Theorist®

Version 2.0

reference manual

Published by Prescience Corporation, © 1990.
Acquired by Waterloo Maple Software, © 1993
Acquired by WebPrimitives, LLC © 1999
Acquired by MathMonkeys, LLC © 2003

Prescience Corporation

Corporate Headquarters:
Waterloo Maple Software
450 Phillip Street
Waterloo, Ontario
Canada N2L 5J2

E-Mail: info@maplesoft.on.ca

European Office:
Waterloo Maple Software
Tiergartenstrasse 17
69121 Heidelberg
Germany

E-Mail: 100275.2111@compuserve.com

Asian Office:
100 Kim Seng Road
Kim Seng Plaza #02-39
Singapore 0923

E-Mail: wmssfc@solomon.technet.sg

For more information, contact Waterloo Maple Software at one of the above addresses.

Theorist Reference Manual

First Edition, September 1994

Published by Prescience Corporation, © 1990.
Acquired by Waterloo Maple Software, © 1993
Acquired by WebPrimitives, LLC © 1999
Acquired by MathMonkeys, LLC © 2003

Contents

Preface

Who This Manual is For

The *Theorist Reference Manual* provides a complete description of Theorist, the software package for algebraic manipulation and visualization.

Theorist is made for both Macintosh and MS Windows and this manual assumes that you are familiar with the system you are using it on.

This manual also assumes that you have a basic understanding of mathematics and want to use Theorist to pursue scientific or engineering research, or you are studying mathematics, and want to improve your grasp of mathematical relationships in your field. This reference manual is not designed to teach mathematics but to describe how Theorist works so that you can explore, and gain new insight and understanding in any mathematical subject.

The basic structure of this manual follows the logical structure of Theorist from the outside in, from the broadest conceptual categories to the finest. For an instructional introduction to Theorist, see the *Theorist Learning Guide*.

WHAT THIS MANUAL IS ABOUT

Introduction Discusses system requirements and describes Theorist's underlying concepts, program algorithms, importing and exporting data and graphics, and printing.

Notebooks Describes Theorist's files.

Propositions Describes the entries you can make in a notebook.

Expressions Describes the fundamental units used to create propositions.

Manipulations Explains how to algebraically manipulate expressions and equations.

Tables Explains how Theorist handles large amounts of data.

Graphs Describes the two- and three-dimensional graphs you can create.

Editing Describes how to edit mathematical expressions.

Tips and Techniques Provides a few tips for improving your speed and efficiency.

Menus Lists all menu commands and provides a brief description of each.

Appendix A Provides equivalent keystroke tables for menus and other commands.

Appendix B Describes how to move expressions between Theorist and Expressionist, mathematical typesetting software from Prescience.

Appendix C Lists the contents of the disks distributed with Theorist.

Glossary Provides brief descriptions of key words used in this manual.

Bibliography Lists some books useful for exploring computer mathematics in greater depth and detail.

Index Cross-references all important topics in this manual.

Introduction

Theorist is a software system designed for algebraic analysis, graphical visualization, and mathematical insight and understanding. This section introduces the program, and presents a discussion of:

- System requirements
- Algorithms and numeric precision
- Mathematical perfection
- Importing and exporting data and graphics
- Printing
- The Propositional Schema

System Requirements for Macintosh

The minimum configuration for using Theorist is:

- a Macintosh Plus or Classic (or any later model; an older Macintosh upgraded to the equivalent of a Plus is satisfactory)
- two megabytes of memory (or more)
- System 6.0.2 (or later system software)
- hard disk with at least 1.5Mb of free space

See "Installation Preparations for Macintosh Systems" in the Theorist Learning Guide for information on the different editions of the Theorist application and the Macintosh systems for which they are made.

Mathematical manipulations can be computer intensive. The more memory (RAM) your machine has, the faster Theorist operates (up to a point). Pay attention to memory usage—if Theorist runs out of memory, it may be forced to quit. A memory "thermometer" is displayed along the top Theorist's palette window so you can see how much memory is in use.

You may receive a message that says you are inexplicably running low of memory (or have run out of memory). These messages can indicate that some memory may no longer be available; it is best to free up all memory associated with the program and start again. Save your work, quit the program, and re-launch Theorist.

System Requirements for Windows

The minimum configuration for using Theorist is:

- An IBM PC AT compatible computer with an 80386 (or later central processing unit)
- four megabytes (4Mb) of memory (or more)
- Microsoft Windows version 3.1 (or later)
- DOS version 3.1 (or later)
- hard disk with at least 2Mb of free space

Best performance is achieved with an 80386 (with math co-processor) or 80486 or better system with four megabytes of memory and a VGA color monitor.

Mathematical manipulations can be computer intensive. The more memory (RAM) your machine has, the faster Theorist operates (up to a point). Pay attention to memory usage—if Theorist runs out of memory, it may be forced to exit.

You may receive a message that says you are inexplicably running low of memory (or have run out of memory). These messages can indicate that some memory may no longer be available; it is best to free up all memory associated with the program and start again. Save your work, exit the program, and re-launch Theorist.

Algorithms and Numeric Precision

Theorist is designed to provide a superior workspace for mathematical investigation while balancing conflicting demands for speed, accuracy, reliability, generality, and compact and efficient code. As the program evolves, different trade-offs will lead to the implementation of a modified set of algorithms.

All of the algorithms used in Theorist are either available to the general public in journal articles or are obvious to anyone skilled in software design. The same is true of all commercially available mathematical software, especially products that claim to use exotic, and by implication, superior, algorithms. Numerical analysis and symbolic algebra are far older than computer science, and computer graphics is a well established field of study. All three are developed technologies, and the best algorithms are well known.

This *Reference Manual,* and the *Theorist Learning Guide,* describe and explain the features of the interface—how you use the program—rather than the internal workings of the program. If you want to discuss the relative merits of mathematical algorithms, or have a problem with our implementation of an algorithm, write us a letter or give us a call. We live for mathematics and computers, and would be delighted to hear from you.

The program endeavors to return fifteen digits of precision for numeric calculations. In some instances, in order to avoid unreasonable processing time, some algorithms return less than fifteen accurate digits. However, the results should be sufficient for most purposes.

In particular, numeric integration usually returns values that are accurate to approximately nine digits. Results of the Gamma function and other special functions (e.g., Bessel functions) are accurate to approximately seven or eight digits. If you need to know the exact degree of numeric precision, check your results and solve equations in another way. Where possible, attack a problem from different directions, and compare the results.

Floating-point round-off errors can also reduce the accuracy of calculations. All floating-point calculations have inherent limitations. For example, the following table shows what happens if you add and subtract a large number from a known value.

Expression	Calculated Value	Digits of Precision
$\frac{1}{3}$	0.333333333333333	15
$\frac{1}{3} + 1 \times 10^{11} - 1 \times 10^{11}$	0.3333282470703125	4
$\frac{1}{3} + 1 \times 10^{12} - 1 \times 10^{12}$	0.3333740234375	3
$\frac{1}{3} + 1 \times 10^{13} - 1 \times 10^{13}$	0.333984375	2
$\frac{1}{3} + 1 \times 10^{14} - 1 \times 10^{14}$	0.328125	1
$\frac{1}{3} + 1 \times 10^{15} - 1 \times 10^{15}$	0.375	0

MATHEMATICAL PERFECTION

Algebra is often a tricky business. Seemingly straightforward derivations can contain serious, and consequential, errors. One reason is that special cases can pop up when you least suspect them. Theorist handles special cases better than most computer algebra systems, but software can not, by its nature, generate mathematical perfection. Even before computers were introduced, the validity of a mathematical derivation was often not an easy thing to prove.

There are a variety of situations where "trustworthy" manipulations give answers that are only true some of the time. Round-off error can build up and overwhelm any significant result. A Taylor series, which is just a close approximation, is "equated" to the original function. If you change the behavior of names in the middle of a derivation, you can invalidate many of the manipulations already performed. If you use Theorist with its Auto Casing option turned on, you soon realize that special cases can multiply like rabbits, especially for long derivations. Sometimes, in order to get anything done, you simply have to turn Auto Casing off.

For another example, $\sqrt{x^2}$ looks like it is equal to x. And this is the value that Simplify returns. But for real numbers, the solution is in fact $|x|$. For complex numbers the solution is even more subtle. Theorist returns the single value x, because the square root operator is designed to return the principle value square root. In most cases this is entirely satisfactory, but there are exceptions.

Even in very simple situations, ambiguity can result. Consider the following equation:

$$x - y = 5 - y$$

The conclusion that $x = 5$ is valid—unless $y = \infty$. In that case, all bets are off.

There are myriad paths to mathematical misunderstanding. And computer-based systems add to this number their own special forms of possible error. No program can keep track of the many ways that you can go wrong and warn you of the dangers.

The moral of this story is that you should always check your work. For example, if you perform a symbolic integration, differentiate the result to confirm that your answer is correct. It is very easy to imagine your answers are always correct. It only takes a few disasters to learn otherwise.

Importing and Exporting Data and Graphics

Theorist is designed to work cooperatively with other software application programs. There are several ways to get numeric, textual, and graphic data into and out of the program. The following table lists the sections in this manual where various importing and exporting activities are described.

To	See
Import data-point values	Tables and Editing: Selecting Matrices
Import text or graphics (as Comments)	Propositions: Comment Propositions
Import mathematical expressions	Editing: Fortranish and Appendix B: Using Expressionist with Theorist
Export graphs (PICT or EPS)	Graphs: Printing and Exporting Graphs

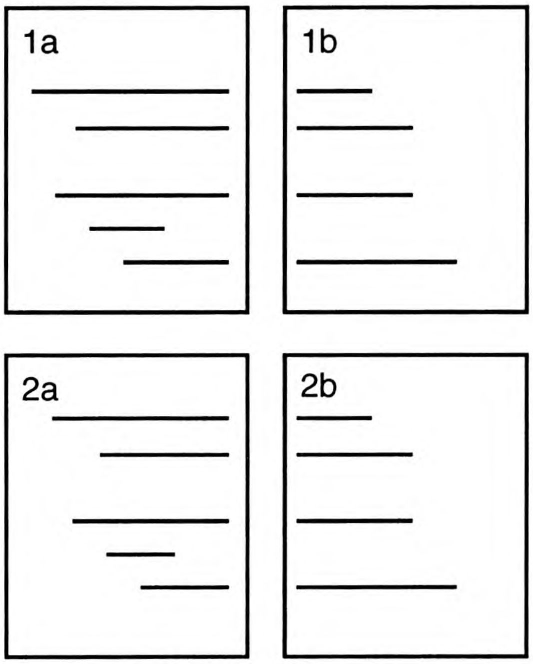
The graphs shown in the figures of this manual may look slightly different on your monitor. The appearance of a graph depends on the type of monitor that you are using. Theorist takes advantage of whatever display technology is available, black and white, gray scale, and all levels of color monitors.

PRINTING

Theorist provides an environment for mathematical investigations, rather than a means to create formal presentations. To present your work, you can easily export expressions and graphs to page layout, graphics, and animation software packages.

For less formal presentations and review, you can print your notebooks at any time: choose **Print...** from the **File** menu. A Theorist notebook is structured as an infinite two-dimensional surface. As such, notebooks do not always lend themselves to printing on fixed-size pieces of paper.

However, you can adjust several factors (e.g., Notebook Font size and Collapsars) to produce a reasonable printout. After making these adjustments, if a notebook is still wider than one page, a dialog box comes up warning you that horizontal pages will be used. Pages are labeled to help you piece them together. The first row of pages is labeled 1a, 1b, 1c, and so on. The second row is labeled 2a, 2b, 2c, and so on. For example, the following figure displays four pages of a two-page wide notebook.



To save paper, if a notebook is four pages long but one wide expression extends to two pages horizontally, only one additional page is printed. That is, your printed notebook could have pages numbered 1a, 2a, 3a, 3b, 4a.

To help you determine in advance how many pages a notebook will require, small tick marks are displayed at the top and bottom of each notebook window at the boundaries between horizontal pages.

If you are using a Macintosh, some complex graphs may be larger than can be stored in the memory reserved for the PrintMonitor utility. If this happens, a message is displayed declaring that the PrintMonitor does not have sufficient memory. To solve the problem, cancel any current printing jobs, select the PrintMonitor (in your System folder), choose **Get Info** from the **File** menu, and increase its memory allocation.

Propositional Schema

The fundamental idea in Theorist is the proposition. A proposition is a mathematical statement of equality (or other relationship), that describes a particular logical structure. Building from known or true propositions, you can derive conclusions that are also true. A theory consists of a group of propositions that are assumed to be true, and various conclusions that *are* true—if the original propositions are true. A theory can have many interrelated propositions, or just a few. For example, a simple theory could contain two equations:

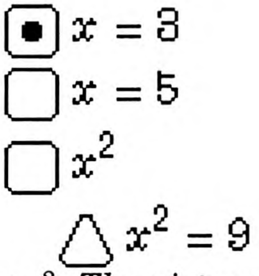
$$x = 3$$

$$y = 2x + 5$$

By manipulating these two propositions, you can see that, in this theory, a new proposition stating that y equals eleven would be another true proposition. In another theory, where x is equal to five, such a proposition would not be a valid conclusion.

You use Theorist to enter propositions that you believe to be true. Then, by manipulating these propositions, you can derive other valid propositions. Theorist files, called “notebooks,” consist of one or more hierarchically organized theories that contain propositions you enter, and valid conclusions derived from them.

Each proposition you enter is assumed to be true—in that theory. If you enter two contradictory propositions (e.g., x = 3 and x = 5), and try to derive a conclusion based on the value of x, Theorist marks which of the two proposition is used to derive the new conclusion.



● $x = 3$

□ $x = 5$

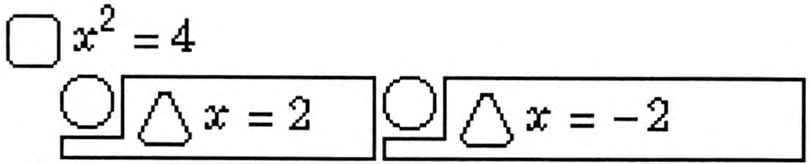
□ x^2

△ $x^2 = 9$

Calculating x^2, Theorist assumes x =3

In effect, the program indicates that a derived proposition is true, only if *these* par ticular propositions are also true.

If you want to discover what valid conclusions can be drawn from different and contradictory propositions, you can easily branch off one or more “case theories” in which you explore situations where certain particular facts are true, in addition to all propositions in any enclosing theory.



□ $x^2 = 4$

○ △ $x = 2$ ○ △ $x = -2$

Theorist makes case theories for the different possible solutions for x

The propositional schema derives from the mathematical understanding that simple known facts of relationship can be combined to create dependable but as yet unknown truths. Though powerful, this scheme is susceptible to what is known to computer programmers as the problem of garbage in, garbage out (GIGO).

If you enter propositions that are contradictory, and manipulate those propositions explicitly, you can prove that which is obviously false. For example, you could enter the following propositions:

$$x = 1$$

$$x = 0$$

Now, if you combine these two propositions, by substituting the value of x established in the first equation into the second equation, your new "valid" conclusion is:

$$1 = 0$$

Clearly, this is not the way to truth and understanding. As you can see, if you put garbage into a theory, you are likely to get garbage out.

Notebooks

A Theorist notebook is a hierarchically organized work space for algebraic, numeric, and graphical study. Whenever you work with Theorist, you work in a notebook. Notebooks contain expressions, equations, theories, comments, declarations, rules, and two- and three-dimensional graphs. Each item in a notebook is either a proposition or part of a proposition.

Each notebook is a self-contained theoretical world displayed in a standard window that you can move, re-size, and print.

