1. Enter the date in number format (see the table on page page 12-1).

   For example: 3.212001 (that is, March 21, 2001).
2. Press CAT DATE+.
3. Enter a real number representing the number of days you want to add to the date entered at step 1.

   For example 13.
4. Press ENTER.

   The answer is 4.032001 (that is, April 3, 2001).

In RPN mode: follow steps 1, 3, and 2.

To subtract days from a given date

1. Enter the date in number format (see the table on page page 12-1).

   For example: 3.212001 (that is, March 21, 2001).
2. Press CAT DATE+.
3. Enter a real number representing the number of days you want to subtract from the date entered at step 1.

   For example 26.
4. Press +/-.

   This key converts the number of days entered at step 3 to a negative number, so that the effect of applying the DATE+ command is to subtract days.
5. Press ENTER.

   The answer is 2.232001 (that is, February 23, 2001).

In RPN mode: follow steps 1, 3, 4, and 2.

To determine the number of days between two dates

1. Press CAT DDAYS.
2. Enter the first date in number format (see the table on page page 12-1). For example: 3.212001 (that is, March 21, 2001).
3. Press ↱ ,.
4. Enter the second date, also in number format. For example 5.232001 (that is, May 23, 2001).
5. Press ENTER. The answer is 63.

In RPN mode: follow steps 2, 4, and 1.

Calculating with times

You can work with times expressed in decimal format or in HMS format. A number in HMS format is displayed as *H.MMSSs* where:

H = zero or more digits representing the number of hours

MM = two digits representing the number of minutes

SS = two digits representing the number of seconds, and

s = two or more digits representing the fractional part of seconds.

To convert decimal time to HMS format

1. Press CAT →HMS.
2. Enter the time in decimal format. For example: 7.125 (that is, seven-and-one-eighth hours).
3. Press ENTER. The answer is 7.07300 (that is, 7 hours, 7 minutes and 30 seconds).

In RPN mode: follow steps 2 and 1.

To convert a time in HMS format to decimal format

1. Press (CAT) HMS→.
2. Enter the time in HMS format.

   For example: 5.1231 (that is, 5 hours, 12 minutes, and 31 seconds).
3. Press (ENTER).

   The answer is 5.20861111 hours.

In RPN mode: follow steps 2 and 1.

To add times in HMS format

1. Press (CAT) HMS+.
2. Enter one time in HMS format.

   For example: 5.1231 (that is, 5 hours, 12 minutes, and 31 seconds).
3. Press (↱)(,).
4. Enter the other time, also in HMS format.

   For example: 4.5320 (that is, 4 hours, 53 minutes, and 20 seconds).
5. Press (ENTER).

   The answer is 10.0551 (that is, 10 hours, 5 minutes and 51 seconds).

In RPN mode: follow steps 2, 4, and 1.

To subtract times in HMS format

1. Press (CAT) HMS–.
2. Enter one time in HMS format.

   For example: 7.2201 (that is, 7 hours, 22 minutes, and 1 second).
3. Press (↱)(,).
4. Enter the other time, also in HMS format.

   For example: 4.13275 (that is, 4 hours, 13 minutes, and 27.5 seconds).
5. Press (ENTER).

   The answer is 3.08335 (that is, 3 hours, 8 minutes and 33.5 seconds).

In RPN mode: follow steps 2, 4, and 1.

System time

System time is kept in ticks of the clock. Each tick is 1/8192 of a second in duration. System time can be converted to standard time (in both decimal format and HMS format).

To display system time

1. Press ↱ TIME TOOLS TICKS.
2. Press ENTER.

   System time is displayed as a binary number.

In RPN mode: follow step 1.

The TICKS command is useful for measuring elapsed time.

To convert system time to HMS time

1. Press ↰ MTH BASE B→R.
2. Enter the system time as a binary number.

   For example: # 1D6861E78DF52h.
3. Press ENTER.

   The result of this step is the real number equivalent of the specified system time.
4. Press ÷.
5. Enter 29491200.

   The result of this step is the specified system time in decimal hours: 17, 542,402.5441.
6. Press CAT →HMS.
7. Press HIST.
8. Select the result of step 5.

   This result should now be the argument of the →HMS command.
9. Press ENTER.

   The result is 17,542,402.3239 (that is, 17,542,402 hours, 32 minutes, and 39 seconds.

In RPN mode: follow steps 2, 1, 5, 4, and 6.

To calculate elapsed time in seconds

1. To prepare to start timing, press ↱ TIME TOOLS TICKS.
2. To start timing, press ENTER.
3. To prepare to stop timing, press ↱ TIME TOOLS TICKS.
4. To stop timing, press ENTER.
5. Press −.
6. Press HIST.
7. Select the result of step 2.
8. Press ENTER twice.
9. Press ↰ MTH BASE B→R.
10. Select the result of step 8.
11. Press ENTER.
12. Press ÷.
13. Enter 8192.
14. Press ENTER.

In RPN mode: follow steps 1, 2, press TOOL STACK SWAP and continue with steps 5, 9, 13, and 12.

Chapter 13

Customization

Creating menus

The HP 49G enables you to create a custom menu. The menu can contain labels for operations, commands, and other objects that you create or group together for your convenience.

A custom menu is identified by the reserved variable CST. Therefore, you create a custom menu by naming a list of menu items CST. You can also use the MENU command to store a list in CST.

A custom menu is a type of function key menu; that is, the items on the menu appear across the bottom of your screen and can be selected by pressing the corresponding function key.

You can create a custom menu—that is, a CST variable—in each directory. This enables you to have a different custom menu in each directory.

To create a custom menu

1. Enter a list of label–object pairs, one pair for each menu item.

   The syntax for each label–object pair is { "label", object }. Also, to prevent an object being evaluated, you should enclose the object in tick marks ("). Each pair is a sub-list within the main list.

   Objects can be of many types, such as variable names, programs, strings, and unit objects. For example:

   {{"m^3", '1_m^s'},{"VOL",VOLUME},{"TIME", 'TIME()'}}

   In this example, the menu items will be labelled m^3, VOL, and TIME respectively. The first inserts the cubic metre unit expression, the second inserts the string "VOLUME", and the third returns the current time.
2. Press ▶ to move the cursor outside the main list.
3. Press STO▶.
4. Enter CST.
5. Press ENTER.

In RPN mode: follow steps 1, 4, and 3.

To display a custom menu

1. Press ↰ CUSTOM.

The menu labels appear across the bottom of your screen. You access a menu item by pressing the corresponding function key.

Customizing the keyboard

You can assign alternative functionality to any key on the keyboard (including alpha and shifted keys). This enables you to customize the keyboard to your particular needs.

Your customized keyboard is called the *user keyboard*, and it is active whenever you are in *user mode*.

User mode

To activate user mode

- If you want to execute only one operation on the user keyboard, press ↰ USER.
- If you want to execute several commands on the user keyboard, press ↰ USER ↰ USER. (You press ↰ USER again to deactivate the user keyboard.)

The ↰ USER key combination is a three-way switch, much like the ALPHA key. Pressing the combination once activates user mode only for the next operation, while pressing it twice consecutively locks user mode on, requiring a third press to turn the mode off. If you prefer to lock user mode on after pressing ↰ USER just once, set flag –61.

Assigning user keys

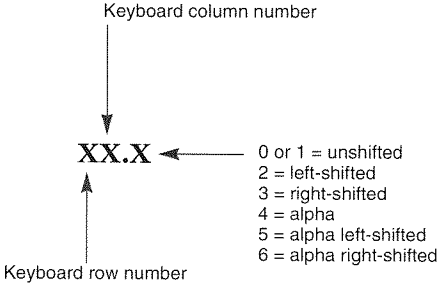
You can assign commands and other objects to a user key (including shifted keys).

To assign an object to a user key

1. Press CAT ASN.
2. Enter the object to be assigned to the user key.
3. Press ↱ ,.
4. Enter the key code that identifies the user key.

   The code is made up of the row number, column number, and shifted status. For example, 23.4 indicates the key at row 2 and column 3 when pressed with the ALPHA key. See the diagram below.
5. Press ENTER.

In RPN mode: follow steps 2, 4, and 1.



For example: 23.4 = ALPHA TOOL

User key codes

To assign a command to a user key

1. Press CAT STOKEYS.
2. Between the parentheses, enter a list with the command name as the first element—enclosed within tick marks—and the user key code as the second element.

   User key codes are explained in the illustration above.

   For example: STOKEYS({'TIME()', 31.0}).
3. Press ENTER.

In RPN mode: follow steps 2 and 1.

To unassign a user key

After you have assigned a user key, the assignment remains in effect until you re-assign the key (using the ASN or STOKEYS commands) or until you unassign the key. An unassigned user key reverts to its default use.

1. Press CAT DELKEYS.
2. Enter the user key code.

   User key codes are explained in the illustration on page 13-3.
3. Press ENTER.

In RPN mode: follow steps 2 and 1.

Disabling keys

Disabling keys in user mode

When you are in user mode, you can disable the keys that have not been assigned a special use (that is, all the keys that still perform their default function). To do this:

1. Press CAT DELKEYS.
2. Type S.
3. Press ENTER.

In RPN mode: follow steps 2 and 1.

Once disabled, a key can be enabled by assigning it a special use (see "To assign an object to a user key" on page 13-3) or by deactivating user mode (by pressing ↰ USER).

To enable disabled keys

1. Press CAT DELKEYS.
2. Type 0 (that is, zero).
3. Press ENTER.

In RPN mode: follow steps 2 and 1.

If you assign or disable the keys necessary to re-enable keys, or necessary to cancel user mode, you will be stuck in user mode. You will then need to reboot the calculator by holding ON, pressing and releasing F3 and then releasing ON.

Recalling and editing user key assignments

To recall the current user key assignments

1. Press CAT RCLKEYS.
2. Press ENTER

In RPN mode: follow step 1.

The RCLKEYS command returns a list of all the current user key assignments. Each assigned object is listed with its corresponding user key code. If the first item in the list is "S", then unassigned user keys are enabled; otherwise, unassigned users keys are disabled.

To edit the user key assignments

1. Recall the current user key assignments (see previous section).
2. Press ▼ to open the command line editor.
3. Edit the key assignments you want to change.
4. Press ENTER.
5. Press CAT STOKEYS.
6. Press HIST.
7. If the user key assignment list you just modified is not highlighted, highlight it.
8. Press ENTER twice.

In RPN mode: follow steps 1, 2, 3, 4, and 5.

Chapter 14

Computer Algebra Commands

This chapter details the computer algebra operations that are available on the HP 49G.

For each operation, the following details are provided:

Type: Function or command. Functions can be used as a part of an algebraic object and commands cannot. When working with functions or commands within Equation Writer:

- When you apply a function to an expression, the function appears as part of the expression. You need to ensure that the expression is selected, then press ↱ EVAL to apply the function to the selection.
- When you apply a command to an expression in Equation Writer, it is evaluated immediately.

Description: A description of the operation.

Access: The menu or choose-list on which an operation can be found, and the keys that you press to access it. If the operation is on a sub-menu, the sub-menu name is in SMALL CAPITALS after the keys.

Input: The input argument or arguments that the operation needs. If the operation uses more than one input argument, details of the arguments and the order in which you supply them are provided: argument order for algebraic mode and stack order for RPN mode.

Output: The output that the operation produces.

- In RPN mode, the outputs are placed on the stack.
- In algebraic mode, the outputs are written to a list.

As with the input arguments, the outputs for both algebraic and RPN mode are described.

Flags: Details of how flag settings affect the operation of the function or command.

Example: An example of the function or command.

See also: Related functions or commands.

Computer algebra command categories

The following is an index of the computer algebra commands, listed in menu order.

Algebra commands

EXPAND	14-19
FACTOR	14-21
LNCOLLECT	14-42
LIN	14-41
SOLVE	14-57
SUBST	14-58
TEXPAND	14-63

Arithmetic commands

DIVIS	14-15
FACTORS	14-22
LGCD	14-40
SIMP2	14-56
PROPFRAC	14-48

Arithmetic Integer commands

EULER	14-18
LÆR	14-29
ICHINREM	14-31
IDIV2	14-32
IEGCD	14-32
IQUOT	14-35
IREMAINDER	14-35
ISPRIME?	14-35
NEXTPRIME	14-44
PA2B2	14-45
PREVPRIME	14-47

Arithmetic Polynomial commands

ABCUV	14-6
CHINREM	14-11
DIV2	14-14
EGCD	14-16
FACTOR	14-21
FCOEF	14-22
FROOTS	14-23
GCD	14-25
HERMITE	14-27
HORNER	14-28
LAGRANGE	14-36
LCM	14-38
LEGENDRE	14-39
PARTFRAC	14-45
PSI	14-48
QUOT	14-49
REMAINDER	14-51

Arithmetic Modulo commands

ADDTMOD	14-7
DIVMOD	14-15
DIV2MOD	14-14
EXPANDMOD	14-20
FACTORMOD	14-21
GCDMOD	14-26
INVMOD	14-34
MODSTO	14-43
MULTMOD	14-44
POWMOD	14-46
SUBTMOD	14-59

Calculus commands

Derivation and integration commands

CURL	14-11
DERIV	14-12
DERVX	14-12
DIV	14-13
FOURIER	14-23
HESS	14-27
IBP	14-30
INTVX	14-34
LAPL	14-37
PREVAL	14-47
RESULTANT	14-52

Limits and series commands

DIVPC	14-16
LIMIT	14-40
SERIES	14-54
TAYLOR0	14-62

Differential equations commands

DESOLVE	14-13
ILAP	14-33
LAP	14-37
LDEC	14-39

Exp and Lin commands

EXPLN	14-20
LIN	14-41
LNCOLLECT	14-42
TEXPAND	14-63
TSIMP	14-66

Matrix-related commands

Create

HILBERT	14-28
VANDERMONDE	14-67

Operations

AXL	14-9
AXM	14-9
TRAN	14-64
MAD	14-43
HADAMARD	14-26

Quadratic form

AXQ	14-10
GAUSS	14-24
QXA	14-50
SYLVESTER	14-59

Linear systems

LINSOLVE	14-41
REF	14-51
RREF	14-53

Eigenvector

JORDAN	14-36
PCAR	14-46

Symbolic solve commands

DESOLVE	14-13
LDEC	14-39
LINSOLVE	14-41
SOLVEVX	14-58
SOLVE	14-57
ZEROS	14-69

Trigonometry commands

Alphabetical command list

The following pages contain the commands in alphabetical order. See "Computer algebra command categories" on page 14-3 to view the commands in the order that they appear on the menus.

ABCUV

Type: Command

Description: Returns a solution in polynomials u and v of $au+bv=c$ where a and b are polynomials, and c is a value.

Access: Arithmetic, ↰ ARITH POLYNOMIAL

Input: Level 3/Argument 1: The polynomial corresponding to a.
Level 2/Argument 2: The polynomial corresponding to b.
Level 1/Argument 3: The value corresponding to c.

Output: Level 2/Item 1: The solution corresponding to u.
Level 1/Item 2: The solution corresponding to v.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Find a solution in polynomials u and v for the following equation:

$$(x^2+x+1)u+(x^2+4)v = 13$$

Command: `ABCUV(X^2+X+1,X^2+4,13)`

Result: `{-(X+3),X+4}`

See also: IABCUV
EGCD

ACOS2S

Type: Command

Description: Transforms an expression by replacing acos(x) in sub expressions with π/2–asin(x).

Access: Trigonometry, ↱ TRIG

Input: The expression to transform.

Output: The transformed expression.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Simplify the following expression:

$$arc\cos\left(\frac{2}{3}\right)+arc\cos(x)$$

Command: `ACOS2S(ACOS(2/3)+ACOS(X))`

Result: `π/2-ASIN(2/3)+π/2-ASIN(X)`

See also: ASIN2C
ASIN2T
ATAN2S

ADDTMOD

Type: Function

Description: Adds two expressions or values, modulo the current modulus.

Access: Arithmetic, ↰ ARITH MODULO

Input: Level 2/Argument 1: The first expression.
Level 1/Argument 2: The second expression.

Output: The sum of the two expressions, modulo the current modulus.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Express the result of the following addition in modulo 7.
$(x^2+3x+6)+(9x+3)$
Note: Use the CAS modes input form to set the modulo to 7

Command: `ADDTMOD(X^2+3*X+6,9*X+3)`

Result: `X^2-2*X+2`

ASIN2C

Type: Command

Description: Transforms an expression by replacing asin(x) sub expressions with π/2–acos(x) sub expressions.

Access: Trigonometry, ⮣ TRIG

Input: An expression.

Output: The transformed expression.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

ASIN2T

Type: Command

Description: Transforms an expression by replacing asin(x) sub expressions with the following:

$$\text{atan}\left(\frac{x}{\sqrt{1-x^2}}\right)$$

Access: Trigonometry, ⮣ TRIG

Input: An expression.

Output: The transformed expression.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

ATAN2S

Type: Command

Description: Transforms an expression by replacing atan(x) sub expressions with the following:

$$\text{asin}\left(\frac{x}{\sqrt{x^2+1}}\right)$$

Access: Trigonometry, ⮣ TRIG

Input: An expression.

Output: The transformed expression.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

AXL

Type: Command

Description: Converts a list to an array, or an array to a list.

Access: Convert, ↰ CONVERT

Input: A list or an array.

Output: If the input is a list, returns the corresponding array. If the input is an array, returns the corresponding list.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Convert the following matrix to a list:

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Command: `AXL([[0,1][1,0]])`

Result: `{{1,0},{0,1}}`

See also: AXM

AXM

Type: Command

Description: Converts a numeric array to a symbolic matrix.

Access: Matrices, ↰ MATRICES OPERATIONS

Input: An array.

Output: The corresponding matrix.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

See also: AXL
AXQ

AXQ

Type: Command

Description: Converts a square matrix into the associated quadratic form.

Access: Convert, ↰ CONVERT

Input: Level 2/Argument 1: An $n\times n$ matrix.
Level 1/Argument 2: A vector containing n variables.

Output: Level 2/Item 1: The corresponding quadratic form.
Level 1/Item 2: The vector containing the variables.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Find the quadratic form, expressed in terms of x, y, and z, associated with the following matrix:

$$\begin{bmatrix} 3 & 6 & 0 \\ 2 & 4 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Command:

```
AXQ([[3,6,0][2,4,1][1,1,1]],[X,Y,Z])
```

Result:

```
{3*X^2+(8*Y+Z)*X+(4*Y^2+2*Z*Y+Z^2),[X,Y,
Z]}
```

See also: GAUSS
QXA

CASCFG

Type: Command

Description: Restores the default CAS mode settings. This command is equivalent to pressing RESET on the CAS Modes input form.

Access: CAT CASCFG

CHINREM

Type: Command

Description: Solves a system of simultaneous polynomial congruences in the ring Z[x].

Access: Arithmetic, ↰ ARITH POLYNOMIAL

Input: Level 2/Argument 1: A vector of the first congruence (expression and modulus).
Level 1/Argument 2: A vector of the second congruence (expression and modulus).

Output: A vector of the solution congruence (expression and modulus).