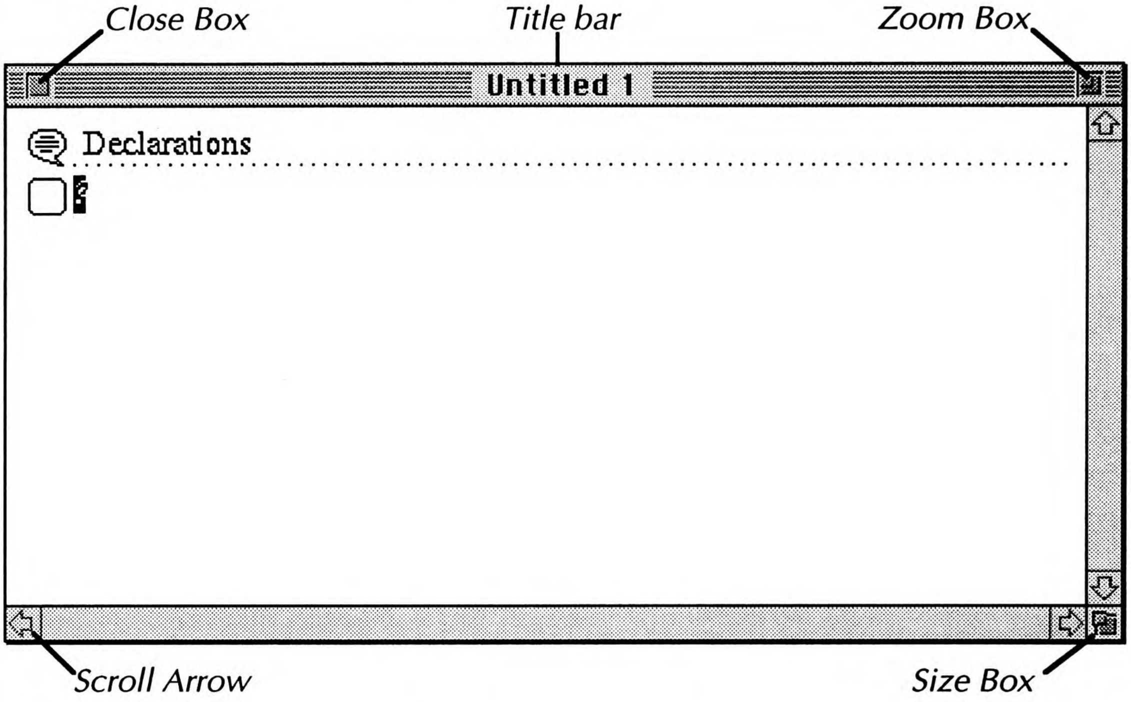
This section describes:

- Notebook files
- Notebook management
- The notebooks included on your Theorist distribution disks
- Customizing Notebooks
- Compatibility with Theorist version 1.5
- Compatibility between Theorist on different computers

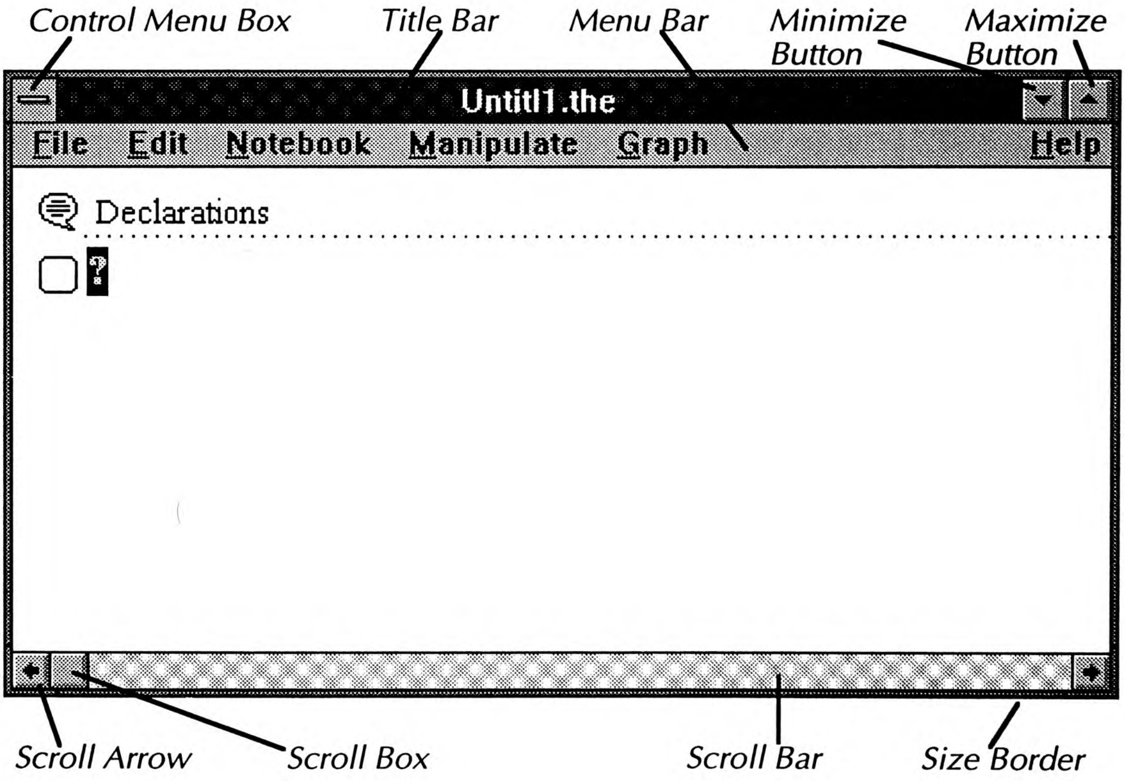
Notebook Files

When you start the Theorist application program, it opens a copy of the New Notebook file and names the notebook window "Untitled 1" (Macintosh) or "Untitl1.the" (Windows).

A new Macintosh notebook window



A new Windows notebook window



A notebook file is displayed in a standard window with the standard controls for scrolling, sizing, zooming, and closing the window. (These controls are slightly different between Macintosh and Windows versions of Theorist.) Several notebooks can be open at once; however, only the front-most window is active. To move a notebook window, click on the title bar, drag the window to a new location, and release.

The default new notebook contains a comment proposition titled "Declarations." The dots under it indicate that daughter propositions are currently hidden. To show (or hide) these propositions, double-click on the proposition icon. The default new notebook contains numerous name declarations and transformation rules, including whole sections devoted to trig, hyperbolic and logarithmic names, and a Standard Rules section that contains trig identities and other rules. (For more information about propositions, name declarations, and transformation rules, see the Propositions section of this manual.)

Use the commands on the File menu to create new notebooks and open, close, save, and print notebook files as you would, for example, word processing files.

Startup Notebooks

For Windows, use "Notebk0.the", "Notebk1.the", "Notebk2.the", and so on, up to "Notebk10.the"

You can customize your Theorist work environment in several ways. If you create one or more notebooks with the titles "Notebook 0", "Notebook 1", "Notebook 2", and so on, up to "Notebook 10", and keep these files in the same directory or folder as the Theorist application, every time you launch the program directly (by double clicking on the application icon), these notebooks also open. When you quit Theorist, any such notebooks are automatically saved. (All other notebooks present a dialog box, asking if you want to save your changes.)

Note for Macintosh users

If you launch Theorist by opening a notebook file (rather than opening the application), startup notebook files do not open (even numbered startup notebooks). Startup notebooks only open if you launch the program directly from the application icon.

Beware of automatically saved notebooks.

If you do not want to save your changes to one of these notebooks, choose **Revert to Saved** from the **File** menu before closing the notebook. Otherwise, your changes are saved automatically.

By automatically opening (and saving when quitting), startup notebook files let you easily maintain several streams of work.

Default Notebook

You can customize Theorist so new notebooks open with fonts, preferences, declarations, and other settings you specify. If you create and save a notebook with the Macintosh title "New Notebook" or Windows title "NewNote.The" and keep it in the same folder or directory as the Theorist application, each new notebook uses the settings saved in that special default notebook. (The New Notebook file supplied with the Theorist disk set may be necessary for tutorials in the Learning Guide.)

If you have a less powerful computer, consider deleting parts of the default New Notebook. The largest part, by far, is the Standard Rules section. If you remove it, most trigonometric, hyperbolic, logarithmic, and exponential rules will be lost. (However, several rules which are "hard-coded" into Theorist will remain intact, such as integration, differentiation, and anti-function rules.) In extreme cases, you may want to remove even more. Theorist can function without any declarations whatsoever, as long as you do only arithmetic.

If you have a more powerful computer, consider augmenting the default New Notebook rules and declarations. You can get identities from other notebooks in the Mathematics directory (supplied with Theorist; see Appendix C). But sometimes the best identities are the ones you derive yourself.

Stationery Files (for Macintosh only)

Using stationery files is another powerful way to customize your work environment. If you find that you regularly want to start a new notebook with a particular set of rules and declarations (other than the minimal default set), create a "stationery file" containing these rules and declarations.

To create a stationery file, modify a notebook to include the particular set of declarations and preferences you want, then save it and close it. Go to the Finder, select the file, and choose **Get Info** from the **File** menu. In the info window, click the checkbox for Stationery. This makes it a special file that acts like a pad of paper. When you open a stationery file, you effectively "tear off" a new untitled notebook with all the declarations and preferences you created.

If you name a stationery file "Notebook 1" (or "Notebook 2", etc.), and keep this file in the same folder as the Theorist program, a new untitled notebook file containing that set of rules and declarations is created every time you launch the program, instead of the default New Notebook. However, when you create new notebooks, they do use the New Notebook file.

Revert Notebook

If you have made changes to a previously saved notebook but you do not want to save your changes, choose **Revert to Saved** from the **File** menu before closing the notebook. The notebook will remain untouched.

This is *not* a preference setting saved with notebooks; you must turn it on every time you launch Theorist.

AutoSave

When this item is checked on the **File** menu, Theorist periodically saves open notebooks.

Deleting Work

If you do not want to save the results of a particular investigation, select one or more propositions and press Delete. Deleting a proposition with daughter propositions deletes those dependent propositions as well. You can safely press Delete repeatedly. Any collapsed proposition (hiding its daughter propositions, and shown with a dotted line underneath) must be explicitly selected to be deleted. The comment proposition at the top of a notebook is usually collapsed and therefore safe from unintended deletion.

Distribution Notebooks

Several notebooks are included on the Theorist disks distributed with the Theorist package. These notebooks contain examples of colorful surface plots and other graphics as well as a wide range of mathematical rules and functions.

These notebooks are organized by hierarchical "bundles." A comment proposition acts as a title and "handle" for a group of daughter propositions that contain useful rules and declarations. To move a bundle from these distribution notebooks, select the comment proposition of the bundle you want, copy it, and paste it into your own notebook.

If there are any name conflicts (e.g., a bundle contains a previously declared name), a message announces that there is a conflict. The first instance of the conflicting name is selected. If the name declarations are identical (e.g., the bundle and your notebook both declare sin as the standard sine function), delete the selected name.

Only the first naming conflict is announced. More conflicts may exist. Usually this is not a problem. The next time a manipulation requires the conflicted name, an appropriate warning is displayed asking that one of the declarations be deleted. You can also flush out any possible naming conflicts by clarifying the notebook repeatedly until all conflicts are reported and cleared.

If a name is used in two different ways (e.g., you have created a function named G and the bundle declares G to be a different function), delete the newly pasted bundle, change the spelling of the conflicting name declaration(s) in your notebook, and re-paste the bundle. To change the spelling of a name, open the Declarations comment at the top of the current notebook and find the name declaration proposition for that name. Then change the spelling in that proposition. Changing the spelling or behavior of a name in its declaration is immediately reflected throughout the notebook.

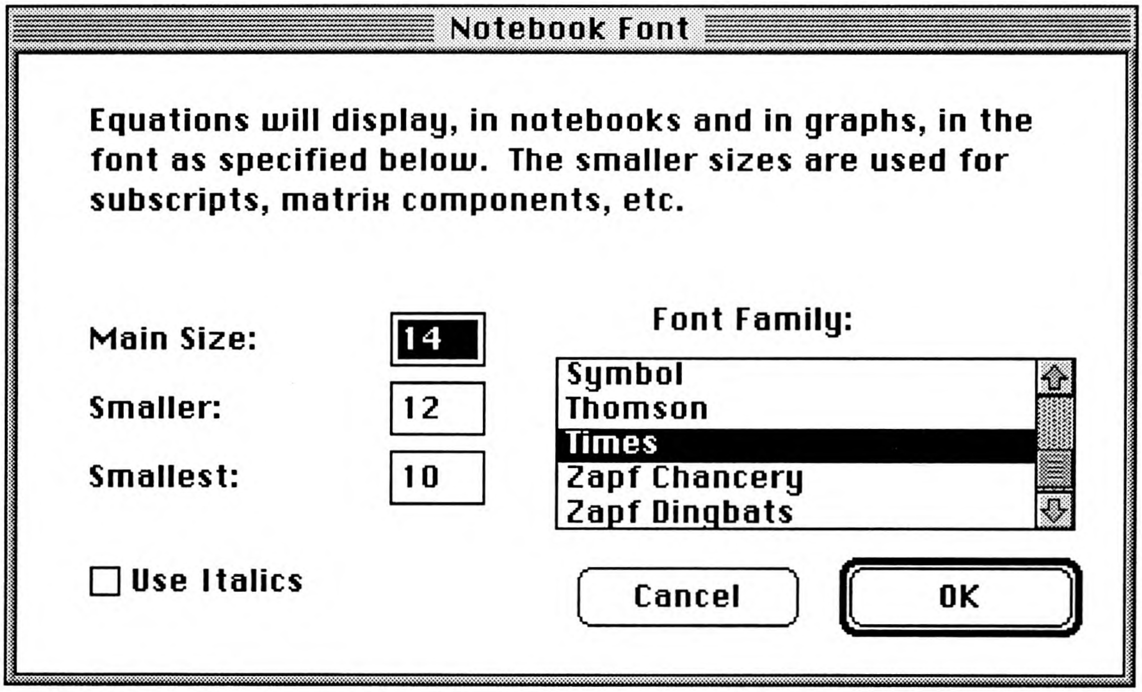
For more information about name declarations and Theorist's outlining tools, see the Propositions section. See Appendix C for a complete list of notebooks included on the Theorist distribution disks.

CUSTOMIZING NOTEBOOKS

Notebook Font

The font and font size for all of the equations in a Theorist notebook may be specified in the Notebook Font dialog. Choose **Notebook Font...** from the **Notebook** menu.

This does *not* affect fonts in comments or the numerical labels in graphs.



Notebook Font

Equations will display, in notebooks and in graphs, in the font as specified below. The smaller sizes are used for subscripts, matrix components, etc.

Main Size: 14

Smaller: 12

Smallest: 10

Font Family: Times

□ Use Italics

Sizes

The Main Size specifies the size of variables, functions, and the like as they are written in equations. The Smaller box represents the size of the first sub- or superscript in an expression and is also used in wording of wordy propositions. Smallest specifies the smallest size for any text in an expression. As an example consider e^{-x^2}. In this expression, e is in the Main size, $-x$ is in the Smaller size, and the 2 is set in the Smallest size. (Any superscripts beyond the 2 would also be in the smallest size.)

To change all sizes simultaneously, simply change the Main size and appropriate smaller and smallest sizes are automatically chosen for you. If you wish, you can then modify the smaller size to something different, and the smallest size changes accordingly. Changing the smallest size has no effect on the other sizes.

Italics

The box at the bottom of the dialog, when checked, tells Theorist to use italics where appropriate. By convention, single character names (usually variable names like x) are italicized, while multi-character names (frequently function names like sin) are not italic in mathematical expressions, and Theorist will enforce this rule if the box is checked.

Theorist uses italics in several other places if the checkbox is on. The non-modifiable parts of tables and the frame labels on the top of animation frames are usually drawn in italic; turn off the check box to have them drawn plain as well.

On the screen and some printers, italics do not always look good; they sometimes look chunky and slanted. On a Macintosh, this is most often because you have a bitmap font that is plain style, and the computer merely slants it over as a cheap imitation of italics. You can avoid this problem by turning off the checkbox so everything is plain and no italics are used. If you want good looking italics on the screen, you can install italic TrueType or Adobe Type Manager (ATM) fonts, or install specially sculpted italic bitmap fonts. You may have to remove some bitmap versions of your fonts because they override TrueType or ATM fonts for screen display (even if one is italic and the other is not).