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Solve the following simultaneous congruences for the polynomial u:

$u \equiv x^2 + 1 \pmod{x+2}$

$u \equiv x - 1 \pmod{x+3}$

Command: `CHINREM([X^2+1,X+2],[X-1,X+3])`

Result: `[X^3+2*X^2+5,-(X^2+5*X+6)]`

See also: ICHINREM

CURL

Type: Function

Description: Returns the curl of a three-dimensional vector function.

Access: Calculus, ↰ CALC DERIV AND INTEG

Input: Level 2/Argument 1: A three-dimensional vector function of three variables.
Level 1/Argument 2: An array comprising the three variables.

Output: The curl of the vector function with respect to the specified variables.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Find the curl of the following vector function:

$(v = x^2 y\underset{\sim}{i} + x^2 y\underset{\sim}{j} + y^2 z\underset{\sim}{k})'$

Command:

```
CURL([X^2*Y, X^2*Y, Y^2*Z],[X,Y,Z])
```

Result:

```
[2*X*Y,X^2,Y^2]
```

See Also: DIV
HESS

DERIV

Type: Function

Description: Returns the partial derivatives of a function, with respect to the specified variables.

Access: Calculus, ⬑ CALC DERIV. & INTEG

Input: Level 2/Argument 1: A function or a list of functions.
Level 1/Argument 2: A variable, or a vector of variables.

Output: The derivative, or a vector of the derivatives, of the function or functions.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Find the gradient of the following function of the spatial variables *x*, *y*, and *z*:

$2x^2y + 3y^2z + zx$

Command:

```
DERIV(2*X^2*Y+3*Y^2*Z+Z*X, [X,Y,Z])
EXPAND(ANS(1))
```

Result:

```
[4*Y*X+Z,2*X^2+6*Z*Y,X+3*Y^2]
```

See also: DERVX

DERVX

Type: Function

Description: Returns the derivative of a function with respect to the current variable.

Access: Calculus, ⬑ CALC DERIV. & INTEG.

Input: The function or list of functions to be differentiated.

Output: The derivative, or a vector of the derivatives, of the function or functions.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

See also: DERIV

DESOLVE

Type: Command

Description: Solves certain first-order ordinary differential equations with respect to the current variable.

Access: Symbolic solve, ↰ S.SLV

Input: Level 2/Argument 1: A first-order differential equation.
Level 1/Argument 2: The function to solve for.

Output: The solution to the equation, either y as a function of x or x as a function of y, or x and y as functions of a parameter.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Solve the following differential equation:

$$y'(x) + 2y(x) = e^{3x}$$

Command: `DESOLVE(d1Y(X)+2*Y(X)=EXP(3*X),Y(X))`

Result: `{Y(X)=(1/5*EXP(5*X)+C0*)(1/EXP(X)^2)}`

See also: LDEC

DIV

Type: Command

Description: Returns the divergence of a vector function.

Access: Calculus, ↰ CALC DERIV. & INTEG.

Input: Level 2/Argument 1: An array representing a vector function.
Level 1/Argument 2: An array containing the variables.

Output: The divergence of the vector function with respect to the specified variables.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Find the divergence of the following vector function:

$$v = x^2 y\underset{\sim}{i} + x^2 y\underset{\sim}{j} + y^2 z\underset{\sim}{k}$$

Command: `DIV([X^2*Y, X^2*Y, Y^2*Z],[X,Y,Z])`

Result: `Y*2*X+X^2+Y^2`

See also: CURL, HESS

DIV2

Type: Command

Description: Performs euclidean division on two expressions. Step-by-step mode is available with this command.

Access: Arithmetic, ↰ ARITH POLYNOMIAL

Input: Level 2/Argument 1: The dividend.
Level 1/Argument 2: The divisor.

Output: Level 2/Item 1: The quotient.
Level 1/Item 2: The remainder.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Perform the following division:

$$\frac{x^2 + x + 1}{2x + 4}$$

Command: `DIV2(X^2+X+1,2*X+4)`

Result: `{1/2(X-1),3}`

DIV2MOD

Type: Command

Description: Performs euclidean division on two expressions modulo the current modulus.

Access: Arithmetic, ↰ ARITH MODULO

Input: Level 2/Argument 1: The dividend.
Level 1/Argument 2: The divisor.

Output: Level 2/Item 1: The quotient.
Level 1/Item 2: The remainder.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Find the result of $\frac{x^3+4}{x^2-1}$, modulo the default modulus, 3.

Command: `DIV2MOD(X^3+4,X^2-1)`

Result: `{X X+1}`

DIVIS

Type: Command

Description: Returns a list of divisors of a polynomial or an integer.

Access: Arithmetic, ↰ ARITH

Input: A polynomial or an integer.

Output: A list containing the expressions or integers that exactly divide into the input.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Find the divisors of the following polynomial:

$x^2 + 3x + 2$

Command: `DIVIS(X^2+3*X+2)`

Result: `{1,X+1,X+2,X^2+3*X+2}`

See also: `DIV2`

DIVMOD

Type: Function

Description: Divides two expressions modulo the current modulus.

Access: Arithmetic, ↰ ARITH MODULO

Input: Level 2/Argument 1: The dividend.
Level 1/Argument 2: The divisor.

Output: The quotient of the terms modulo the current modulus.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Modulo the default modulus, 3, divide 5x^2+4x+2 by x^2+1.

Command: `DIVMOD(5*X^2+4*X+2,X^2+1)`

Result: `-((X^2-X+1)/X^2+1))`

DIVPC

Type: Command

Description: Returns a Taylor polynomial for the quotient of two expressions.

Access: Calculus, ⬑ CALC LIMITS & SERIES

Input: Level 3/Argument 1: The numerator expression.
Level 2/Argument 2: The denominator expression.
Level 1/Argument 3: The degree of the Taylor polynomial.

Output: The Taylor polynomial at $x = 0$ of the quotient of the two expressions, to the specified degree.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Find the fourth degree Taylor polynomial for the following:

$$\frac{x^3 + 4x + 12}{11x^{11} + 1}$$

Command: `DIVPC(X^3+4*X+12,11*X^11+1,4)`

Result: `12+4*X+X^3`

EGCD

Type: Command

Description: Given two polynomials, a and b, returns polynomials u, v, and c where:
$au+bv=c$
In the equation, c is the greatest common divisor of a and b.

Access: Arithmetic, ↰ ARITH POLYNOMIAL

Input: Level 2/Argument 1: The expression corresponding to *a* in the equation.
Level 1/Argument 2: The expression corresponding to *b* in the equation.

Output: Level 3/Item 1: The result corresponding to *c* in the equation.
Level 2/Item 2: The result corresponding to *u* in the equation.
Level 1/Item 3: The result corresponding to *v* in the equation.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Find the polynomials for *u*, *v*, and *c*, where *c* is the greatest common divisor of *a* and *b* such that:

$$u(x^2+1)+v(x-1) = c$$

Command: `EGCD(X^2+1,X-1)`

Result: `{2,1,-(X+1)}`

See also: IEGCD
ABCUV

EPSX0

Type: Function

Description: Replaces all coefficients in a polynomial that have an absolute value less than that held in the variable EPS, with 0. The value in EPS must be less than 1.

Access: Catalog, CAT

Input: A polynomial.

Output: The polynomial with conforming coefficients replaced with 0.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

EULER

Type: Function

Description: For a given integer, returns the number of integers less than the integer that are co-prime with the integer. (Euler's Φ function.)

Access: ↰ ARITH INTEGER

Input: A non-negative integer.

Output: The number of positive integers, less than, and co-prime with, the integer.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

EXLR

Type: Command

Description: Returns the left- and right-hand sides of an equation as discrete expressions.

Access: Catalog, CAT

Input: An equation.

Output: Level 2/Argument 1: The expression to the left of the "=" sign in the original equation.
Level 1/Argument 2: The expression to the right of the "=" sign in the original equation

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Split the following equation into its two component expressions:
sin(x)=5x+y

Command: `EXLR(SIN(X)=5*X+Y)`

Result: `{SIN(X) 5*X+Y}`

EXPAN

Type: Command

Description: Expands and simplifies an algebraic expression. This command is identical to the EXPAND command. It is included to ensure backward-compatibility with the HP 48-series calculators.

Access: Catalog, CAT

Input: An expression

Output: The expanded and simplified expression.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

See also: EXPAND

EXPAND

Type: Command

Description: Expands and simplifies an algebraic expression.

Access: Algebra, ↱ ALG

Input: An expression, or an array of expressions.

Output: The expanded and simplified expression or array of expressions.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Simplify the following expression:

$$\frac{(x^2 + 2x + 1)}{x + 1}$$

Command: `EXPAN((X^2+2*X+1)/(X+1))`

Result: `X+1`

See also: EXPAN

EXPANDMOD

Type: Function

Description: Expands and simplifies an algebraic expression, modulo the current modulus.

Access: ↰ ARITH MODULO

Input: An expression.

Output: The expanded and simplified expression modulo the current modulus.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Expand the following expression and give the result modulo 3 (the default modulo setting):

$(x+3)(x+4)$

Command: `EXPANDMOD((X+3)*(X+4))`

Result: `X^2+X`

EXPLN

Type: Command

Description: Transforms the trigonometric terms in an expression to exponential and logarithmic terms.

Access: Convert, ↰ CONVERT

Input: An expression

Output: The transformed expression.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).
Complex mode must be set (flag –103 set).