Editing Difficulties

If the font you are using does not allow you to select names easily by clicking and dragging, consider:

❶ Increasing the font size.
❷ Turning off the italics.
❸ Changing to another font.

Frequently it is easier to work on a notebook in a larger size, then print it using a smaller font size.

'Old' Theorist Look

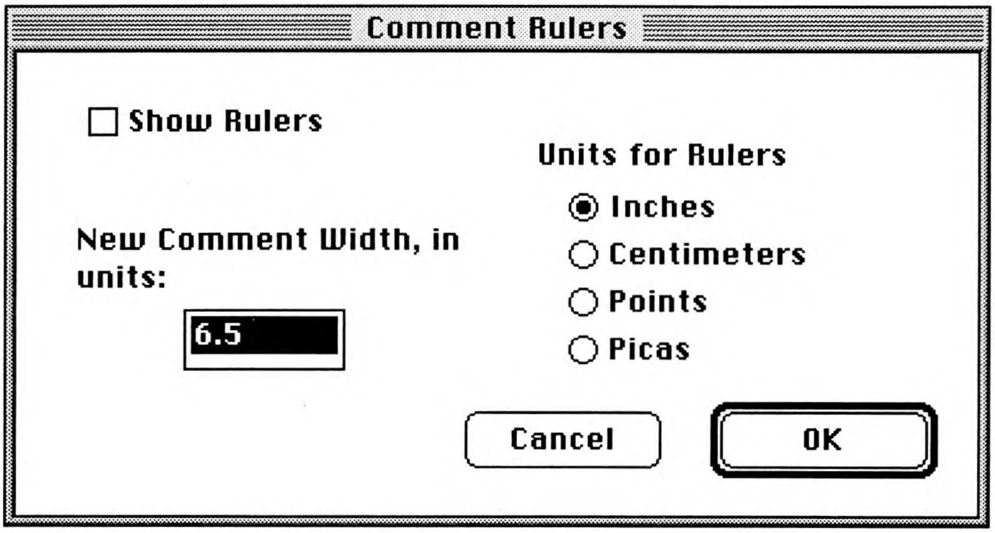
If you are nostalgic about the look of equations in older Macintosh versions of Theorist, you can recreate the look. Use the old Bookman (including italic) bitmapped screen fonts from Adobe. (The newer TrueType or ATM versions do not look the same.)

Comment Default Font

The default comment font characteristics follow the settings of the default Notebook Font as described above. However, comments do not use italics by default, regardless of the italics setting in the dialog.

Comment Rulers

Choose **Comment Rulers...** from the **Notebook** menu to bring up the comment rulers dialog, which lets you show or hide the comment rulers and set their units of measurement.



Comment Rulers

□ Show Rulers

New Comment Width, in units: 6.5

Units for Rulers

● Inches
○ Centimeters
○ Points
○ Picas

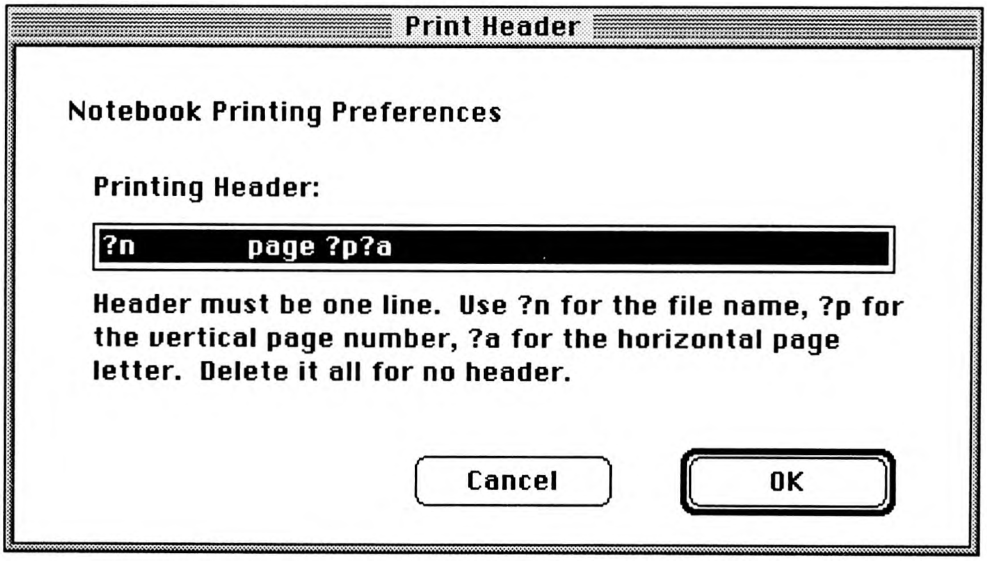
You can also show or hide the rulers with the 📏 button on the 💬 pop-up palette.

Page Break

Choose **Notebook▸Insert▸Page Break** to insert the page break proposition in the notebook. This forces the following propositions to start at the top of a new page when the notebook is printed.

Notebook Header

To change the print header for notebooks use the notebook print header dialog. Choose **Notebook▸Preferences▸Print Header...**.



Print Header

Notebook Printing Preferences

Printing Header:

?n   page ?p?a

Header must be one line. Use ?n for the file name, ?p for the vertical page number, ?a for the horizontal page letter. Delete it all for no header.

Edit the text in the box (e.g., add your name and organization, the date, etc.) to change the print header.

You can insert special codes into your header text so that special information, such as the notebook name or page number, is printed in your headers. For example, the text and codes displayed in the above screen shot of the dialog make the first printed header read:

Notebook 7 page 1a

As another example, the text and codes below,

George's (??) File: ?n, Page#?p-?a

make the printed header on page four of a notebook read:

George's (?) File: My Thesis, Page#4-a

The header is printed in the Equation Font described previously.

The following lists Theorist header codes and their meanings.

Code	Meaning
?n	notebook file name (without directory path)
?p	page number (vertical)
?a	page letter for extra-wide pages (horizontal a, b, c, etc.)
??	question mark

Compatibility with Theorist Version 1.5

Theorist 1.5 was available only for the Macintosh.

Theorist can open notebook files created with Theorist version 1.5 (i.e., any of versions 1.51, 1.52, 1.53).

When you copy expressions from 1.5 and paste to the current Theorist, there are some precedence changes. Cross product and dot product previously were bound tighter than division and function call; now they bind looser than multiplication. Therefore:

Expression	1.5 Meaning	Pasted Becomes
$z w \bullet y$	z*(w • y)	(z*w) • y
$z w \times y$	z*(w × y)	(z*w) × y

This problem is compensated for when you open an old notebook with the current Theorist, so your old files will not have this problem. The problem exists only if you copy from 1.5 and paste to the current version.

Notebooks and copied equations can not always be moved from the current version to version 1.5. Only certain expressions can be copied, mostly those with just + – * / ^ ! and functions.

Compatibility Between Platforms

Theorist operates as platform-independently as possible so Theorist for Macintosh and Theorist for Windows can share notebook files. There is no translation or exporting or importing necessary; just save a notebook under one platform and open it under another. (Of course, you must have the necessary hardware and/or software to transfer the files between the platforms.)

Theorist notebooks are saved in a format which can be copied as 7-bit text to avoid problems when e-mailing or otherwise transferring between platforms.

Propositions

Notebooks contain a set of propositions organized hierarchically. Propositions can contain other propositions and are often related to other propositions in various ways. Each notebook always contains at least one proposition, a Main Theory proposition. The main theory is essentially equivalent to a notebook. All propositions that you enter are contained within the main theory.

All types of propositions are presented in this section:

- Statements
- Working Statements
- Case Theories
- Graphs and Plots
- Comments
- Name Declarations
- Independence Declarations
- Transformation Rules
- Outlining
- Page Breaks

The section ends with a discussion of the tools you can use to organize and arrange your propositions.

To find out more about any proposition, select it and choose **Get Info** from the **Notebook** menu. If the proposition is derived from other propositions, you can move to those propositions through a hypertext-like network. The hierarchical display of propositions in a notebook presents the overt story of mathematical investigation and discovery. Using Get Info, you can trace the hidden connections and logical dependencies between propositions.

STATEMENTS

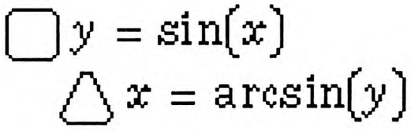
A statement is an expression in a Theorist notebook. Every statement is either an “assumption” or a “conclusion.” Every statement you enter is an assumption. Conclusions are logically derived from one or more statements. To create a new assumption, press the Return key.

Statements can be expressions (e.g., $3x$, $\log(y)$) or equations (e.g., $x = \sin(y)$, $y = x^3 + 2x + 4$). The following recursive definition defines statements exhaustively. A statement can be:

- A number (e.g., 4, 27, 2.3)
- A name (e.g., x, y, sin)
- A wildcard variable (e.g., $\mathscr{x}$, $\mathscr{y}$)
- An op that encloses one or more expressions (e.g., $4 + y$, $x = y$)

Numbers, names, and wildcard variables are the elementary particles of expressions and equations. Ops are used to put these elements together. For more information about ops, see the Expressions section.

Assumptions are displayed with a square icon; conclusions with a triangular icon:



□ $y = \sin(x)$

△ $x = \arcsin(y)$

Each conclusion is the result of a valid manipulation. Only you can create assumptions, and only Theorist can create conclusions. You can not edit a conclusion; if you try to, the statement is duplicated as a new assumption that you can edit.

When you manipulate a selected expression, the conclusion is displayed directly beneath that expression, indented to the right. Other statements may be used to derive the conclusion.

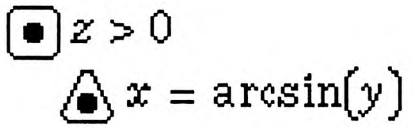
You can always discover which statements a conclusion is derived from. Select the conclusion and choose **Get Info** from the **Notebook** menu. The displayed dialog box states the type of the proposition. Click on the **More Info** button to see a list of all statements used to generate the conclusion.

If you rearrange the statements in a notebook, you do not change their logical relationship. However, if you delete a conclusion or an assumption, all conclusions directly derived from it become invalid and produce error messages. If you modify an assumption, its dependent conclusions are remanipulated by Theorist if the **Always ReManipulate** option is on (under the **Manipulate** menu). Modified assumptions may also produce invalid conclusions with error messages. The Divert Cascade command can take care of these situations. See the Complicated Integral tutorial in the Learning Guide for an example of how logical dependence affects propositions.

Working Statements

▣ ◭

Working statements are assumptions or conclusions used by a manipulation to generate a new conclusion or used to generate a graph. You can specify which statements to use as working statements, or they can be designated automatically when you perform a manipulation. Working assumptions and conclusions are displayed with square and triangular icons, respectively. They can be distinguished from ordinary statements by the dot contained within the icon:



$z > 0$

$x = \arcsin(y)$

All working statements provide specific information about the name on the left-hand side of the statement.

To make a statement a working statement, select it and choose **Make Working Stmt** from the **Notebook** menu.

In some instances, an assumption or conclusion is automatically promoted to a working statement. For example, if you are calculating the value of an expression that contains the name *y*, Theorist searches the current theory (and all enclosing theories) for a statement of the form *y* = *value*. The first such statement becomes a working statement and its icon is marked with a dot.

The search for a possible working statement proceeds through the current theory from the top to the bottom, then the enclosing theory (if any) from top to bottom, and so on.

You can change a working statement into an ordinary statement by selecting it and choosing **Stop Working Stmt** from the **Notebook** menu. You can also turn a working statement off by selecting an alternate statement of the same type that defines the same name, and making the new statement a working statement. There can be only one working statement of each type for a particular name in a particular theory.

There are six types of working statements:

Statement Type	Example	
Value	⊡ $x = 5$	
Maximum	⊡ $x < 5$	⊡ $x \leq 10$
Minimum	⊡ $x > 5$	⊡ $x \geq 10$
Increment	⊡ $dy = 0.0001$	
Function	⊡ $f(\boldsymbol{x}) = \sin(\boldsymbol{x})$	
Order	⊡ $y^5 = 0$	

Value Statement

⊡ $x = 5$

The Value Statement is the most common working statement. It defines a working value for a name. The name must be of class Constant, Variable, or M-Linear Operator. (It can not be a D-Linear Operator or a Function.)

Value statements are used by the Calculate manipulation and to create graphs. Symbolic manipulations (e.g., Simplify, Factor) do not evaluate expressions and therefore do not take into account value statements.

Maximum Statement

⊡ $x < 5$

⊡ $x \leq 10$

Minimum Statement

⊡ $x > 5$

⊡ $x \geq 10$

Maximum and Minimum statements define an exclusive or inclusive upper or lower bound for a name. The name must be of class Constant or Variable. Only one maximum statement and one minimum statement can be in force at a given time for a given name.

Maximum and minimum statements are often used to control the original display of graphs. If you plot a function, such as $y = \sin(x)$ in two dimensions, the graph theory usually uses default bounds of $x > -3$ and $x < 3$. Once created, graphs can be easily altered, but you can also set larger or smaller bounds explicitly before creating a graph.

If you only create a single maximum or minimum statement (e.g., $x > 0$), graphs are created using this as one of the boundary values; the other is derived from internal heuristics.

If Auto Casing is on, maximum and minimum statements can also affect the Move Over and Isolate manipulations. If alternate solution values for these manipulations are within the range determined by a maximum and minimum statement, more than one case theory is generated. For example, if $x > 0$ is a working statement, $x^2 = 9$ has only one solution: $x = 3$. Otherwise, two case theories are generated for $x = 3$ and $x = -3$.

The iterative ops, Summations (Σ) and Pi Products (Π), are also affected by minimum and maximum statements. For example, if $k \le 3$ is a working statement, and you expand the summation of x^k *as* k goes from 1 to 10, the solution is as follows.

$$\sum_{k=1}^{10} x^k = x^3 + x^2 + x$$

Increment Statement

▣ $dy = 0.1$

The Increment Statement defines an increment value for a variable. The name must be of class Variable. The increment value is used for numerical calculations of derivatives. In the absence of an increment statement, a default value of 0.0001 is used.

If the increment value for a variable is set to zero (e.g., $dy = 0$), the variable (y) acts as a constant and all derivatives taken with respect to it are zero in that theory.

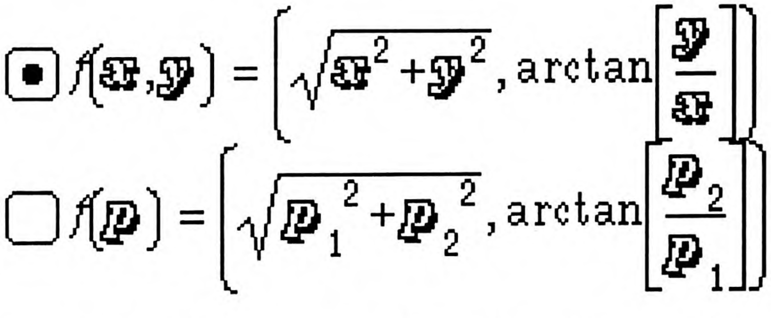
Function Statement

▣ $f(x) = x^2$

The Function Statement defines a functional relationship between one or more input values and one or more output values. The function name must be of class Function. Function arguments must be wildcard variables (e.g., x, y). (See the Expressions section for more information.)

Function statements are used to create graphs. They are also used by the Calculate manipulation to numerically evaluate the function. Symbolic manipulations do not take function statements into account.

If a function statement has more than one argument, all occurrences of that function must have the same number of arguments. You can also pass functions a vector as a single argument that contains more than one element. In the following examples, x and y are single arguments, whereas p is a two-element vector.



▣ $f(x,y) = \left[\sqrt{x^2+y^2}, \arctan\left[\frac{y}{x}\right]\right]$

□ $f(p) = \left[\sqrt{p_1^2+p_2^2}, \arctan\left[\frac{p_2}{p_1}\right]\right]$

Usually it is preferable to create function statements with vector Wildcard variables so that you can pass in vector expressions such as the product of two matrices. Use an equation of the second form to pass vector arguments.

Order Statement

▣ $y^5 = 0$

The Order Statement reduces to zero the powers of a name greater than a designated value. The name must be of class Constant, Variable, or M-Linear Operator. Use order statements only if the absolute value of the name (or, if the name is a matrix or M-Linear Operator, the largest eigenvalue) is much smaller than one.

The order statement $y^5 = 0$ declares that y^5 and all greater powers of y can be reduced to zero. This statement is taken into account by the Simplify and Expand manipulations.

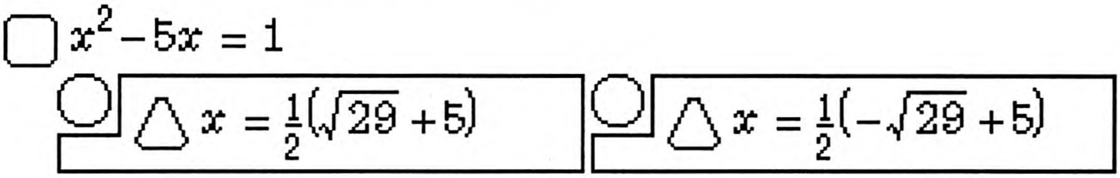
CASE THEORIES

A theory is a list of related propositions. Every notebook consists of a Main Theory that contains propositions, including all "case theory" propositions. Case theories can be nested in any hierarchical fashion and are displayed with a circle icon and a variable sized box containing all internal propositions.

Case theories are self-contained work areas that you can use to explore divergent trains of thought. For example, in a particular mathematical situation, you can explore a world where $x = 3$ and another where $x = 2$.

If Auto Casing is on (choose **Manipulate▸Preferences▸Auto Casing**), any manipulation that generates multiple solutions creates a case theory for each solution. For example, if you solve the following equation for x by making a multiple selection (each x highlighted) and isolating, both solutions are displayed.

Case Theories usually appear side-by-side but may appear one atop the other.



□ $x^2 - 5x = 1$

○ △ $x = \frac{1}{2}(\sqrt{29} + 5)$

○ △ $x = \frac{1}{2}(-\sqrt{29} + 5)$

Generating Case Theories

Only two manipulations, Isolate and Move Over, generate case theories, and only when Auto Casing is on. If Auto Casing is off, only the first solution is given within the current theory.

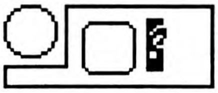
Manipulations that would generate a large number of case theories may display a single solution that contains arbitrary constants. For example, the solution in x of $x^5 = 1$ is either $x = 1$ (if Auto Casing is off) or (if Auto Casing is on):

$$x = e^{2/5\, \pi\, n_{106}\, i}$$

The second solution includes the arbitrary constant n_{106}, which indicates an arbitrary integer value. For more information on arbitrary constants, see the Expressions section.

Creating Case Theories

You can also create any number of case theories by choosing **Notebook▸Insert▸Case Theory**. All selected statements (and conclusions based on those statements) are included in the new case theory. If no statements are selected, a single empty assumption appears in the new case theory:



Because Theorist is non-procedural, you can enter conflicting assumptions into a single theory, such as $x = 1$ and $x = 0$. In practice, it is best to put conflicting assumptions in different case theories. If you perform a manipulation in a theory that contains conflicting assumptions, you can end up with confusing and contradictory results.

In a few instances, it is useful to have conflicting statements within a single theory. If you approximate the value of an expression (e.g., by creating a Taylor Series), this appears as a strict equality:

□ $y = \sin(x)$

△ $y = -\frac{1}{5040}x^7 + \frac{1}{120}x^5 - \frac{1}{6}x^3 + x$ *Seventh Order Taylor Series*

Using the Substitute and Move Over manipulations, you can determine the closeness of the approximation for various values of x ($x = 0.1$, 1, 10):

□ $y = \sin(x)$

△ $y = -\frac{1}{5040}x^7 + \frac{1}{120}x^5 - \frac{1}{6}x^3 + x$ Taylor

△ $\sin(x) = -\frac{1}{5040}x^7 + \frac{1}{120}x^5 - \frac{1}{6}x^3 + x$ Substitute

△ $0 = -\sin(x) - \frac{1}{5040}x^7 + \frac{1}{120}x^5 - \frac{1}{6}x^3 + x$ MoveOver

□ $x = 0.1$

△ $0 = -\sin(0.1) + 0.099833$ Substitute

△ $0 = -2.7478\times10^{-15}$ Calculate

□ $x = 1$

△ $0 = -\sin(1) + \frac{4241}{5040}$ Substitute

△ $0 = -2.7308\times10^{-6}$ Calculate

□ $x = 10$

△ $0 = -\sin(10) - \frac{1}{5040}\cdot10^7 + \frac{2030}{3}$ Substitute

△ $0 = -1306.9$ Calculate

Usually you specifically indicate the equations and definitions used for a manipulation. However, if a particular proposition is needed for a manipulation, and can not be found in the current case theory, the manipulation may search enclosing case theories for the needed definition. Case theories are searched from the innermost to the outermost, and from top to bottom within each theory. If an external statement is used as a working statement, its icon is marked with a dot.

The following propositions, though they may be outside the theory, are available to any enclosed case theory:

- Working statements
- Name declarations
- Independence declarations
- Transformation rules

Case theories always describe more specific conditions than their enclosing theory. Most case theories start with one or more equations distinguishing the theory from its rival theories.

Graph theories are similar to case theories. They contain propositions describing the various graphical elements that make up a graph. Most of these propositions are only valid inside graph theories. You can also create case theories within graph theories.

Graph Theories

Two- and three-dimensional graph theories are special theories, and all plots are propositions contained within graph theories.

Graph theories, like case theories, are displayed with a variable box containing all internal propositions. The icons for graphs and plots are:

Line plot

Contour plot

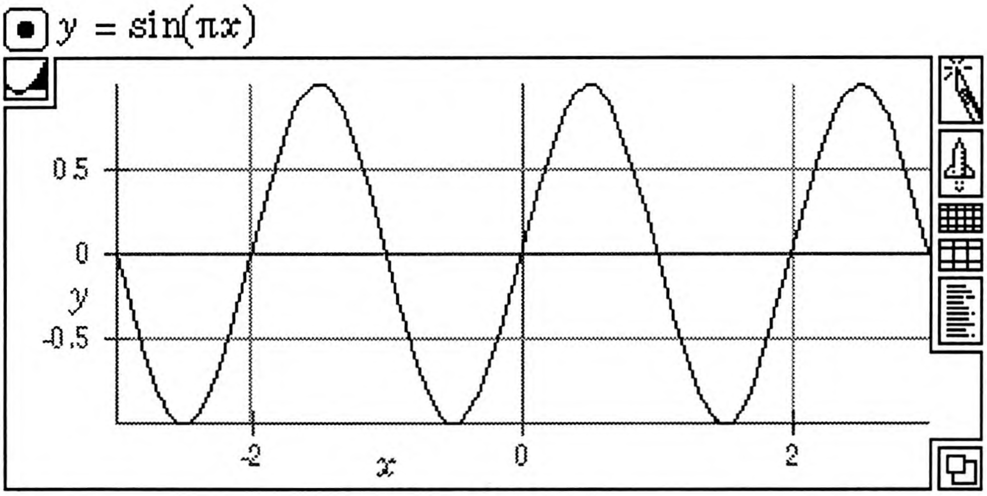
Surface plot

Scatter plot

Two-Dimensional graph theory

Three-Dimensional graph theory

The following is a simple two-dimensional graph theory displaying a single plot:



For more information on graph theories and plot propositions, see the Graphs section of this manual.

Comments

Comment propositions use a balloon icon and contain text and/or graphics used to annotate your notebooks. They are ignored during all manipulations and operations.

The following is a comment proposition:

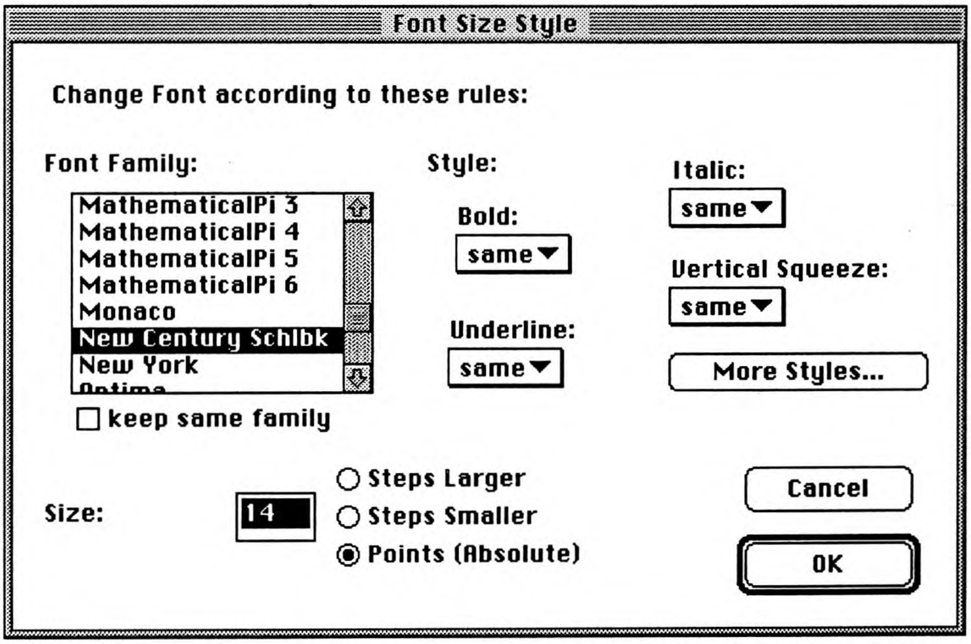
And then, I said to myself, that must mean that when an object is at rest, it has a rest energy equivalent to $E = mc^2$

The first proposition in a new notebook is the comment proposition called Declarations. It contains a set of name declarations as daughter propositions. Each new name that you declare is placed within this proposition. If you delete this comment proposition, new names are placed at the top of the notebook.

The keystroke(s) used to create comments can be changed with the Enter Key Prefs dialog.

To create a comment proposition, press enter on the numeric keypad. If you create a new assumption by accident, immediately press Enter and it turns into a text comment.

To alter the font, size, or style of the text choose **Comment Font...** from the **Edit** menu and use the following dialog:



Font Size Style

Change Font according to these rules:

Font Family:
- MathematicalPi 3
- MathematicalPi 4
- MathematicalPi 5
- MathematicalPi 6
- Monaco
- New Century Schlbk
- New York
- Optima

□ keep same family

Style:

Bold: same▾

Underline: same▾

Italic: same▾

Vertical Squeeze: same▾

Size: 14

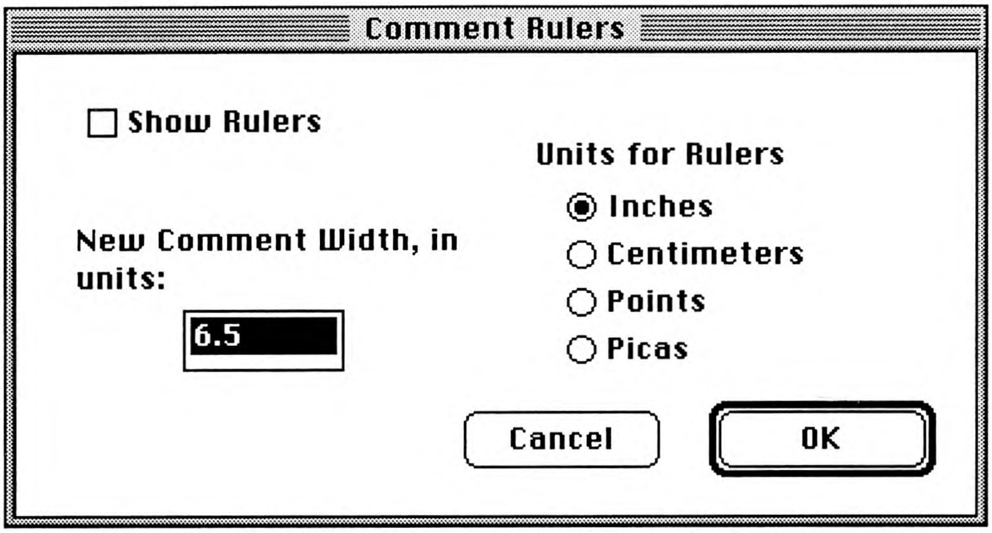
○ Steps Larger
○ Steps Smaller
● Points (Absolute)

To change the color of the text choose an item from the **Comment Color** submenu of the **Edit** menu.

To enter a hard carriage return within a text comment (i.e., to force a new line), press shift return .

You can also specify the font and size of all future text comment propositions. The default comment font characteristics follow the settings of the default Notebook Font, set from the Notebook menu.

To adjust the size and shape of a text comment, alter its ruler settings. (To show comment rulers, choose **Comment Rulers...** from the **Notebook** menu to bring up the comment rulers dialog, and click the **Show Rulers** checkbox.)



Comment Rulers

□ Show Rulers

New Comment Width, in units: 6.5

Units for Rulers

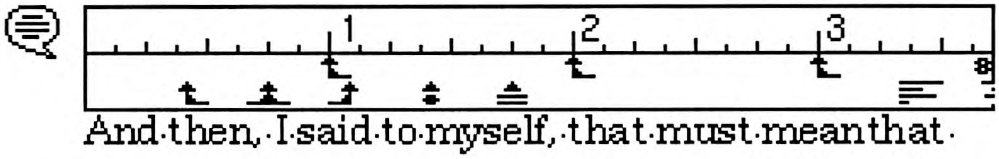
● Inches

○ Centimeters

○ Points

○ Picas

Comment rulers show above the comments. Notice that when rulers are showing, small dots appear where spaces are in the text.



And·then,·I·said·to·myself,·that·must·meanthat·

You can set, move, and remove tab stops (left, center, right, decimal point, and equal sign justified) on the ruler. You can also make the text left, center, or right justified. Use the square handle on the right of the ruler to move the right margin of the comment.

Comments can be split and joined with the Enter and Delete keys, respectively, which work intuitively.

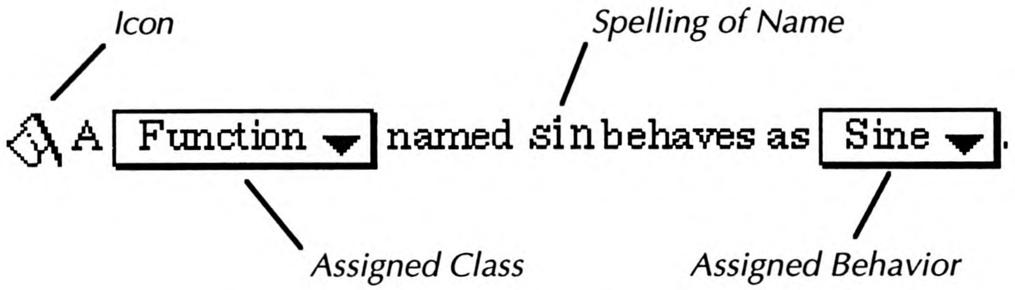
You can paste graphics stored on the clipboard into comment propositions. You can enter equations into comments just like you can enter them into assumption propositions. Equations, pictures, and text can all be mixed in a comment.

To take a snapshot of a graph, select it and choose **Copy as Picture** from the **Edit** menu. When you Paste it into a notebook, it is displayed in a comment proposition.