Example: Transform the following expression and simplify the result using the EXPAND command:

$.64\cos(x^2$

Command:
```
EXPLN(2*COS(X^2))
EXPAND(ANS(1))
```

Result: $\frac{\mathrm{EXP}(iX^2)^2+1}{\mathrm{EXP}(iX^2)}$

See also: SINCOS

FACTOR

Type: Command

Description: Factorizes a polynomial or an integer:

- The function expresses a polynomial as the product of irreducible polynomials.
- The function expresses an integer as the product of prime numbers.

Access: Algebra, ↱ ALG

Input: An expression or an integer.

Output: The factorized expression, or the integer expressed as the product of prime numbers.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Factorize the following:

$x^2 + 5x + 6$

Command: `FACTOR(X^2+5*X+6)`

Result: `(X+2)(X+3)`

See also: EXPAND

FACTORMOD

Type: Function

Description: Factorizes a polynomial modulo the current modulus. The modulus must be less than 100, and a prime number.

Access: Arithmetic, ↰ ARITH MODULO

Input: The expression to be factorized.

Output: The factorized expression modulo the current modulus.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Factorize the following expression modulo the default modulus, 3.

x^2+2

Command: `FACTORMOD(X^2+2)`

Result: `(X+1)*(X-1)`

FACTORS

Type: Command

Description: For a value or expression, returns a list of prime factors and their multiplicities.

Access: Arithmetic, ↰ ARITH

Input: A value or expression.

Output: A list of prime factors of the value or expression, with each factor followed by its multiplicity.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example 1: Find the prime factors of 100.

Command: `FACTORS(100)`

Result: `{5 2. 2 2.}`

Example 2: Find the irreducible factors of: $x^2 + 4x + 4$

Command: `FACTORS(X^2+4*X+4)`

Result: `{X+2,2.}`

FCOEF

Type: Command

Description: From an array of roots and multiplicities/poles, returns a rational polynomial with a leading coefficient of 1, with the specified set of roots or poles, and with the specified multiplicities.

Access: Arithmetic, ↰ ARITH POLYNOMIAL

Input: An array of the form [Root 1, multiplicity/pole 1, Root 2, Multiplicity/pole 2, . .] The multiplicity/pole must be an integer. A positive number signifies a multiplicity. A negative number signifies a pole.

Output: The rational polynomial with the specified roots and multiplicities/poles.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Find the rational polynomial corresponding to the following set of roots and poles:
1, 2, 3, –1

Command: `FCOEF([1,2,3,-1])`

Result: `(X-1)^2/(X-3)`

See also: FROOTS

FOURIER

Type: Function

Description: Returns the n^{th} coefficient of a complex Fourier series expansion. The `PERIOD` variable must be in the current path, and set to hold L, the period of the input function.

Access: Calculus ↰ (CALC) DERIV. & INTEG

Input: Level 1/Argument 2: An expression
Level 2/Argument 1: The number, n, of the coefficient to return.

Output: The nth Fourier coefficient of the expression.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).
Complex mode must be set, that is, flag –103 must be set.

FROOTS

Type: Command

Description: For a rational polynomial, returns an array of its roots and poles, with their corresponding multiplicities.

Access: Arithmetic, ↰ (ARITH) POLYNOMIAL

Input: A rational polynomial.

Output: An array of the form [Root 1, Multiplicity 1, Root 2, Multiplicity 2 . . .]
A negative multiplicity indicates a pole.

Flags: Exact mode must be set (flag –105 clear). Numeric mode must not be set (flag –03 clear).

If complex mode is set (flag –103 set), FROOTS looks for complex solutions as well as real solutions.
If approximate mode is set (flag –105 set) FROOTS searches for numeric roots.

See also: FCOEF

FXND

Type: Command

Description: Splits an object into a numerator and a denominator.

Access: Catalog, CAT

Input: A fraction, or an object that evaluates to a fraction.

Output: The object split into numerator and denominator.
Level 2/Item 1: The numerator.
Level 1/Item 2: The denominator.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Return the numerator and the denominator of the following expression:

$$\frac{(x-3)^2}{z+4}$$

Command: `FXND((X-3)^2/(Z+4))`

Result: Level 2/Item1: `(X-3)^2`
Level 1/Item 2: `Z+4`

GAUSS

Type: Command

Description: Returns the diagonal representation of a quadratic form.

Access: Matrices, ↰ MATRICES QUADRATIC FORM

Input: Level 2/Argument 1: The quadratic form.
Level 1/Argument 2: A vector containing the independent variables.

Output: Level 4/Argument 1: An array of the coefficients of the diagonal.
Level 3/Argument 2: A matrix, P, such that the quadratic form is represented as P^TDP, where the diagonal matrix

D contains the coefficients of the diagonal representation.
Level 2/Argument 3: The diagonal representation of the quadratic form.
Level 1/Argument 4: A list of the variables.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Find the Gaussian symbolic quadratic form of the following:

$x^2 + 2axy$

Command: `GAUSS(X^2+2*A*X*Y,[X,Y])`

Result:
```
{[1,-A^2] [[1,A][0,1]],-
(A^2*Y^2)+(A*Y+X)^2,[X,Y])
```

See also: AXQ
QXA

GCD

Type: Function

Description: Returns the greatest common divisor of two objects.

Access: Arithmetic, ↰ ARITH POLYNOMIAL

Input: Level 2/Argument 1: An expression, or an object that evaluates to a number.
Level 1/Argument 2: An expression, or an object that evaluates to a number.

Output: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Flags: For a symbolic result, clear the CAS modes Numeric option (flag –03 clear).

Example: Find the greatest common divisor of 2805 and 99.

Command: `GCD(2805,99)`

Result: `33`

See also: GCDMOD
EGCD
IEGCD
LCM

GCDMOD

Type: Function

Description: Finds the greatest common divisor of two polynomials modulo the current modulus.

Access: Arithmetic, ↰ ARITH MODULO

Input: Level 2/Argument 1: A polynomial expression.
Level 1/Argument 2: A polynomial expression.

Output: The greatest common divisor of the two expressions modulo the current modulus.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

See also: GCD

HADAMARD

Type: Command

Description: Performs an element by element multiplication of two matrices (Hadamard product).

Access: Matrices, ↰ MATRICES OPERATIONS

Input: Level 2/Argument 1: Matrix 1.
Level 1/Argument 2: Matrix 2.
The matrices must have the same order.

Output: The matrix representing the result of the multiplication.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Find the Hadamard product of the following two matrices:

$$\begin{bmatrix} 3 & -1 & 2 \\ 0 & 1 & 4 \end{bmatrix} \text{ and } \begin{bmatrix} 2 & 3 & 0 \\ 1 & 5 & 2 \end{bmatrix}$$

Command:

```
HADAMARD([[3,-
1,2][0,1,4]],[2,3,0][1,5,2]])
```

Result:

```
[[6,-3,-0][0,5,8]]
```

HALFTAN

Type: Command

Description: Transforms an expression by replacing sin(x), cos(x) and tan(x) sub expressions with tan(x/2) terms.

Access: Trigonometry, ↱ TRIG

Input: An expression

Output: The transformed expression.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

HERMITE

Type: Function

Description: Returns the nth Hermite polynomial.

Access: Arithmetic, ↰ ARITH POLYNOMIAL

Input: A non-negative integer.

Output: The corresponding polynomial expression.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Find the Hermite polynomial with degree 4.

Command: `HERMITE(4)`

Result: `16*X^4-48*X^2+12`

HESS

Type: Command

Description: Returns the Hessian matrix and the gradient of an expression with respect to the specified variables.

Access: Calculus ↰ CALC DERIV & INTEG

Input: Level 2/Argument 1: An expression.
Level 1/Argument 2: A vector of the variables.

Output: Level 3/Item 1: The Hessian matrix with respect to the specified variables.
Level 2/Item 2: The gradient with respect to the variables.
Level 1/Item 3: The vector of the variables.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

See also: CURL
DIV

HILBERT

Type: Command

Description: Returns a square Hilbert matrix of the specified order.

Access: Matrices, ↰ MATRICES CREATE

Input: A positive integer, representing the order.

Output: The Hilbert matrix of the specified order.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Find the order 3 Hilbert matrix.

Command: `HILBERT(3)`

Result:

$$\begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \end{bmatrix}$$

HORNER

Type: Command

Description: Executes a Horner scheme on a polynomial. That is, for a given polynomial *P*, and a number *r*, HORNER returns $P/(x–r)$, r and P(r)

Access: Arithmetic, ↰ ARITH POLYNOMIAL

Input: Level 2/Argument 1: A polynomial, *P*.
Level 1/Argument 2: A number, *r*.

Output: Level 3/Item 1: $P/(x–r)$
Level 2/Item 2: r
Level 1/item 3: $P(r)$, the remainder of the division process.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: For r = 3, find the result of executing a Horner scheme on the following polynomial:

$x^2 + x + 1$

Command: `HORNER(X^2+X+1,3)`

Results: `(X+4,3,13)`

I→R

Type: Command

Description: Converts an integer into a real number.

Access: Catalog, CAT

Input: Level 1/Argument 1: An integer.

Output: Level 1/Item 1: The integer converted to a real number.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

See also: R→I

IABCUV

Type: Command

Description: Returns a solution in integers u and v of $au + bv = c$, where a, b, and c are integers.

Access: Arithmetic, ↰ ARITH INTEGER

Input: Level 3/Argument 1: the value of a.
Level 2/Argument 2: the value of b.
Level 1/Argument 3: the value of c.

Output: Level 2/Item 1: The value for u.
Level 1/Item 2: The value for v.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Find a solution in integers of the equation:
$6a + 11b = 3$

Command: `IABCUV(6,11,3)`

Result: `{6,-3}`

See also: ABCUV
IEGCD

IBERNOULLI

Type: Function

Description: Returns the *n*th Bernoulli number for a given integer.