You can store any graphic from any source in a comment. For example, the following was taken from a supercomputer simulation of fluid dynamics:

Name Declarations

Name declaration propositions designate the class and behavior of each name in a theory. Names can consist of Roman characters, Greek characters, and the symbols ∞, ⟨, ⟩, ℏ, ℜ, ℑ, ℵ, and ′. Each name declaration proposition appears with a flag icon:



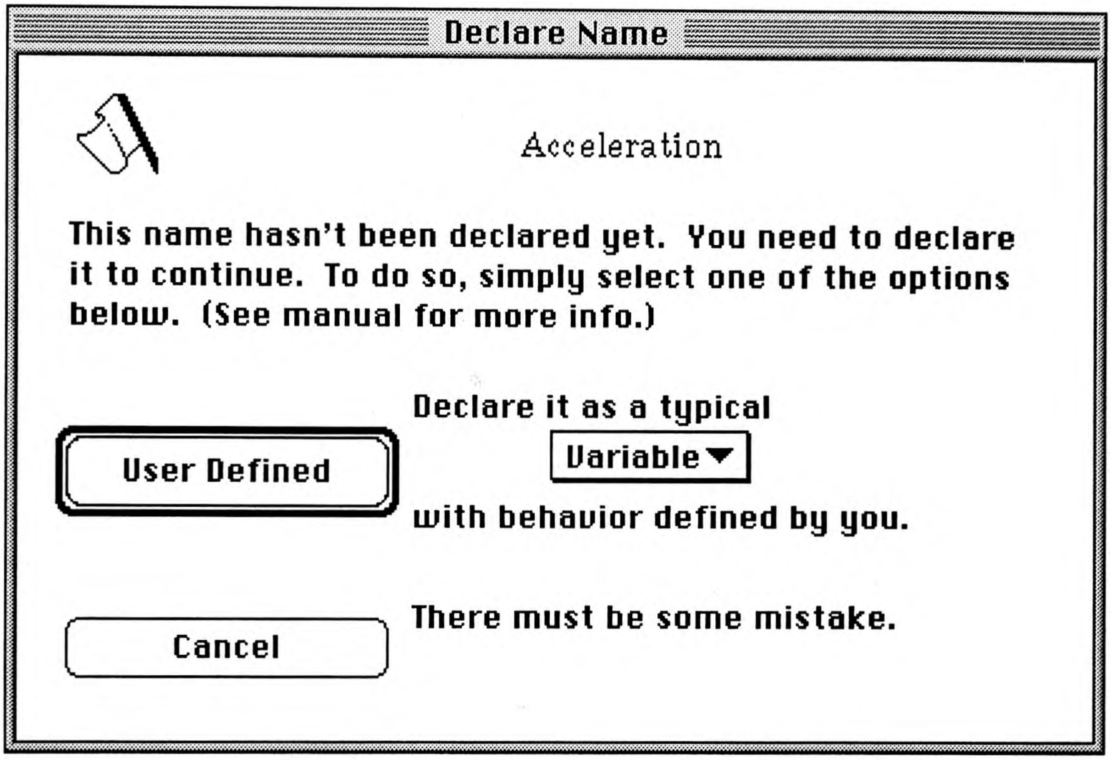
A Function ▾ named sin behaves as Sine ▾.

This proposition declares that the name "sin" is of class Function and has the predefined behavior called Sine. Names can be of class Constant, Variable, M-Linear Operator, D-Linear Operator, or Function. The behavior of a name can be one of sixty-three predefined behaviors or a behavior you define yourself. Defining the behavior of a name can be as simple as creating an equation using that name. The statement $x = 13$ defines the behavior of x.

When you create a new notebook from the New Notebook file, its name declarations are collapsed under the Declarations comment proposition. (If you move or delete this bundle, new name declarations are placed at the top of the current notebook.) Other name declarations are within the bundles under Declarations.

If a name has not been declared when it is first involved in a manipulation (or used to create a graph), a dialog appears asking you to declare the name. If your notebook has undeclared names, choose **Clarify** from the **Notebook** menu to bring up this dialog.

For example, if you use the name Acceleration in a notebook, the first time you manipulate an expression (or Clarify the notebook) the following dialog appears.



Acceleration

This name hasn't been declared yet. You need to declare it to continue. To do so, simply select one of the options below. (See manual for more info.)

Declare it as a typical Variable▼ with behavior defined by you.

There must be some mistake.

Accept the default (click User Defined or press Return) to declare Acceleration as a variable. Use the pop-up menu on the right of the dialog box to indicate the name's class (Constant, Variable, M-Linear, D-Linear, Function).

To edit a name declaration, open the Declarations comment (double-click on its icon). You can change the class or behavior by clicking on the pop-up menus in the name declaration proposition. You can also change the spelling of a name. Changing the spelling, class, or behavior of a name is immediately reflected throughout a notebook. However, changing the class or behavior of a name may invalidate existing derivations; manipulations are *not* re-executed.

There are sixty-three predefined names with sixty-three associated predefined behaviors. To use a predefined name, enter the name and accept the default behavior associated with it when the dialog box appears. To assign a predefined behavior to a different name, enter the predefined name associated with the behavior that you want, accept the default associated behavior, then change the name's spelling. For more information on predefined names and behaviors, see the Expressions section.

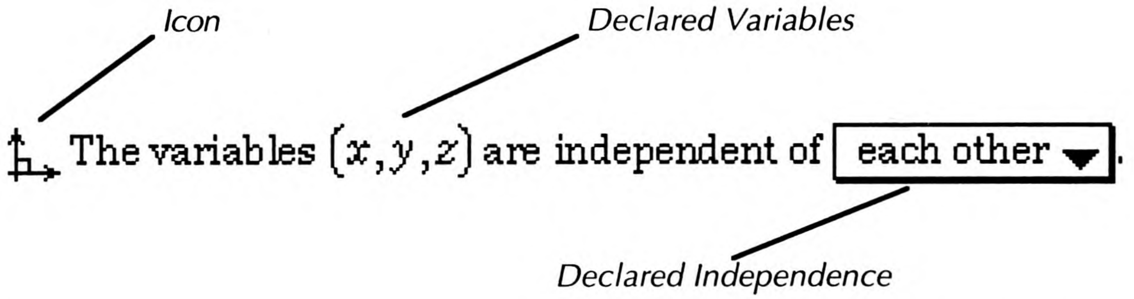
Wildcard variables are not names and should not be used in name declarations.

Independence Declarations

Independence Declaration propositions assert that certain variables are independent of each other (or are independent of all variables). If two variables are independent of each other, the derivative of one with respect to the other is zero; there is no relationship linking them together. By default, all variables in Theorist are dependent on one another.

Independence declarations can also declare that two M- or D-Linear Operators commute with each other. Declarations of this type are only used by the Commute manipulation, not Simplify.

Independence declarations are shown with an icon displaying two orthogonal axes:



The variables (x,y,z) are independent of each other ▾.

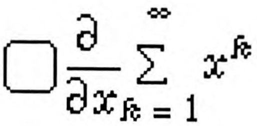
This proposition asserts that *x*, *y*, and *z* are independent of each other. Their derivatives with respect to each other are zero.

You should declare variables independent of each other for any situation that has multiple variables whose derivatives with respect to each other should be zero, such as:

- Spaces with multiple orthogonal dimensions
- Multiple integrations such as: $\int_0^1 \left(\int_0^1 xydx\right) dy$
- Two-dimensional (or higher dimension) Taylor series

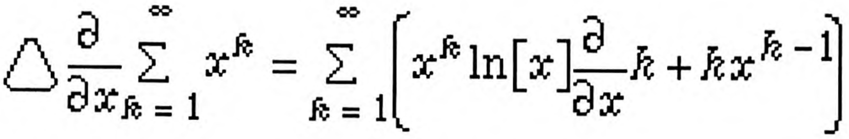
The pop-up menu at the end of the proposition can declare the given variables to be independent of each other (the default) or independent of all variables. If you use the Summation (Σ) and Pi Product (Π) ops, you may want to declare the iteration variable (usually *k*) to be independent of all variables, especially if you take derivatives of expressions containing the iteration variable.

For example, you should probably declare *k* independent of all variables for simplifying the following expression.



$$\square\frac{\partial}{\partial x}\sum_{k=1}^{\infty} x^{k}$$

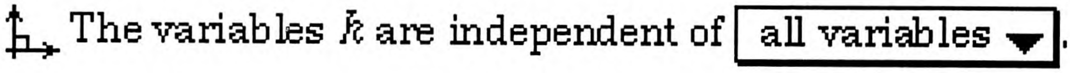
Without an independence declaration, the Simplify manipulation produces the following:



$$\triangle\frac{\partial}{\partial x}\sum_{k=1}^{\infty} x^{k} = \sum_{k=1}^{\infty}\left[x^{k}\ln[x]\frac{\partial}{\partial x}k + kx^{k-1}\right]$$

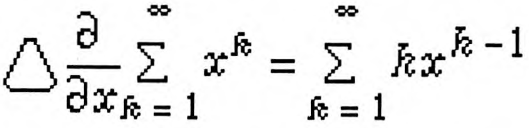
Please forgive the poor grammar; it's only a computer program.

If you declare k independent of all variables,



The variables k are independent of all variables ▾.

...the result is much more manageable:



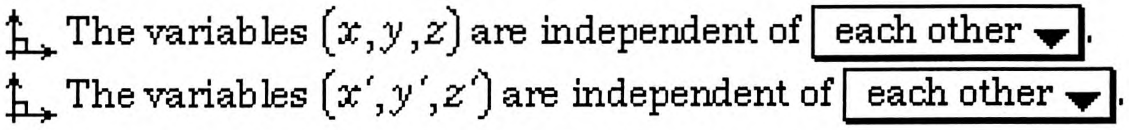
$$\triangle\frac{\partial}{\partial x}\sum_{k=1}^{\infty} x^{k} = \sum_{k=1}^{\infty} kx^{k-1}$$

For more information on the Summation and Pi Product ops, see the Expressions section.

Use independence declarations with caution. If x and y are declared independent, you can not create a graph of the relationship between these two variables, even if you explicitly state that $y = \sin(x)$. The dependent variable (y) is assumed to be a constant with respect to the independent variable (x); the graph produced is a straight line.

Independence declarations take precedence in any theory (or enclosed case theory).

A variable can not appear in more than one independence declaration (unless the declarations are in sibling case theories). For example, if you have two coordinate systems (x, y, z) and (x', y', z), where (x, y) and (x', y') are rotated around the z axis relative to each other, with z shared between the two coordinate systems, you can not make two independence declarations both listing z. Instead, create a variable z', make $z = z'$ a working statement, and create two independence declarations:



The variables (x, y, z) are independent of each other ▾.

The variables (x', y', z') are independent of each other ▾.

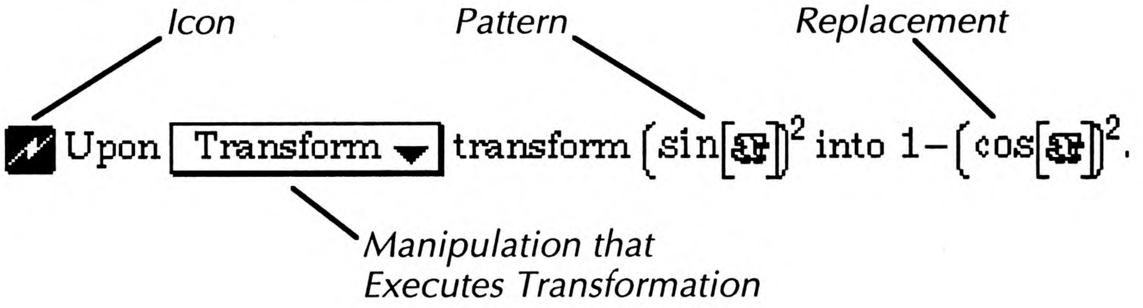
To create an independence declaration, choose **Notebook▸Insert▸ Independence Decl.**. If variables are selected when you create the declaration, they are inserted in the declaration.

TRANSFORMATION RULES

Transformation rule propositions describe transformations invoked when you perform a Transform manipulation. Transformation rules provide a means to extend the capabilities of Theorist. You can designate different transformation rules to execute when the following manipulations are performed:

- Transform
- Simplify
- Expand
- Collect (executes Expand rules in reverse)

Transformation rules are displayed with a lightning bolt icon:



Upon Transform ▾ transform $(\sin[x])^2$ into $1-(\cos[x])^2$.

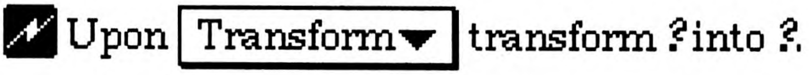
Use the pop-up menu to designate which manipulation invokes a particular transformation.

Each transformation rule describes the transformation of an expression or equation (the pattern) into another expression or equation (the replacement) that is logically equivalent. (Be wary of creating fallacious transformation rules which may either let you do powerful manipulations or let you shoot yourself in the foot.)

The pattern can contain regular variables or wildcard variables. Regular variables match the named variable explicitly, whereas wildcard variables match any variable of a given type, or even entire expressions. (See Wildcard variables in the Expressions section.)

Transformation rules are similar to—but more powerful than—the Substitute manipulation. All transformation rules in the current theory (and all enclosing theories) execute when you invoke the Transform manipulation.

To create a transformation rule proposition, choose **Notebook▸Insert▸Transform Rule**. If one or more equations are selected, one transformation rule is created from each equation. The left-hand side of an equation becomes the pattern and the right-hand side becomes the replacement. If no equations are selected, a transformation rule is created with two question marks:



Upon Transform▼ transform ?into ?.

Edit existing transformation rules as you would any other proposition.

For more information on transformation rules, see the Manipulations section.

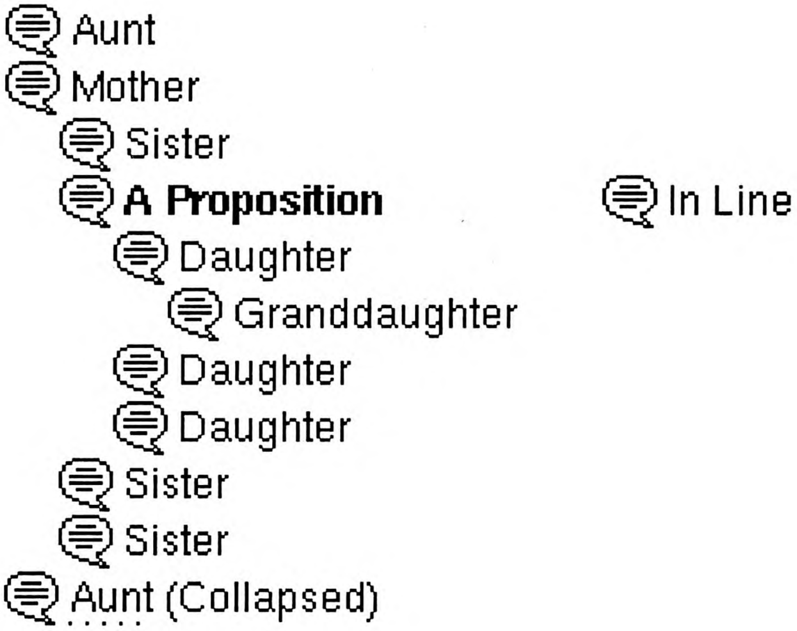
In some situations, more than one transformation rule that executes with the Simplify or Expand manipulation will match a selected expression. In these cases, rather than displaying a dialog box, the manipulation executes the first matching rule. The manipulation searches the current theory from top to bottom, then the parent theory from top to bottom, and so on.

Transformation rules that execute "Upon Simplify" are carried out *after* executing the Simplify manipulation, whereas transformation rules that execute "Upon Expand" are carried out *before* executing the Expand manipulation.

The Transform manipulation is not recursive. It executes only once on the outermost layer of an expression. If nested expressions can still be transformed, repeat the manipulation. To make a Transform manipulation recursive, change the pop-up menu to Simplify or Expand. Both of these manipulations execute recursively.