Access: Catalog, CAT

Input: Level 1/Argument 1: an integer.

Output: Level 1/Item 1: The corresponding nth Bernoulli number for the integer.

Flags: Numeric mode must not be set (flag –03 clear).

IBP

Type: Command

Description: Performs integration by parts on a function. The function must be able to be represented as a product of two functions, where the antiderivative of one of the functions is known:
f(x) = u(x).v'(x)

Note that the command is designed for use in RPN mode only.

Access: Calculus, ⬑ CALC DERIV & INTEG

Input: Level 2: The integrand expressed as a product of two functions, u(x).v'(x)
Level 1: The antiderivative of one of the component functions, v(x).

Output: Level 2: u(x)v(x)
Level 1: -u'(x)v(x)

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Use integration by parts to calculate the following:

$$\int x\cos(x)\,dx$$

Command 1: Apply the IBP command:

Level 2: `X*COS(X)`
Level 1: `SIN(X)`

Result: Level 2: `SIN(X).X`
Level 1: `-SIN(X)`

Command 2: Apply the INTVX command to level 1, `-SIN(X)`

Result: Level 2: `SIN(X).X`
Level 1: `COS(X)`

Command 3: Press ⊕ to add the result to the value at level 2 to obtain the final result.

Result: `SIN(X).(X)+COS(X)`

See also: INTVX
INT
PREVAL
RISCH

ICHINREM

Type: Command

Description: Solves a system of two congruences in integers using the Chinese Remainder theorem.

Access: Arithmetic, ↰ ARITH INTEGER

Input: Level 2/Argument 1: A vector of the first value and the modulus.
Level 1/Argument 2: A vector of the second value and the modulus.

Output: A vector of the solution.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Solve the following system of congruences:

$x \equiv 2$ Modulo 3

$x \equiv 1$ Modulo 5

Command: `ICHINREM([2,3],[1,5])`

Results: `[-4, 15]`

See also: CHINREM

IDIV2

Type: Command

Description: For two integers, *a* and *b*, returns the integer part of *a*/*b*, and the remainder, *r*.

Access: Arithmetic, ↰ ARITH INTEGER

Input: Level 2/Argument 1: *a*.
Level 1/Argument 2: *b*.

Output: Level 2/Item 1: The integer part of *a*/*b*.
Level 1/Item 2: The remainder.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Return the integer part and the remainder of 11632/864.

Command: `IDIV2(11632,864)`

Result: `{13,400}`

See also: DIV2

IEGCD

Type: Command

Description: Given two integers *x* and *y*, returns three integers, *a*, *b*, and *c*, such that:
$ax+by=c$

Access: Arithmetic, ↰ ARITH INTEGER

Input: Level 2/Argument 1: *x*.
Level 1/Argument 2: *y*.

Output: Level 3/Item 1: *c*.
Level 2/Item 2: *a*.
Level 1/Item 3: *b*.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

See also: EGCD

ILAP

Type: Function

Description: Returns the inverse Laplace transform of an expression. The expression must evaluate to a rational fraction.

Access: Calculus, ↰ CALC DIFFERENTIAL EQNS

Input: A rational expression.

Output: The inverse Laplace transformation of the expression.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Find the inverse Laplace transform of:

$$\frac{1}{(x-5)^2}$$

Command: `ILAP(1/(X-5)^2)`

Result: `X*EXP(5*X)`

INT

Type: Function

Description: Calculates the antiderivative of a function for a given variable, at a given point.

Access: Catalog, CAT

Input: Level 3/Item 1: A function.
Level 2/Item 2: The variable to obtain the derivative with respect to.
Level 1/Item 3: The point at which to calculate the antiderivative. This point can be a variable or an expression.

Output: The antiderivative of the function for the given variable, at the point you specified.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

See also: INTVX
RISCH

INTVX

Type: Function

Description: Finds the antiderivative of a function symbolically, with respect to the current default variable.

Access: Calculus, ↰ CALC DERIV. & INTEG

Input: An expression.

Output: The antiderivative of the expression.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Find the antiderivative of the following:

$x^2 \ln x$

Command: `INTVX(X^2*LN(X))`

Result: `1/3*X^3*LN(X)+(-1/9)X^3`

See also: IBP
RISCH
PREVAL

INVMOD

Type: Function

Description: Performs modular inversion on an object modulo the current modulus.

Access: Arithmetic, ↰ ARITH MODULO

Input: An object.

Output: The modular inverse of the object.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Solve the following for x, modulo the default modulus, 3.

$(2x \equiv 1)$

Command: `INVMOD(2)`

Result: `-1`

IQUOT

Type: Function

Description: Returns the integer quotient of two integers. That is, given two integers, a and b, returns the integer q, such that:

$a = qb + r$, and $0 \leq r < b$

Access: Arithmetic, ↰ ARITH INTEGER

Input: Level 2/Item 1: The dividend.
Level 1/Item 2: The divisor.

Output: The integer quotient.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

See also: QUOT
IDIV2

IREMAINDER

Type: Function

Description: Returns the remainder of an integer division.

Access: Catalog, CAT

Input: Level 2/Argument 1: The numerator.
Level 1/Argument 2: The denominator.

Output: The remainder.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

See also: IDIV2

ISPRIME?

Type: Function

Description: Tests if a number is prime.

Access: Arithmetic, ↰ ARITH INTEGER

Input: An object that evaluates to a number.

Output: 1 (True) if the number is prime, 0 (False) if it is not.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

See also: NEXTPRIME
PREVPRIME

JORDAN

Type: Command

Description: Computes the eigenvalues, eigenvectors, minimum polynomial, and characteristic polynomial of a matrix.

Access: Matrices, ↰ MATRICES EIGENVECTORS

Input: An $n \times n$ matrix.

Output: Level 4/Item 1: The minimum polynomial.
Level 3/Item 2: The characteristic polynomial.
Level 2/Item 3: A list of characteristic spaces tagged by the corresponding eigenvalue (either a vector or a list of Jordan chains, each of them ending with an "Eigen:"-tagged eigenvector).
Level 1/Item 4: An array of the eigenvalues, with multiplicities

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

LAGRANGE

Type: Command

Description: Returns the interpolating polynomial of minimum degree for a pair of values.

Access: Arithmetic, ↰ ARITH POLYNOMIAL

Input: A two × n matrix of the n pairs of values.

Output: The polynomial that results from the Lagrange interpolation of the data.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Find an interpolating polynomial for the data (1,6), (3,7), (4,8), (2,9)

Command: `LAGRANGE(` $\begin{bmatrix} 1 & 3 & 4 & 2 \\ 6 & 7 & 8 & 9 \end{bmatrix}$ `)`

Result: $\dfrac{8x^3 - 63x^2 + 151x - 60}{6}$

LAP

Type: Function

Description: Performs a Laplace transform on an expression with respect to the current default variable.

Access: Calculus, ↰ CALC DIFFERENTIAL EQNS

Input: An expression.

Output: The Laplace transform of the expression.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Find the Laplace transform of e^x.

Command: `LAP(EXP(X))`

Result: `1/(X-1)`

LAPL

Type: Command

Description: Returns the Laplacian of a function with respect to a list of variables.

Access: ↰ CALC DERIV & INTEG

Input: Level 2/Argument 1:An expression.
Level 1/Argument 2: A vector of variables.

Output: The Laplacian of the expression with respect to the variables.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Find, and simplify, the Laplacian of the following expression:

$e^x \cos(zy)$

Command: `LAPL(EXP(X)*COS(Z*Y),[X,Y,Z])`
`EXPAND(ANS(1))`

Result: `-((Y^2+Z^2-1)*EXP(X)*COS(Z*Y))`

LCM

Type: Function

Description: Returns the least common multiple of two objects.

Access: Arithmetic, ↰ ARITH POLYNOMIAL

Input: Level 2/Argument 1: An expression, a number, or object that evaluates to a number.
Level 1/Argument 2: An expression, a number, or object that evaluates to a number.

Output: The least common multiple of the objects.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Find the least common multiple of the following two expressions:

$x^2 - 1$

$x{-}1$

Command: `LCM(X^2-1,X-1)`

Results: `X^2-1`

See also: GCD

LCXM

Type: Command

Description: From a program with two arguments, builds a matrix with the specified number of rows and columns, with $a_{ij} = f(i,j)$.

Access: Catalog, CAT

Input: Level 3/Argument 1: The number of rows you want in the resulting matrix.
Level 2/Argument 2: The number of columns you want in the resulting matrix.
Level 1/Argument 3: A program that uses two arguments.

Output: The resulting matrix.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Build a 2 × 3 matrix with a_{ij}=i+2j.

Command: `LCXM(2,3,<<→I J 'I+2*J'>>)`

Result:

$$\begin{bmatrix} 3 & 5 & 7 \\ 4 & 6 & 8 \end{bmatrix}$$

LDEC

Type: Command

Description: Solves a linear differential equation with constant coefficients, or a system of first order linear differential equations with constant coefficients.

Access: Symbolic solve, ↰ (S.SLV)

Input: Level 2/Argument 1: For a single equation, the function forming the right hand side of the equation. For a system of equations, an array comprising the terms not containing the dependent variables.
Level 1/Argument 2: For one equation, the auxiliary polynomial. For a system of equations, the matrix of coefficients of the dependent variables.

Output: The solution.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

LEGENDRE

Type: Function

Description: Returns the *n*th degree Legendre polynomial.

Access: Arithmetic, ↰ (ARITH) POLYNOMIAL

Input: An integer, *n*.

Output: The *n*th Legendre polynomial.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Find the Legendre polynomial with degree 4.