OUTLINING

All propositions in a notebook are arranged in a hierarchy. You can rearrange propositions as you would topics in a word-processing outliner. Each proposition can have sister, daughter, granddaughter, or inline relatives. The following example is created entirely from comment propositions to show the types of outline relationships:



- Aunt
- Mother
  - Sister
  - **A Proposition** In Line
    - Daughter
      - Granddaughter
    - Daughter
    - Daughter
  - Sister
  - Sister
- Aunt (Collapsed)

To move a proposition, select it (by clicking on its icon), then hold down the ⌘ key (Macintosh) or Ctrl key (Windows). The mouse pointer becomes a pointing hand. Use this pointing hand to click on the icon of a selected proposition and drag it to a new location. Possible release points highlight. Release the mouse where you want to put the proposition. You can also move selected propositions to the left or right by pressing ⌘[and ⌘] (Macintosh) or Ctrl[and Ctrl] (Windows).

You can rearrange propositions within any case theory by selecting and dragging. However, you must cut (or copy) and paste propositions to move them from one theory to another.

To regroup several propositions, shift-click to select more than one proposition, click on any of the selected propositions with the pointing hand, and move the group to a new location. The propositions are displayed in the order you selected them.

To collapse the daughters under a proposition, select the proposition and choose **Collapse** from the **Notebook** menu. A dotted line is displayed beneath the first line of a collapsed proposition. To expose a collapsed family, select the proposition and choose **Expose** from the

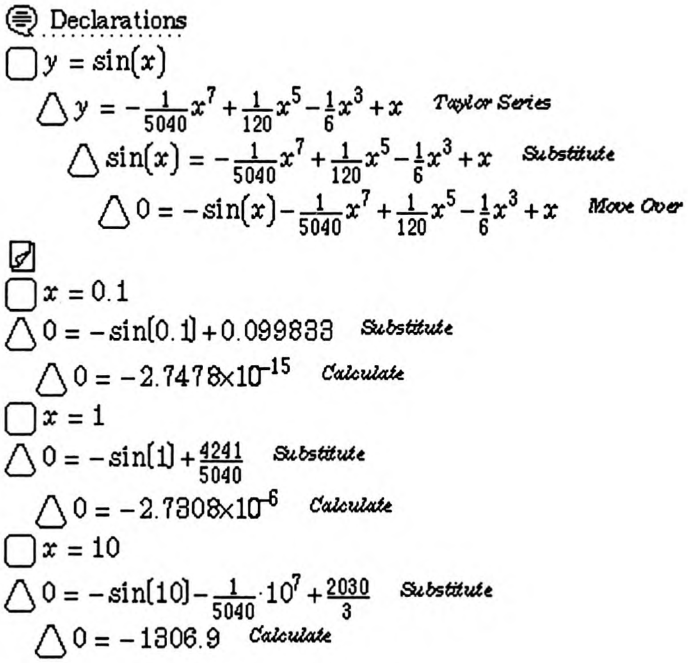
Notebook menu. You can also double-click on a selected proposition icon with the mouse pointer to collapse or expose a family of propositions.

Collapsed families are called "bundles." The notebooks on the Theorist disks contain a number of bundles with useful name declarations. To copy a bundle into your notebook, select it by its title (or main proposition) and use Copy and Paste.

To expose all propositions in a notebook, choose **Expose All** from the **Notebook** menu. Conversely, choose **Collapse All** to collapse all propo sitions in a notebook.

Page Break

Choose **Notebook▸Insert▸Page Break** to insert the page break proposition ☑ in the notebook. This forces the following propositions to start at the top of a new page when the notebook is printed. For example, consider the following notebook contents. (Note the location of the page break proposition icon.)



Declarations

□ $y = \sin(x)$

△ $y = -\frac{1}{5040}x^7 + \frac{1}{120}x^5 - \frac{1}{6}x^3 + x$ *Taylor Series*

△ $\sin(x) = -\frac{1}{5040}x^7 + \frac{1}{120}x^5 - \frac{1}{6}x^3 + x$ *Substitute*

△ $0 = -\sin(x) - \frac{1}{5040}x^7 + \frac{1}{120}x^5 - \frac{1}{6}x^3 + x$ *Move Over*

☑

□ $x = 0.1$

△ $0 = -\sin(0.1) + 0.099833$ *Substitute*

△ $0 = -2.7478\times10^{-15}$ *Calculate*

□ $x = 1$

△ $0 = -\sin(1) + \frac{4241}{5040}$ *Substitute*

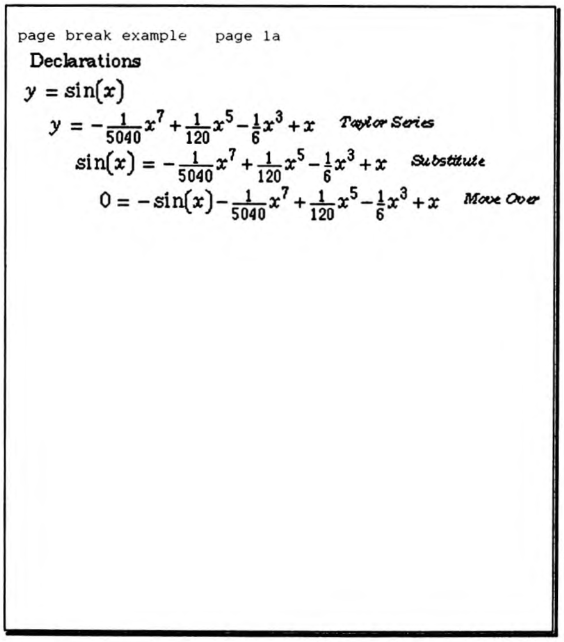
△ $0 = -2.7308\times10^{-6}$ *Calculate*

□ $x = 10$

△ $0 = -\sin(10) - \frac{1}{5040}\cdot10^7 + \frac{2030}{3}$ *Substitute*

△ $0 = -1306.9$ *Calculate*

Though it appears together in the notebook, it appears on two pages when printed.



page break example page 1a

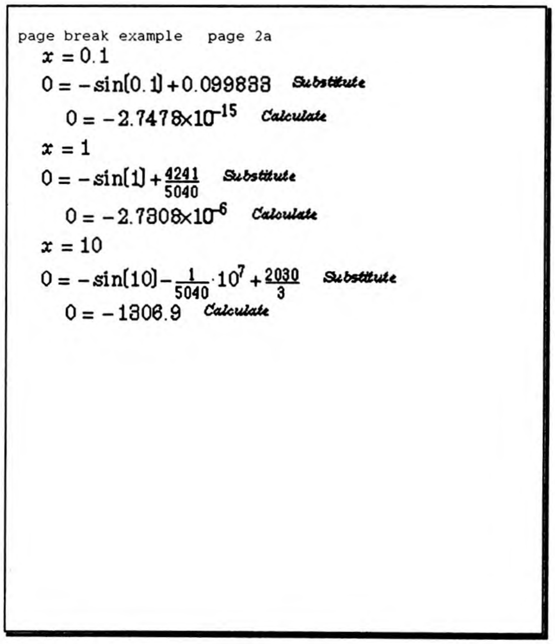
Declarations

$y = \sin(x)$

$y = -\frac{1}{5040}x^7 + \frac{1}{120}x^5 - \frac{1}{6}x^3 + x$ *Taylor Series*

$\sin(x) = -\frac{1}{5040}x^7 + \frac{1}{120}x^5 - \frac{1}{6}x^3 + x$ *Substitute*

$0 = -\sin(x) - \frac{1}{5040}x^7 + \frac{1}{120}x^5 - \frac{1}{6}x^3 + x$ *Move Over*



page break example page 2a

$x = 0.1$

$0 = -\sin(0.1) + 0.099833$ *Substitute*

$0 = -2.7478\times10^{-15}$ *Calculate*

$x = 1$

$0 = -\sin(1) + \frac{4241}{5040}$ *Substitute*

$0 = -2.7308\times10^{-6}$ *Calculate*

$x = 10$

$0 = -\sin(10) - \frac{1}{5040}\cdot10^7 + \frac{2030}{3}$ *Substitute*

$0 = -1306.9$ *Calculate*

Expressions

Expressions are Theorist's fundamental objects. This section begins with a discussion of the class scheme Theorist uses to manage expressions. The section then presents all types of expressions and expression components:

- Numbers
- Names (including all predefined names)
- Ops
- Wildcard variables
- Collapsars
- Parentheses
- Undefined values

Class

Theorist can represent a wide variety of objects, ranging from simple numbers to abstract operators for which there is no concrete numerical value. Theorist uses a class system to determine what rules to use when manipulating an object. Each object belongs to one of five classes. This section discusses the five classes:

- Constant
- Variable
- M-Linear Operator
- D-Linear Operator
- Function

The class of an expression determines how it can be used. For example, D-Linear Operators are multiplied onto an argument, whereas functions contain or enclose their arguments, and constants have no arguments at all. See the Summation op (a D-Linear Operator) and Sine (a function) described later in this section.

You must declare the class of named objects the first time you manipulate an expression containing a new name or create a graph that uses the name. You can also declare any undeclared names by choosing the **Clarify** item from the **Notebook** menu. Each object is manipulated according to the rules of its declared class. The default class of a new object is Variable. However, you may get erroneous results if this is not the class you intend.

For example, if you create two matrices A and B and accept the default class Variable (rather than declaring them to be M-Linear Operators), the expressions AB and BA are equivalent. But they are not necessarily equivalent because matrix multiplication is not commutative. To avoid errors of this type, you must assign named objects to the correct class.

To see the class of any expression, select the expression, then choose **Get Info** from the **Notebook** menu.

The class of a particular expression depends on what the expression is, and the classes of any subexpressions it contains. For example, if a Variable x is added to a matrix, an M-Linear Operator, the result is an M-Linear Operator. However, if the matrix contains one or more D-Linear Operators, the object as a whole is a D-Linear Operator.

Each predefined name has a preset class. For example, π is always of class Constant and arctan is always of class Function.

Constants

A constant is a scalar (a single value) whose derivative is zero. Constants do not need to be defined to have a specific value. You can treat them as abstract unknowns.

If you create a working definition for a constant named *c,* it has the value you assign to it. Its actual value may not be constant (it may depend upon other variables) but it is treated as a constant for manipulations and for creating graphs.

All numeric values are constants as well as certain predefined names (e.g., π, e, ∞, i). All expressions consisting entirely of constants (including functions of constants) are also constants.

The following are all of class Constant:

$$5 \qquad \pi \qquad \sqrt{5} \qquad \infty \qquad \arctan(\pi^2)$$

Variables

A variable can take on the value of any integer or real or complex number. A variable's derivative is not necessarily zero. Variable multiplication is commutative. The value of a variable can not be a matrix or derivative.

The relationship between any two variables is, by default, undefined. However, you can define either in terms of the other with an equation or a working statement, or you can declare them to be independent with an independence declaration.

All names declared as variables are of class Variable, as are all expressions made up of constants and variables.

The following are all (usually) of class Variable:

$$x \qquad y \qquad t \qquad \sqrt{xy}$$

M-Linear Operators

An M-Linear Operator is a matrix or behaves like a matrix in certain ways (e.g., Hermitian operators). Matrix multiplication is associative but not commutative. However, multiplication of a matrix and a variable (or a constant) is commutative.

As an example of non-commutative algebra, if A and B are declared to be M-Linear Operators, and you Expand $(A + B)^3$ the result is:

$$A^3 + BA^2 + B^2A + A\,BA + B^3 + A\,B^2 + A^2B + BA\,B$$

Because multiplication is not commutative for M-Linear Operators, the terms do not recombine when simplified.

Note: If you declare the relative independence of A and B (with an independence declaration) you can reduce an expression of this form. However, you must Commute each expression by hand. Then click the palette button (or choose **Simplify** from the **Manipulate** menu) to reduce the expression.

All matrices with constant, variable, and M-Linear elements are M-Linear Operators. A matrix with one or more D-Linear Operators as an element is a D-Linear Operator.

The following are all of class M-Linear Operator:

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \qquad \begin{pmatrix} x & y \\ z & 0 \end{pmatrix}\begin{pmatrix} a & b & c \\ d & e & f \end{pmatrix} \qquad \begin{pmatrix} 0 & y \\ -y & 0 \end{pmatrix}^2 + 1\,.$$

D-Linear Operators

A D-Linear Operator is a derivative operator, or an operator that behaves like one. Multiplication of a D-Linear Operator and any other object (other than a constant) is not necessarily associative nor commutative.

D-Linear Operators are multiplied onto the expression immediately preceding them. The three most common D-Linear Operators are the predefined name *d,* the partial derivative op ($\partial / \partial x$) and the summation op (Σ). These three all operate by varying the value of particular variables in the expression immediately following them.

D-Linear Operators do not commute with variables, although they do commute with constants. For example:

$$\sum_{k=0}^{10} 5x^k = 5 \sum_{k=0}^{10} x^k$$

However:

$$\sum_{k=0}^{10} kx^k \neq k \sum_{k=0}^{10} x^k$$

If the right-most factor in a product is a D-Linear Operator, then the expression as a whole is of that class.

The following are all of class D-Linear Operator:

$$x + \frac{\partial}{\partial x} \qquad \frac{\partial}{\partial x} + x \qquad y \frac{\partial}{\partial x}$$

However, the following are all of the same class as *y:*

$$\frac{\partial}{\partial x} y \qquad \left(x + \frac{\partial}{\partial x}\right) y$$

If you multiply a D-Linear Operator onto a constant, variable, or matrix, then the class of the new expression is of class Constant, Variable, or M-Linear Operator. That is, if the D-Linear Operator is on the left-hand side of a product, the term as a whole is of the same class as the partial term on the right.

Functions

Functions enclose one or more expressions as arguments. (Actually, functions are just names of class Function. The function op, symbolized by a pair of parentheses, encloses the arguments.) Each function maps an input argument to an output argument. Theorist provides many predefined functions. These are listed later in this section under Predefined Names.

To create a function of your own design, write an equation that defines its value with wildcard variables as arguments. Then choose **Make Working Stmt** from the **Notebook** menu.

For example, you can define the function $f(x)$ to be:

$$f(x) = \frac{2^{x} + 2^{-x}}{e}$$

You can also create functions with more than one argument:

$$f(x,y) = \frac{\sin(x) + \sin(y)}{2}$$

You can also pass a function a vector of elements as a single variable. In the following example, p is a two-element vector:

$$g(p) = \frac{\sqrt{p_1} + \sqrt{p_2}}{2}$$

See the Special Functions directory in the Mathematics directory on your Theorist disks for examples of functions and their definitions.

NUMBERS

Theorist can display and manipulate numeric values in decimal or scientific notation. Type a period for a decimal point, and "e" or "E" to enter a number in scientific notation. You can precede the exponent with a plus or minus sign. For example, to enter 1.23×10^8, type **1.23e8** or **1.23e+8**. You can also raise any number to a power using the circumflex (^) character.

Negative numbers are preceded by a minus sign that encloses the number. If you select the minus sign, you select both the number and the sign. The negation op is perhaps the simplest op. All other ops work in a similar fashion. Ops enclose one, two, or more expressions, making those expressions a single object. To change the sign of a number, select the whole number and type **-**, or insert the cursor before the number and enter a minus sign.

Complex numbers (e.g., $4 + 3i$) are represented as a symbolic sum in rectangular coordinates. Numeric matrices are represented as matrices with numeric components. To create a matrix, type an opening parenthesis and separate elements with commas, rows with semicolons. Infinity (∞) and negative infinity (−∞) are also supported.

The largest number less than infinity that can be represented is about 1.7976×10^{308}. The smallest number greater than zero that can be represented is about 1.9407×10^{-324}. Overflow and underflow yield infinite and zero values, respectively.

An undefined number is displayed as a question mark (?). It is roughly equivalent to "not a number" or NaN values as specified by the IEEE floating point standard. For example, 0/0 = ?.

Theorist stores numeric values with fifteen digits of precision, enough to measure the age of the known universe to within a second, or to measure the radius of the sun to within one angstrom. You can set the displayed precision of numeric values with the **Display Precision** submenu of the **Notebook** menu; this setting has no effect on the internally maintained precision.