Command: `LEGENDRE(4)`

Result: `(35*X^4-30*X^2+3)/8`

LGCD

Type: Function

Description: Returns the greatest common divisor of a list of expressions or values.

Access: Arithmetic, ↰ ARITH

Input: A list of expressions or values.

Output: Level 2/Item 1: The list of elements.
Level 1/Item 2: The greatest common divisor of the elements.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

LIMIT

Type: Function

Description: Returns the limit of a function as it approaches a specified value.

Access: Calculus, ↰ CALC LIMITS&SERIES

Input: Level 2/Argument 1: An expression.

Level 1/Argument 2: An expression of the form $x = y$, where x is the variable and y is the value at which the limit is to be evaluated.

Output: The limit of the expression at the limit point.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Find the following limit:

$$\left(\lim_{x \rightarrow y}\right)\frac{x^n - y^n}{x - y}$$

Command: `LIMIT((X^N-Y^N)/(X-Y), X=Y)`

Result: `N*EXP(N*LN(Y))/Y`

See also: SERIES

LIN

Type: Command

Description: Linearizes expressions involving exponential terms.

Access: Exponential and logarithm, ↰ EXP&LN

Input: An expression.

Output: The linearized expression.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Linearize the following expression:

$x(e^x e^y)^4$

Command: `LIN(X*(EXP(X)*EXP(Y))^4)`

Result: `X*EXP(4X+4Y)`

LINSOLVE

Type: Command

Description: Solves a system of linear equations.

Access: Symbolic solve, ↰ S.SLV

Input: Level 2/Argument 1: An array of equations.
Level1/Argument 2: A vector of the variables to solve for.

Output: Level 3/Item 1: The system of equations.
Level 2/Item 2: A list of the pivot points.
Level 1/Item 3: The solution.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

LNAME

Type: Command

Description: Returns the variable names contained in a symbolic expression.

Access: Catalog, CAT

Input: A symbolic expression.

Output: A vector containing the variable names.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

See also: LVAR

LNCOLLECT

Type: Command

Description: Simplifies an expression by collecting logarithmic terms.

Access: Algebra, ↱ ALG

Input: An expression.

Output: The simplified expression.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Simplify the following expression:
2(ln(*x*)+ln(*y*))

Command: `LNCOLLECT(2(LN(X)+LN(Y))`

Result: `LN(X^2+Y)`

LVAR

Type: Command

Description: Returns a list of variables in an algebraic object.

Access: Catalog, CAT

Input: An algebraic object.

Output: Level 2/Item 1: The algebraic object.
Level 1/Item 2: A vector of the variables that the object contains.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

MAD

Type: Command

Description: Returns details of a square matrix.

Access: Matrices, ↰ MATRICES OPERATIONS

Input: A square matrix

Output: Level 4/Item 1: The determinant.
Level 3/Item 2: The formal inverse.
Level 2/Item 3: The matrix coefficients of the polynomial, p, defined by $(xi-a)p(x)=m(x)i$, where a is the matrix, and m is the characteristic polynomial of a.
Level 1/Item 4: The characteristic polynomial.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

MENUXY

Type: Command

Description: Displays a function key menu of the computer algebra commands in the specified range.

Access: Catalog, CAT

Input: Level 2/Argument 1: The number of the first command in the range that you want to display.
Level 1/Argument 2: The number of the last command in the range that you want to display.

Output: On the function key menu, a list of the computer algebra commands in the range that you specified.

MODSTO

Type: Command

Description: Changes the modulo setting to the specified number. The number that you set is reflected in the CAS Modes input form.

Access: Arithmetic, ↰ ARITH MODULO

Input: The modulo value that you want to set.

Output: The modulo setting is changed to the specified number.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

MULTMOD

Type: Function

Description: Performs modular multiplication of two objects, modulo the current modulus.

Access: Arithmetic, ↰ ARITH MODULO

Input: Level 2/Argument 1: A number or an expression.
Level 1/Argument 2: A number or an expression.

Output: The result of modular multiplication of the two objects, modulo the current modulus.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Find the product of $2x$ and $38x^2$, modulo the default modulus, 3.

Command: `MULTMOD(2*X,38*X^2)`

Result: `X^3`

NEXTPRIME

Type: Function

Description: Given an integer, returns the next prime number larger than the integer.

Access: Arithmetic, ↰ ARITH INTEGER

Input: An integer.

Output: The next prime number larger than the integer.

Example: Find the closest, larger prime number to 145.

Command: `NEXTPRIME(145)`

Result: `149`

See also: ISPRIME?
PREVPRIME

PA2B2

Type: Command

Description: Takes a prime number, p, such that p=2 or $p \equiv 1$ modulo 4, and returns a Gaussian integer $a + ib$ such that $p = a^2 + b^2$. This function is useful for factorizing Gaussian integers.

Access: Arithmetic, ↰ ARITH INTEGER

Input: A prime number, p, such that p=2 or $p \equiv 1$ modulo 4

Output: A Gaussian integer a+ib such that p=a^2+b^2

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

See also: GAUSS

PARTFRAC

Type: Command

Description: Performs partial fraction decomposition on a partial fraction.

Access: Arithmetic, ↰ ARITH POLYNOMIAL

Input: An algebraic expression.

Output: The partial fraction decomposition of the expression.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Perform a partial fraction decomposition of the following expression:

$$\frac{1}{x^2 - 1}$$

Command: `PARTFRAC(1/(X^2-1))`

Result: `(-1+(1/X+^2))`

PCAR

Type: Command

Description: Returns the characteristic polynomial of an $n \times n$ matrix.

Access: Matrices, ↰ MATRICES EIGENVECTORS

Input: A square matrix.

Output: The characteristic polynomial of the matrix.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Find the characteristic polynomial of the following matrix:

$$\begin{bmatrix} 5 & 8 & 16 \\ 4 & 1 & 8 \\ -4 & -4 & -11 \end{bmatrix}$$

Command: `PCAR([[5,8,16][4,1,8][-4,-4,-11]])`

Result: `X^3+5*X^2+3*X-9`

POWMOD

Type: Function

Description: Raises an object (number or expression) to the specified power, and expresses the result modulo the current modulus.

Access: Arithmetic, ↰ ARITH MODULO

Input: Level 2/Argument 1: The object.
Level 1/Argument 2: The exponent.

Output: The result of the object raised to the exponent, modulo the current modulus.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

PREVAL

Type: Function

Description: With respect to the current default variable, returns the difference between the values of a function at two specified values of the variable.

PREVAL can be used in conjunction with INTVX to evaluate definite integrals. See the example below.

Access: Calculus, ↰ CALC DERIV. & INTEG.

Input: Level 3/Argument 1: A function.
Level 2/Argument 2: The lower bound.
Level 3/Argument 1: The upper bound.
The bounds can be expressions.

Output: The result of the evaluation.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Evaluate the following:

$$\int_0^3 (x^3 + 3x)dx$$

Command: `PREVAL(INTVX(X^3+3*X),0,3)`

Result: `135/4`

PREVPRIME

Type: Function

Description: Given an integer, finds the closest prime number smaller than the integer.

Access: Arithmetic, ↰ ARITH INTEGER

Input: An integer.

Output: The closest prime number smaller than the integer.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Find the closest, smaller prime number to 145.

Command: `PREVPRIME(145)`

Result: `139`

See also: ISPRIME?
NEXTPRIME

PROPFRAC

Type: Command

Description: Splits an improper fraction into an integer part and a fraction part.

Access: Arithmetic, ↰ ARITH

Input: An improper fraction, or an object that evaluates to an improper fraction.

Output: A proper fraction.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Express the following as a proper fraction:

$$\frac{x^3+4}{x^2}$$

Command: `PROPFRAC((X^3+4)/X^2))`

Result: `X+(4/X^2)`

PSI

Type: Function

Description: Calculates the polygamma function in one point.

Access: Catalog, CAT

Input: A complex expression.

Output: The polygamma function.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Psi

Type: Function

Description: Calculates the digamma function in one point. The digamma function is the derivative of the natural logarithm (ln) of the gamma function. The function can be represented as follows:

$$\Psi(z) = \frac{d}{dz}(\ln\Gamma(z)) = \frac{\Gamma'(z)}{\Gamma(z)}$$

Access: Catalog, CAT

Input: Level 2/Argument 1: A complex expression
Level 1/Argument 2: A non-negative integer.

Output: The digamma function at the specified point.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

PTAYL

Type: Function

Description: Returns the Taylor polynomial at $x = a$ for a specified polynomial.

Access: Arithmetic, ↰ ARITH POLYNOMIAL

Input: Level 2/Argument 1: A polynomial, P.
Level 1/Argument 2: A number, a.

Output: A polynomial, Q such that Q(x – a)=P(x).

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

QUOT

Type: Function

Description: Returns the quotient part of the Euclidean division of two polynomials.

Access: Arithmetic, ↰ ARITH POLYNOMIAL

Input: Level 2/Argument 1: The numerator polynomial.
Level 1/Argument 2: The denominator polynomial.

Output: The quotient of the Euclidean division.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Find the quotient of the division of $x^3 + 6x^2 + 11x + 6$ by $x^2 + 5x + 6$.

Command: `QUOT(X^3+6*X^2+11*X+6, X^2+5*X+6)`

Result: `X+1`

See also: REMAINDER

QXA

Type: Command

Description: Expresses a quadratic form in matrix form.

Access: Catalog, CAT

Input: Level 2/Argument 1: A quadratic form.
Level 1/Argument 2: A vector containing the variables.

Output: Level 2/Item 1: The quadratic form expressed in matrix form.
Level 1/Item 2: A vector containing the variables.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Express the following quadratic form in matrix form: $x^2 + xy + y^2$

Command: `QXA(X^2+X*Y+Y^2, [X,Y])`

Result: `{[[1,1/2][1/2,1]],[X,Y]}`

See also: AXQ
GAUSS

R→I

Type: Command

Description: Converts a real number to an integer.

Access: Catalog, CAT

Input: Level 1/Argument 1: A real number.

Output: Level 1/Item 1: The real number converted to an integer.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

See also: I→R

REF

Type: Command

Description: Reduces a matrix to echelon form.

Access: Matrices, ↰ MATRICES LINEAR SYSTEMS

Input: A matrix.

Output: The equivalent matrix in echelon form.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

See also: RREF

REMAINDER

Type: Function

Description: Returns the remainder of the Euclidean division of two polynomials.

Access: Arithmetic, ↰ ARITH POLYNOMIAL

Input: Level 2/Argument 1: The numerator polynomial.
Level 1/Argument 2: The denominator polynomial.

Output: The remainder resulting from the Euclidean division.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

See also: QUOT

REORDER

Type: Function

Description: Given a polynomial expression and a variable, reorders the variables in the expression in the order of powers set on the CAS Modes screen, that is, either in increasing or decreasing order.

Access: Catalog, CAT

Input: Level 2/Argument 1: The polynomial expression.
Level 1/Argument 2: The variable with respect to which the reordering is performed.

Output: The reordered expression.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

RESULTANT

Type: Function

Description: Returns the resultant of two polynomials of the current variable. That is, it returns the determinant of the Sylvester matrices of the two polynomials.

Access: Catalog, CAT

Input: Level 2/Argument 1: The first polynomial.
Level 1/Argument 2: The second polynomial.

Output: The determinant of the two matrices that correspond to the polynomials.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

RISCH

Type: Function

Description: Performs symbolic integration on a function using the Risch algorithm. RISCH is similar to the INTVX command, except that it allows you to specify the variable of integration.

Access: Calculus ↰ CALC DERIV. & INTEG.

Input: Level 2/Argument 1: The function to integrate.
Level 1/Argument 2: The variable of integration.

Output: The antiderivative of the function with respect to the variable.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Find the antiderivative of the following function, with respect to y:

$y^2 + 3y + 2$

Command: `RISCH(Y^3-3*Y+2,Y)`

Result: `1/3*Y^3+3/2*Y^2+2*Y`

See also: IBP
INT
INTVX

RREF

Type: Command

Description: Reduces a matrix to row-reduced echelon form.

Access: Matrices, ↰ (MATRICES) LINEAR SYSTEMS

Input: A matrix.

Output: An equivalent matrix in row reduced echelon form.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

rref

Type: Command

Description: Reduces a matrix to row-reduced echelon form.

Access: Matrices, ↰ (MATRICES) LINEAR SYSTEMS

Input: A matrix.

Output: Level 2/Argument 1: The pivot points.

Level 1/Argument 2: An equivalent matrix in row reduced echelon form.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

RREFMOD

Type: Command

Description: Performs modular row-reduction to echelon form on a matrix, modulo the current modulus.

Access: Catalog, CAT

Input: A matrix.

Output: The modular row-reduced matrix. The modulo value is set using the Modes CAS input form.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

SERIES

Type: Command

Description: For a given function, computes Taylor series, asymptotic development and limit at finite and infinite points.

Access: Calculus, ↰ CALC LIMITS & SERIES

Input: Level 3/Argument 1: The function $f(x)$
Level 2/Argument 2: The variable if the limit point is 0, or an equation $x = a$ if the limit point is a.
Level 1/Argument 3: The order for the series expansion. Note the following points:

- The minimum value is 2, and the maximum value is 20.
- If the order is a positive or negative real number, the series is unidirectional.
- For bidirectional series expansions, you need to give the order as a binary integer, for example, #5d.

Output: Level 2/Item 1: A list containing the bidirectional limit, an expression approximating the function near the limit point, and the order of the remainder. These are expressed in terms of a small parameter h.
Level 1/Item 2: An expression for h in terms of the original variable.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

SEVAL

Type: Function

Description: In the given expression, evaluates any existing variables that the expression contains and substitutes these back into the expression.

Access: Catalog, CAT

Input: Level 1/Item 1: An algebraic expression.

Output: The expression with existing variables evaluated.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

SIGMA

Type: Function

Description: Calculates the discrete antiderivative of a function with respect to the variable that you define.

Access: Catalog, CAT

Input: Level 2/Item 1: A function
Level 1/Item 2: The variable to calculate the antiderivative with respect to.

Output: The antiderivative of the function.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

See also: SIGMAVX, RISCH

SIGMAVX

Type: Function

Description: Calculates the discrete antiderivative of a function with respect to the current variable.

Access: Catalog, CAT

Input: Level 2/Item 1: A function.

Output: The antiderivative of the function.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

See also: SIGMA, RISCH

SIGNTAB

Type: Command

Description: Tabulates the sign of a rational function of one variable.

Access: Catalog, CAT

Input: An algebraic expression.

Output: A list containing, the points where the expression changes sign, and for each point, the sign of the expression between the points.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

SIMP2

Type: Command

Description: Simplifies two objects by dividing them by their greatest common divisor.

Access: Arithmetic, ⬑ ARITH

Input: Level 2/Argument 1: The first object.
Level 1/Argument 2: The second object.

Output: Level 2/Item 1: The first object divided by the greatest common divisor.
Level 1/Item 2: The second object divided by the greatest common divisor.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Divide the following expressions by their greatest common divisor:

$$x^3 + 6x^2 + 11x + 6$$

$$x^3 - 7x - 6$$

Command: `SIMP2(X^3+6*X^2+11*X+6, X^3-7*X-6)`

Result: `{X+3,X-3}`

SINCOS

Type: Command

Description: Converts complex logarithmic and exponential expressions to expressions with trigonometric terms.

Access: Trigonometry, ↱ TRIG

Input: An expression with complex linear and exponential terms.

Output: The expression with logarithmic and exponential sub expressions converted to trigonometric and inverse trigonometric expressions.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).
Must be in complex mode (flag –103 set).

Example: Express e^{ix} in trigonometric terms.

Command: `SINCOS(EXP(i*X))`

Result: `COS(X)+iSIN(X)`

SOLVE

Type: Command

Description: Finds zeros of an expression equated to 0, or solves an equation.

Access: Symbolic solve, ↰ S.SLV

Input: Level 2/Argument 1: The expression or equation.
Level 1/Argument 2: The variable to solve for.

Output: A list of zeros or solutions.

Flags: If exact mode is set (flag –105 clear) and there are no exact solutions, the command returns a null list even when there are approximate solutions.

Example: Find the zeros of the following expression:

$x^3 - x - 9$

Command: `SOLVE(X^3-X-9,X)`

Result: `{X=2.24004098747}`

See also: SOLVEVX

SOLVEVX

Type: Command

Description: Finds zeros of an expression with respect to the current variable, or solves an equation with respect to the current variable. (You use the CAS modes input form to set the current variable.)

Access: Symbolic solve, ↰ S.SLV

Input: A function or equation in the current variable.

Output: A list of zeros or solutions.

Flags: For a symbolic result, clear the CAS modes Numeric option (flag –03 clear).
If Exact mode is set (flag –105 clear) and there are no exact solutions, the command returns a null list even when there are approximate solutions.

Example: Solve the following expression for 0, where X is the default variable on the calculator:

$x^3 - x - 9$

Command: `SOLVEVX(X^3-X-9)`

Result: `{X=2.2400}`
Note that if exact mode is set, this example returns a null list as there are no exact solutions to the equation.

See also: SOLVE

SUBST

Type: Function

Description: Substitutes a value for a variable in an expression. The value can be numeric or an expression.

Access: Algebra, ↱ ALG

Input: Level 2/Argument 1: An expression.
Level 1/Argument 2: The value or expression to be substituted.

Output: The expression with the substitution made.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Substitute $x = z+1$ for x in the following expression, and apply the EXPAND command to simplify the result:

$x^2 + 3x + 7$

Command:

```
SUBST(X^2+3*X+7,X=Z+1)
EXPAND(ANS(1))
```

Result:

```
Z^2+5*Z+11
```

SUBTMOD

Type: Function

Description: Performs a subtraction, modulo the current modulus.

Access: Arithmetic, ↰ ARITH MODULO

Input: Level 2/Argument 1: The object or number to be subtracted from.
Level 1/Argument 2: The object or number to subtract.

Output: The result of the subtraction, modulo the current modulus.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

SYLVESTER

Type: Command

Description: For a symmetric matrix A, returns D and P where D is a diagonal matrix and

$A = P^TDP$

Access: Catalog, CAT

Input: A symmetric matrix.

Output: Level 2/Item 1: the diagonal matrix, D.
Level 1 1/Item 2: The matrix P.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

TABVAL

Type: Command

Description: For an expression and a list of values, returns the results of substituting the values for the default variable in the expression.

Access: Catalog, CAT

Input: Level 2/Argument 1: An algebraic expression in terms of the current variable.
Level 1/Argument 2: A list of values for which the expression is to be evaluated.

Output: Level 2/Item 1: The algebraic expression.
Level 1/Item 2: A list containing two lists: a list of the values and a list of the corresponding results.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Substitute 1, 2, and 3 into $x^2 + 1$.

Command:

```
TABVAL(X^2+1,{1 2 3})
```

Result:

```
{{1 2 3},{2 5 10}}
```

TABVAR

Type: Command

Description: For a rational function of the current variable, computes the variation table, that is the turning points of the function and where the function is increasing or decreasing.

Access: Catalog, CAT

Input: A rational function of the current variable.