If you copy an expression out as text (with **Copy** from the **Edit** menu), the numbers contain all fifteen digits of precision. If you copy an expression out as a picture (with **Copy as Picture** from the **Edit** menu), you get a picture of the expression as it appears on screen.

Names

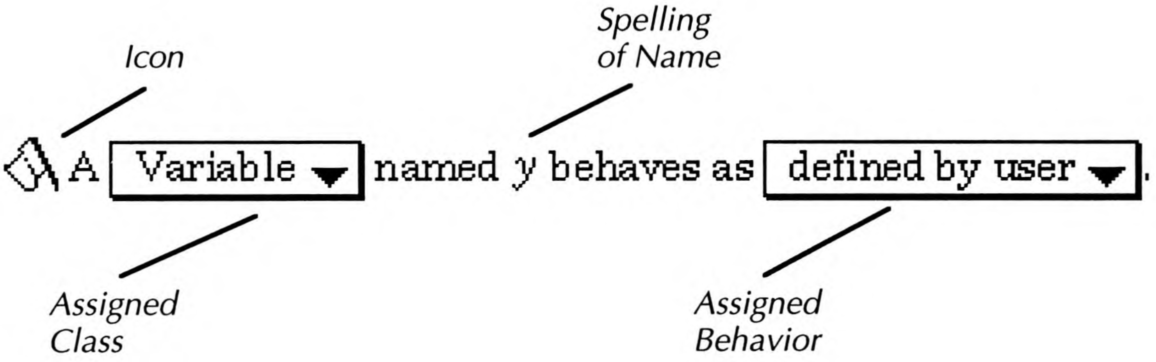
A name is one or more characters that represent a mathematical entity. A name can represent an object of any class (Constant, Variable, M- or D-Linear Operator, or Function).

Simple names can contain one or more of the following:

- Upper- and lowercase Roman letters
- Upper- and lowercase Greek letters
- The symbols ∞ ∇ ⟨ ⟩ ℑ ℜ ℵ ′ ℏ

Names can not contain digits or any other characters.

The first time you manipulate an expression containing a new name (or create a graph from the expression), a dialog appears requesting a declaration. Once created, a name declaration appears as a proposition with a flag icon:



A Variable ▾ named y behaves as defined by user ▾.

A name declaration specifies the spelling of the name, its class, and its behavior. The class must be Constant, Variable, M- or D-Linear, or Function. You define the behavior with a working statement or you can select a predefined behavior.

By default, all new names are placed in the Declarations comment proposition at the top of each notebook. If you do not have a comment proposition at the top of a notebook (named Declarations, or anything else), newly declared names appear at the top of the current notebook.

The behavior of all names is either predefined or defined by you. The predefined names and their associated behaviors are listed later in this section.

Changing a name declaration instantly updates all occurrences of that name throughout a notebook. Names can be spelled any way you want (as long as no two names are spelled the same way). Names are case sensitive (e.g., Log ≠ log).

To edit a name or change its behavior or class, select it and choose **Get Info** from the **Notebook** menu. A dialog box appears. Click on **More Info** to see the name declaration. If you click on the declaration in the dialog box, the dialog box disappears, the Declarations comment is opened (if necessary), and the selected name declaration is highlighted. Edit the name declaration as you want. Or you can open the Declarations comment directly and search for the name there.

Subscripted Names

Use subscripts to create families of names. For example, instead of using variables x, y, and z, you can use variables x_1, x_2, and x_3. Subscripted names inherit the class of the un-subscripted name, but they can be used in distinct working statements and independence declarations.

Subscripts can be either integers in the range ±32,767 or names. Names must be previously declared and user defined. The subscript's class is irrelevant; the name merely serves as a unique identifier. A name can have more than one subscript. A name can have a list (row vector) as a subscript, and a subscripted name can have a second subscript, or a combination. For example, you can use the following sets of variables (which are not equivalent):

- $x_{(1,1,1,1)}, x_{(1,1,1,2)}, x_{(1,1,1,3)} \cdots$
- $x_{1_A}, x_{1_B}, x_{1_C}, \cdots$
- $x_{A_{(1,1,1,1)}}, x_{A_{(1,1,1,2)}}, x_{A_{(1,1,1,3)}}, \cdots$

Note: The name $A_{(1,2)}$ can refer to a component of a matrix *or* to a unique name. If A is a matrix, the value of the subscript (1, 2) determines the matrix element. (See the Index op, below.) If A is not a matrix, $A_{(1,2)}$ is a unique name.

If the subscript is a name, and it has a defined value in the range ±32767, Theorist rounds this number to the nearest integer and interprets the subscript numerically.

For example, if x is of class Variable, working statements for x_2 and x_n specify two different variables. If you add another working statement that $n = 2$, the two names become equivalent.

You can also use subscripted names with functions (e.g., $J_0(x)$, $J_1(x)$, $J_2(x)$, ...).

Note

Log and natural log (ln) functions interpret subscripts as the base to use for that function and not as unique names. The base can be any real number or expression. Any name assigned the predefined behavior of the log or natural log functions uses subscripts in this fashion.

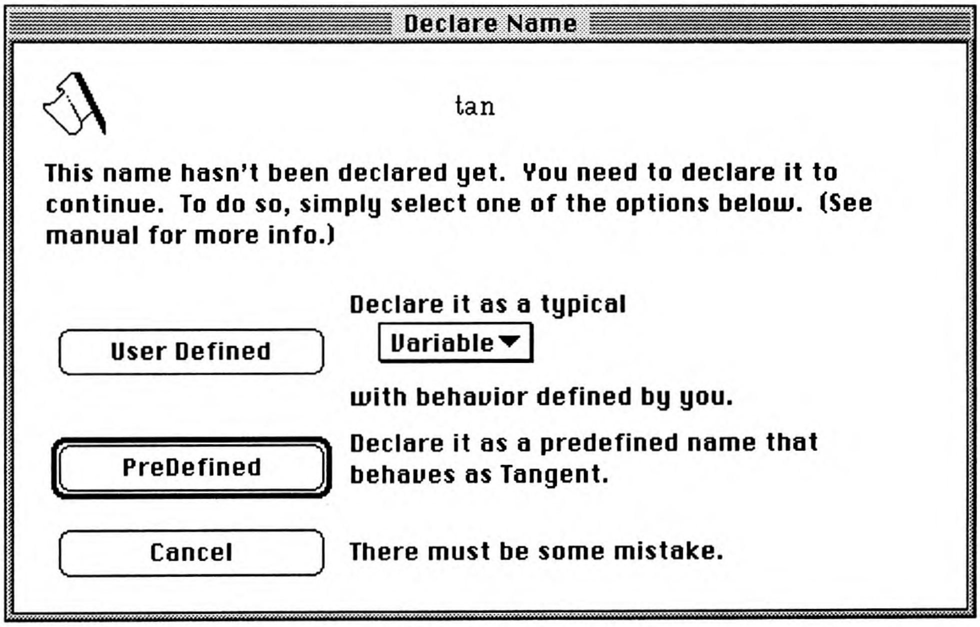
Predefined Names

Theorist provides sixty-three (63) predefined names, each with an associated behavior.

To use any one of the other predefined names, include it in an expression and when asked to declare it (when you first manipulate the expression or create a graph that uses the expression), accept the default behavior presented in the dialog box.

Predefined names are not "reserved" words. You can associate any name with any predefined behavior, and you can associate a predefined name with a different behavior, either user defined or predefined (although this last is not recommended).

To assign a predefined behavior to a name other than the associated predefined name, proceed as follows. Use the predefined name in an expression and manipulate the expression (or choose **Clarify** from the **Notebook** menu). A dialog box appears, such as:



Declare Name

tan

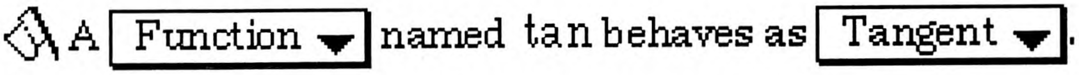
This name hasn't been declared yet. You need to declare it to continue. To do so, simply select one of the options below. (See manual for more info.)

Declare it as a typical Variable▼ with behavior defined by you.

Declare it as a predefined name that behaves as Tangent.

There must be some mistake.

Accept the default, associating the predefined name tan with the predefined behavior Tangent. Open the Declarations comment at the top of the Notebook, and find the newly created name declaration:

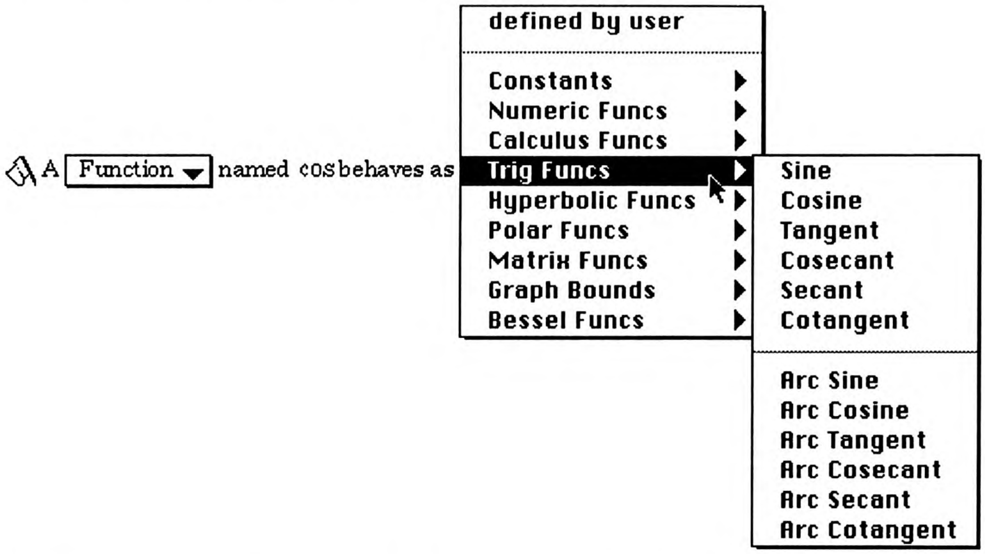


A Function▼ named tan behaves as Tangent▼.

Select the name "tan" and change it to "Tan" or "trigtan" (or whatever) by typing the new name. This change is immediately reflected throughout the notebook.

You can edit existing names in the same way (by opening the Declarations comment), or you can select the name and choose **Get Info** from the **Notebook** menu to find the name declaration.

To change the behavior associated with an existing name, adjust the pop-up menu at the end of the name declaration:



A Function named cos behaves as

The following table lists all sixty-three predefined names and their associated behaviors.

Predefined Name	**Description**
Constants	
∞	Infinity
π	Pi: 3.14159265358979 (actual value used)
i	Square root of negative one ($\sqrt{-1}$)
e	Base of natural logarithms, 2.71828182845905 (actual value used)
Numeric Functions	
round( )	Rounds argument up or down to nearest integer
floor( )	Rounds argument down to nearest integer
ceiling( )	Rounds argument up to nearest integer

mod( )

Modulus; mod takes two arguments, x and y, and evaluates to the remainder after dividing x by y. The result is always between zero (inclusive) and y (exclusive).

Re( )

Evaluates to the real part of a complex argument

Calculus Functions

log( )

Common logarithm (base 10); defined over (almost) entire complex plane; use subscripts to specify other bases: type **log_** and the base

ln( )

Natural logarithm (base e); defined over (almost) entire complex plane; use subscripts to specify other bases: type **ln_** and the base

exp( )

Value of e raised to a given power ($\exp(x) = e^x$); defined over entire complex plane

Γ ( )

Gamma function ($\Gamma(x+1) = x!$); defined over entire complex plane (infinite for non-positive integers)

d

Differential operator that takes total derivatives

Trig and Arc Trig Functions

sin( ) cos( )
tan( ) cot( )
sec( ) csc( )

Trig functions that take radian arguments; to use degrees, divide argument by π and multiply by 180; defined over (almost) entire complex plane

arcsin( ) arccos( )
arctan( ) arccot( )
arcsec( ) arccsc( )

Anti-Trig functions that return radian results; to get degrees, multiply argument by π and divide by 180; defined over (almost) entire complex plane

Hyperbolic Functions

sinh( ) cosh( )
tanh( ) coth( )
sech( ) csch( )

Hyperbolic functions that take real or complex arguments; defined over (almost) entire complex plane

arcsinh( )
arccosh( )
arctanh( )
arccsch( )
arcsech( )
arccoth( )

Anti-Hyperbolic functions that take real or complex arguments; defined over (almost) entire complex plane

Bessel Functions

I( ) Ai( )
J( ) Bi( )
K( ) Y( )

Bessel, modified Bessel, and Airy functions

Polar Functions

FromPolar( )
ToPolar( )

Functions that convert a 2-vector between polar (r, θ) and rectangular coordinates; r is the radius; θ, the angle, is zero at the positive x axis and increases counter-clockwise

FromCylindrical( )
ToCylindrical( )

Functions that convert a 3-vector between cylindrical (r, θ, z) and rectangular coordinates; r and θ are interpreted as in FromPolar; z is the height and is unchanged by this function.

FromSpherical( )
ToSpherical( )

Functions that convert a 3-vector between spherical (r, θ, φ) and rectangular coordinates; r is the radius; θ starts at the top of a sphere (the “north pole”), increases to $\pi/2$ at the equator, and equals π at the bottom (“south pole”); φ starts at the positive x axis, increases to $\pi/2$ at the positive y axis, equals π at the negative x axis.

Matrix Functions

RowsOf( )

Returns the number of rows, given a matrix argument

ColumnsOf( )

Returns the number of columns, given a matrix argument

Graph Bounds

left
right

Horizontal (first coordinate, usually x) minimum and maximum for current 2-D graph, for rectangular coordinates; valid only in 2D graph theories

bottom
top

Vertical (last coordinate, usually y or z) minimum and maximum for current 2-D or 3-D graph, for rectangular coordinates; valid only in graph theories

west
east

First coordinate (usually x) minimum and maximum for the current 3-D graph, for rectangular coordinates; valid only in 3-D graph theories

south
north

Second coordinate (usually y) minimum and maximum for the current 3-D graph, for rectangular coordinates; valid only in 3-D graph theories

radius

Maximum radius in all directions; used for polar, cylindrical and spherical plots; valid only in 2-D and 3-D graph theories

Ops

Ops connect expressions in an equation. If expressions are nouns in mathematical sentences, then ops are the verbs. In the expression $a + b$, the Addition op connects the two expressions a and b to create a new expression.

Ops enclose expressions. Unary ops enclose a single expression; binary ops enclose two expressions. Two ops (+ and ×) are n-ary: they can enclose any number of expressions (e.g., $1 + 2 + 3 + 4 + 5$). When you select an op, you also select the expression or expressions it encloses.

Ops do not perform calculations or manipulations. For example, there is no op that expands a polynomial. To perform this function, select the polynomial, then click on the palette (or choose **Expand** from the **Manipulate** menu). Manipulations (such as Expand and Calculate) are actions, whereas ops describe the relationship between names and quantities. Nothing happens spontaneously when you enter an expression or equation; you have to invoke a particular manipulation.

All ops can be entered from the keyboard or the palette. For detailed examples of entering and modifying expressions, see the Editing section. In this section, keyboard examples are given for entering the more complex ops.

The following table lists all twenty-three ops. After the table, each op is described individually.

Op	Notation
Addition	+
Negation	–
Subtraction	–
Multiplication	*
Division	/

Op	Notation
Equals	$=$
Relational	$\neq, <, >, \leq, \geq$
Power	x^8
Index (Subscript)	x_8
Absolute Value	$\lvert x \rvert$
Square Root	$\sqrt{x}$
Adjoint	$x^{\dagger}$
Matrix (Vector)	$\begin{pmatrix} 1 & 2 \\ 3 & x \end{pmatrix}$
Dot Product	$\bullet$
Cross Product	$\times$
Partial Derivative	∂
Integral	$\int_0^{-\pi} x^2 \, dx$
Summation	Σ
Pi Product	Π
Conditional	$\begin{cases} 0 & (x < 0) \\ x^2 & (x \geq 0) \end{cases}$
Evaluate At	${}^{x=-1}_{x=1}\left[\frac{1}{3} x^3 \right.$
Range	$-1...1$

Addition

$x + 5$

The Addition op (+) can add together any number of terms from 2 to 32,000. Arguments can be any expressions of class Constant, Variable, M- or D-Linear Operator.