Output: Level 3/Item 1: The original rational function.
Level 2/Item 2: A list of two lists. The first list indicates the variation of the function (where it is increasing or decreasing) in terms of the independent variable. The second list indicates the variation in terms of the dependent variable.
Level 1/Item 3: A graphic object that shows how the variation table was computed.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

TAN2SC

Type: Command

Description: Replaces tan(x) sub-expressions with sin(x)/(1–cos(2x) or (1–cos(2x))/sin(2x).

Access: Trigonometry, ↱ TRIG

Input: An expression

Output: The transformed expression.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

TAN2SC2

Type: Command

Description: Replaces tan(x) terms in an expression with sin(2x)/1+cos(2x) terms.

Access: Trigonometry, ↱ TRIG

Input: An expression

Output: The transformed expression.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).
Flag –116 – Prefer sin/cos affects the result:

- If flag –116 is set (Prefer sin()) then Tan(x) terms are replaced with:
  1 – Cos(2x)/Sin(2x)
- If flag –116 is clear (prefer Cos()) then Tan(x) terms are replaced with:
  Sin(2x)/1 + Cos(2x)

TAYLOR0

Type: Function

Description: Performs a fourth-order Taylor expansion of an expression at $x = 0$.

Access: Calculus, ↰ CALC LIMITS & SERIES

Input: An expression

Output: The Taylor expansion of the expression.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

TCHEBYCHEFF

Type: Function

Description: Returns the nth Tchebycheff polynomial.

Access: Catalog, CAT

Input: A non-negative integer, n.

Output: The nth Tchebycheff polynomial.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

TCOLLECT

Type: Command

Description: Linearizes products in a trigonometric expression by collecting sine and cosine terms, and by combining sine and cosine terms of the same argument.

Access: Trigonometry, ↱ TRIG

Input: An expression with trigonometric terms.

Output: The simplified expression.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

TEXPAND

Type: Command

Description: Expands transcendental functions.

Access: Trigonometry, ↰ TRIG

Input: An expression.

Output: The transformation of the expression.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Simplify the following expression:
$\ln(\sin(x+y))$

Command:

```
TEXPAND(LN(SIN(X+Y)))
```

Result:

```
LN(COS(Y)*SIN(X)+SIN(Y)* COS(X))
```

TLIN

Type: Command

Description: Linearizes and simplifies trigonometric expressions. Note that this function does not collect sin and cos terms of the same angle.

Access: Trigonometry, ↱ TRIG

Input: An expression.

Output: The transformation of the expression.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Linearize and simplify the following:

$(\cos(x))^4$

Command:

```
TLIN(COS(X)^4)
```

Result:

```
(1/8)*COS(4X)+(1/2)*COS(2X)+(3/8)
```

TRAN

Type: Command

Description: Returns the transpose of a matrix.

Access: Matrices, ↰ MATRICES OPERATIONS

Input: A matrix.

Output: The transposed matrix.

Flags: Exact mode must be set (flag –105 clear). Numeric mode must not be set (flag –03 clear).

Example: Transpose the following matrix: $\begin{bmatrix} 1 & 7 \\ 2 & -3 \end{bmatrix}$

Command: `TRAN([[1,7][2,-3]])`

Result: `[[1,2][7,-3]]`

TRIG

Type: Command

Description: Converts complex logarithmic and exponential sub expressions into their equivalent trigonometric expressions.

Access: Trigonometry, ↱ TRIG

Input: A complex expression with logarithmic and/or exponential terms.

Output: The transformed expression.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).
Must be in Complex mode (flag –103 set).

Example: Express the following in trigonometric terms:
$\ln(x+i)$

Command: `TRIG(LN(X+i))`

Result:

$$\frac{\mathrm{LN}(X^2+1)+2\cdot i\cdot \mathrm{ATAN}\left(\frac{1}{x}\right)}{2}$$

TRIGCOS

Type: Command

Description: Simplifies a trigonometric expression by applying the identity:

$(\sin x)^2 + (\cos x)^2 = 1$

Returns only cosine terms if possible.

Access: Trigonometry, ↱ TRIG

Input: An expression with trigonometric terms.

Output: The transformed expression.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

See also: TRIGSIN, TRIGTAN

TRIGSIN

Type: Command

Description: Simplifies a trigonometric expression by applying the identity:

$(\sin x)^2 + (\cos x)^2 = 1$

Returns only sine terms if possible.

Access: Trigonometry, ↱ TRIG

Input: An expression.

Output: The transformed expression.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

See also: TRIGCOS, TRIGTAN

TRIGTAN

Type: Command

Description: Replaces sin and cos terms in a trigonometric expression with tan terms.

Access: Trigonometry, ↱ TRIG

Input: An expression.

Output: The transformed expression.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

See also: TRIGCOS, TRIGSIN

TRUNC

Type: Command

Description: Truncates a series expansion.

Access: Catalog, CAT

Input: Level 2/Argument 1: The expression that you want to truncate.
Level 1/Argument 2: The expression to truncate with respect to.

Output: The expression from Level 2/Argument 1, with terms of order greater than or equal to the order of the expression in Level 1/Argument 2 removed.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

TSIMP

Type: Command

Description: Performs simplifications on expressions involving exponentials and logarithms.

Access: Exponential and logarithms, ↰ EXP&LN

Input: An expression

Output: The simplified expression.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

See also: TEXPAND
TLIN

VANDERMONDE

Type: Command

Description: Builds a Vandermonde matrix from a list of objects. That is, for a list of n objects, the command creates an $n \times n$ matrix. The i^{th} row in the matrix consists of the list items raised to the power of $(i-1)$.

Access: Matrices, ⇦ MATRICES CREATE

Input: A list of objects.

Output: The corresponding Vandermonde matrix.

Flags: Exact mode must be set (flag –105 clear).
Numeric mode must not be set (flag –03 clear).

Example: Build a Vandermonde matrix from the following list of objects:
$\{x, y, z\}$

Command: `VANDERMONDE({x,y,z})`

Result:

$$\begin{bmatrix} 1 & 1 & 1 \\ x & y & z \\ x^2 & y^2 & z^2 \end{bmatrix}$$

VER

Type: Command

Description: Returns the Computer Algebra System version number, and date of release.

Access: Catalog, CAT

Input: No input required.

Output: The version and release date of the Computer Algebra System software.

XNUM

Type: Command

Description: Converts an object or a list of objects to approximate numeric format.

Access: Catalog, CAT

Input: An object or list of objects.

Output: The objects in numeric format.

Example: Find the approximate value of π/2, 3*e*, and 4cos(2).

Command:
```
XNUM({π/2,3*e,4*COS(2})
```

Results:
```
{1.5707963268 8.15484548538 -
1.66458734619}
```

XQ

Type: Command

Description: Converts a number, or a list of numbers in decimal format, to rational format.

Access: Catalog, CAT

Input: A number, or a list of numbers.

Output: The number or list of numbers in rational format.

Example: Express .3658 in rational format:

Command:
```
XQ(.3658)
```

Results:
```
1829/5000
```

ZEROS

Type: Command

Description: Returns the zeros of a function of one variable, without multiplicity.

Access: Symbolic solve, ⬑ S.SLV

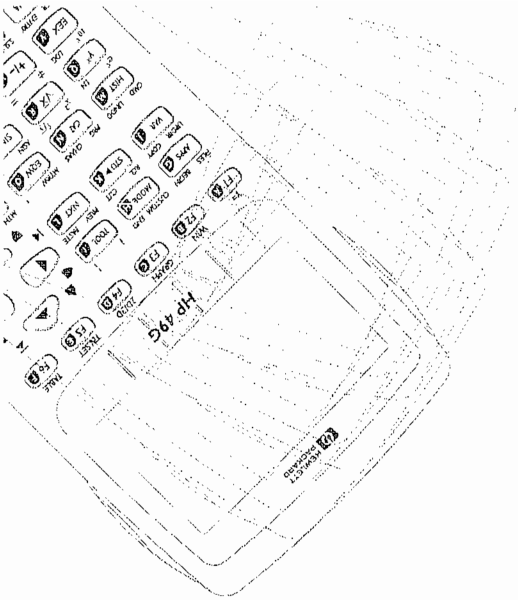
Input: Level 2/Argument 1: An expression.
Level 1/Argument 2: The variable to solve for.

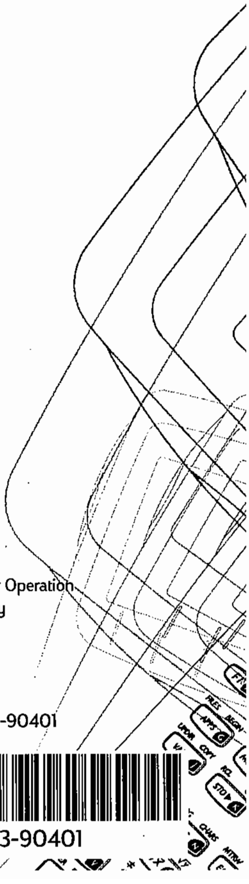
Output: The solution or solutions for the expression equated to 0.

Flags: For a symbolic result, clear the CAS modes Numeric option (flag –03 clear).
The following flag –settings affect the result:

- If Exact mode is set (flag –105 is clear), the function attempts to find exact solutions only. This may return a null list, even if approximate solutions exist.
- If Approximate mode is set (flag –105 set), the function finds numeric roots.
- If Complex mode is set (flag –103 set), the function searches for real and complex roots.

Index





Australian Calculator Operation
351 Burwood Highway
Forest Hill, 3131
Victoria, Australia

Printed in Singapore

Part number F1633-90401

F1633-90401

HEWLETT®
PACKARD