You can add a scalar and matrix only if the matrix is square. Theorist automatically Expands it to a diagonal matrix by multiplying it by an identity matrix.

$$\begin{pmatrix} 1\,2\,3 \\ 4\,5\,6 \\ 7\,8\,9 \end{pmatrix} + x = \begin{pmatrix} x+1 & 2 & 3 \\ 4 & x+5 & 6 \\ 7 & 8 & x+9 \end{pmatrix}$$

Negation
$-x$

The Negation op (–) is a unary operator that encloses its one operand. Arguments can be any expressions of class Constant, Variable, M- or D-Linear Operator.

Subtraction
$x - y$

The Subtraction op (–) is actually a combination of the addition and negation ops. Arguments can be any expressions of class Constant, Variable, M- or D-Linear Operator.

Multiplicat-ion
$5x$

The Multiplication op (*) can multiply together any number of factors from 2 to 32,000. The Multiplication op performs different functions, depending upon the class of its operands.

Operand Class	Action
Constant	Ordinary multiplication
Variable	Ordinary multiplication
M-Linear Operator	Matrix multiplication (components must be numeric for calculations); or any abstract operation you define
D-Linear Operator	Derivative or summation; or any abstract operation you define

Multiplication of two M-Linear Operators (e.g., matrices) is not necessarily commutative ($AB \neq BA$). Multiplication of a D-Linear Operator (e.g., d, ∂) and any other operator (other than a constant) is not necessarily commutative or associative: $(AB)C \neq A(BC)$.

To enter $5x - xy$, type **5*x-x*y**. You can also use spaces in place of the asterisks. In some cases, simple concatenation indicates multiplication. To enter $5x$, type **5x**. To enter $x5$, type **x5**.

Division
$\frac{5}{x}$

The Division op (/) is a binary op that can be used with integers, real and complex numbers, and matrices. When you type a slash, Theorist creates a fraction. In the case of matrices (or other M-Linear Operators), A/B is interpreted as $B^{-1}A$. Division by zero almost always yields infinity (∞). Indeterminate forms (e.g., 0/0, ∞/∞) yield a question mark (?).

Equals

$x = 5$

The Equals op (=) has three uses. Most often, it is used to create an equation, algebraic statement, or working statement. You can also use the Equals op to create relational expressions that can be embedded in other expressions (see below). The Equals op is also used as part of the Evaluate At op (see below).

Expressions can only contain one equals op (unless it is used as part of a relational expression or Evaluate At op).

Note: Theorist is non-procedural. The equals op (like all ops) states a relationship, and is not a command. You do not use the equals op to calculate the right-hand side of an equation and store the value in a variable on the left, as you would in many programming languages.

Relational

$x \neq 5$
$x < 5$
$x > 5$
$x \leq 5$
$x \geq 5$

Five Relational ops are supported in addition to the equal sign: <, >, ≤, ≥, and ≠. They evaluate to 1 or 0, depending on whether the proposition is true or false, respectively. Arguments can be integers or real or complex numbers.

Use Relational ops (except for ≠) to set limits on the value of real variables in working statements. For example, if x should never assume values outside the range –1 to +1, enter the inequalities "$x \geq -1$" and "$x \leq 1$", and make each of them working statements. These restrictions are used to create graphs and for other purposes.

Relational ops can also be used as constraints during automatic case generation. (See the Manipulations section.)

To enter the Relational ops, type **<** , and **>** from the keyboard. Enter these with **<>**, **>=**, and **<=** (as is the convention in Pascal). (Macintosh users can also press option =, option >, and option < for ≠, ≥, and ≤.)

Function

$\sin(x)$

The Function Call op is an invisible op that connects a function to its argument. You create a Function Call op by entering a function name and an opening parenthesis. A vector argument is treated as multiple arguments; type commas between arguments.

Do not create a function with the same name as a variable (e.g., $y = y(x)$). If a function always takes the same argument, state this as an equation between variables instead. For example, if you want to express the relationship $y(x) = \sin(x)$, and you never plan to use the function with any other argument (e.g., $y(z)$, or $y(t)$),write this as an equation in two variables: $y = \sin(x)$, rather than defining y as another name for the sine function.

If you raise a function to a power, enclose the function in parentheses. For example, write $[\sin(x)]^2$ rather than $\sin^2 x$. The function as a whole is raised to the power of the exponent. Similarly, use the function $\arcsin(x)$ rather than $\sin^{-1}(x)$. If you use a function of the form $\sin^2 x$ in an expression, the function converts to the more literal form $(\sin[x])^2$ the next time you execute the Simplify manipulation. Expressions of the form $\sin^{-1}(x)$ simplify to $\arcsin(x)$.

Power

x^8

The Power op always has two operands, the base and the exponent. For integer exponents, the base is multiplied by itself a number of times as indicated by the exponent. The base can be any scalar or matrix (or any other operator, for symbolic operations), or any sum of such expressions.

If the exponent is an integer, the base can be of class Constant, Variable, or M- or D-Linear. If the exponent is not an integer, the base can only be of class Variable or Constant. If the exponent is a fraction or an irrational or complex number, the result can, in theory, have several different values. However, Theorist returns only one, the principal value.

If the base is positive, and the exponent is real, the result is always positive. If the base is negative or complex, the result may also be negative or complex.

To enter x^8, type **x^8**, or **x**8**.

Index or Subscript

x_8

The Index or Subscript op can be applied to a name, a matrix, or a vector. If applied to a matrix or a vector, it indicates an element or row. If applied to a name, it creates another name of the same class.

If you create the indexed name P_n, this name is independent of the names P and n. However, if P_n, has not been defined, Theorist uses any available information about P, possibly selecting the n-th element of P.

In most cases, the Index op is used for matrix and vector element selection. For example, if M is a matrix, then M_3 evaluates to the third row of M, a vector. The name $M_{(2,3)}$ refers to the element of M in the second row of the third column of M. If V is a row or column vector, then V_n selects the n-th element of V. If S is a scalar, then $S_{(m,n)}$ equals zero if $m \neq n$, or is the value S if $m = n$. (An indexed scalar is multiplied by an identity matrix.)

Index numbering starts at one. Indices that are non-positive, not real, or greater than the number of elements in the row or column of the matrix, yield either an error or the unknown value (?). Real-valued indices are rounded off to the nearest integer.

To enter M_3, type **M_3**. (Use shift - to enter the underbar.)

Absolute Value

$|x|$

The Absolute Value op returns the absolute value of scalar arguments, the length of vector arguments, and the determinant of matrices.

To enter $|x|$, type **| x**. Do not type the closing vertical bar. As an alternative, type **abs(x)**.

Factorial

$x!$

The Factorial op can be applied to all scalars. For positive integers, it returns the product of all smaller integers down to 1. For other values it uses the Gamma function.

To enter $n!$, type **n!**.

Square Root

$\sqrt{x}$

The Square Root op can be applied to all scalars. It is equivalent to the power op (see above) with an exponent of 1/2. If the argument is real and positive, it returns a positive result.

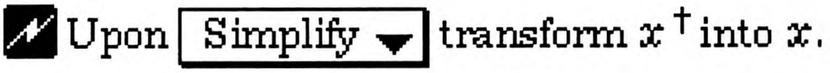
To enter $\sqrt{x}$, type **\x**. As an alternative, type **sqrt(x)**.

Adjoint

$x^{\dagger}$

The Adjoint op creates the transpose of a real matrix, the conjugate of a complex number, or the adjoint of a complex matrix (the transpose of the matrix and the conjugate of all elements). It has no effect on self-adjoint entities. However, in symbolic manipulations, it may not be apparent that a given entity is self-adjoint.

To declare that a given expression is real or self-adjoint, use a transformation rule such as:



Upon Simplify ▾ transform $x^{\dagger}$ into x.

To enter $x^{\dagger}$, type **Adjoint(x)**.

Matrix

$\begin{pmatrix} 1 & 2 \\ 3 & x \end{pmatrix}$

The Matrix op can hold up to 32,000 rows and up to 32,000 columns. A vector is a matrix with only one row or column, and can hold two to 32,000 elements. (In practice, the size is limited by available memory.) A one-by-one matrix can not exist. Matrix and vector elements must be scalars for numeric manipulations. For symbolic manipulations, matrix and vector elements can be any object.

Matrix addition and multiplication work according to conventional rules. When a scalar is added to a matrix, the scalar is multiplied by an appropriate identity matrix. Scalars can only be added to square matrices.

Taking the reciprocal of a matrix with the division or power op yields the matrix inverse (only on square matrices). Taking the absolute value of a matrix yields the determinant of a square matrix, or the "length" of a vector (the square root of the sum of the squares of the components). Taking the adjoint of a matrix transposes it, and then takes the adjoint of all of its elements.

For symbolic manipulations, matrices can contain other matrices or other M- or D-Linear Operators as elements.

To enter:

$$\begin{pmatrix}1&2&3\\4&5&6\\7&8&9\end{pmatrix}$$

...type **(1,2,3;4,5,6;7,8,9)**. Use commas to separate elements in a row, and semicolons to separate rows. To add rows to a matrix, insert the cursor after an element and type a semicolon.

Dot Product $x \bullet r$

The Dot Product op returns the pairwise product of the elements of two vectors. The two must have the same number of elements, but can be either row or column vectors. For example:

$$(a, b, c) \bullet (a, b, c) = a^2 + b^2 + c^2$$

The dot product of two vectors is independent of their orientation; either can be a row or column vector. The result is always a scalar.

In contrast, normal multiplication of two vectors is only allowed if they are of different types (one is a column vector, one is a row vector). If the first is a column vector, the result is the outer product of the two vectors (a matrix). If the first is a row vector, the result is the inner product (a scalar).

To enter $x \bullet r$, type **x*@r**.

Cross Product $x \times r$

The Cross Product op takes two 2-vectors or two 3-vectors (whether row or column) and returns the cross product. Vector elements can be of class Constant, Variable, or M- or D-Linear. The result is a scalar for 2-vectors or a 3-vector for 3-vectors. For example:

$$(a, b, c) \times (d, e, f) = (-ce + bf, cd - af, -bd + ae)$$

Click (or choose **Expand** from the **Manipulate** menu) to derive the symbolic expansion.

To enter $x \times r$, type **x*#r**.

Partial Derivative

$$\frac{\partial}{\partial x}$$

The Partial Derivative op is an abstract D-Linear Operator that takes a single variable and is multiplied onto the expression that it acts upon.

The partial derivative of a variable with respect to itself is one. If two variables are declared independent, the partial derivative of one with respect to the other is zero. (See the Independence Declaration in the Propositions section.)

To calculate an expression containing a partial derivative, select it and click 🖩 (or choose **Calculate** from the **Manipulate** menu). An increment value of 0.0001 is used unless you specify another. To specify an alternate increment value, create a working statement such as:

$$\boxed{\bullet}\; dx = 0.01$$

If you click ✨ (or choose **Simplify** from the **Manipulate** menu), derivatives are processed symbolically.

To take the full derivative of an expression, multiply the predefined name *d* onto the expression. To do this, select an expression and type an opening parenthesis. This encloses the expression. Then click on the left most parenthesis and type **d**.

To enter $\frac{\partial}{\partial x}$, type **Derivative(x)** or click $\frac{\partial}{\partial a}$ and type **x**.

Integral

$$\int_0^{-\pi} x^2\,dx$$

The Integral op takes one, two, or three arguments, depending on whether it has no bounds, or one or two bounds. Bounds can be infinite.

Most integral expressions contain some differential expression. However, if the integrand is zero, no differential is required.

To perform a symbolic integration, select the expression and choose either **Simplify** or **Expand** from the **Manipulate** menu.

To perform a numeric integration, select the expression and click 🖩 (or choose **Calculate** from the **Manipulate** menu). An integral must have upper and lower bounds to calculate its value.

To enter an integral with no limits, such as:

$$\int xdx$$

...type **$x*d*x**.

To enter an integral with limits, such as:

$$\int_1^2 x dx$$

...type **Integral(x*d*x):(1):(2)**.

To enter an integral with only one limit, create an integral with two limits and delete one of the limits.

Summation

$$\sum_{k=1}^{100} a_k x^k$$

The Summation op takes as its three arguments a variable, and two integers for lower and upper bounds. Summations are D-Linear Operators. They are multiplied onto the expression that they act on.

The Summation op iteratively adds together instances of the expression it is multiplied onto as the variable increments from its lower to upper bound.

To numerically evaluate a summation, click 🖩 (or choose **Calculate** from the **Manipulate** menu). To symbolically expand a summation, click 🔍 (or choose **Expand** from the **Manipulate** menu).

To enter:

$$\sum_{k=1}^{100} a_k x^k$$

...type **Summation(k):1:100)a_k*x^k**. It may be easier to enter summations from the palette.

Pi Product

$$\prod_{k=1}^{100} \left(x - a_k\right)$$

The Pi Product op takes four arguments: an iteration variable, two integers (for lower and upper bounds), and an expression as a main argument. The main argument should include at least one instance of the iteration variable. The Pi Product op iteratively multiplies together instances of the main argument, as the variable increments from its lower to upper bound.

The Pi Product op, unlike the Summation op, encloses its main argument rather than being multiplied onto it.

To evaluate a Pi Product, click 🖩 (or choose **Calculate** from the **Manipulate** menu). To symbolically expand a Pi Product, click 🔍 (or choose **Expand** from the **Manipulate** menu).

To enter:

$$\prod_{k=1}^{100} x - a_k$$

...type **#(x-a_k):k:1:100**. It may be easier to enter Pi Products from the palette.

Conditional

$$\begin{cases} 0 \; (x < 0) \\ x^2 \; (x \geq 0) \end{cases}$$

The Conditional op takes an even number of arguments (four or more) arranged in a two-column matrix. It functions as a case statement in Pascal or C. The first column of the matrix contains possible return values and the second column contains conditional statements that determine which value is returned (if any). Use the Conditional op for constructing piece-wise continuous functions.

If the first conditional statement evaluates to True (non-zero), the operator returns the expression in the first column. If the first conditional statement evaluates to False (zero), the operator evaluates the second conditional statement, and so on, for as many rows as there are in the matrix.

If none of the conditional statements evaluate to True, the undefined value, a question mark, is returned.

To evaluate, click 🖩 or ✎ (or choose **Calculate** or **Simplify** from the **Manipulate** menu.

To create a Conditional op click {a▨ b▨ or type **Conditional(** and then insert a two-column matrix. Type semicolon **;** to add a row to the matrix and type comma **,** to add a column.

Evaluate At

$$\left[\frac{1}{3}x^3\right]_{x=1}^{x=-1}$$

The Evaluate At op has two forms. The most common form contains two limits that evaluate an enclosed expression, using two substitutions, and subtracts the two results. It is used to evaluate definite integrals.

If you use only one limit, that value is substituted into the expression.

To evaluate, click 🖩 or ✎ (or choose **Calculate** or **Simplify** from the **Manipulate** menu.

To enter:

$$\left[\frac{1}{3}x^3\right]_{x=1}^{x=-1}$$

type **EvaluateAt((1/3)*x^3):x=1:x=-1**.

Range
– 1...1

The Range op represents a range of values, from minimum to maximum. Both limits must be integers or real numbers or names that evaluate to real numbers. The range op is only used in graph theories.

To enter –1...1, type **-1...1**. Enter at least two periods; extra periods are ignored.

WILDCARD VARIABLES

$\mathbb{a}$ to $\mathbb{z}$

Wildcard variables are special symbols that can represent any expression. Each of the twenty-six letters of the alphabet can be used as a wildcard variable. A range of letters is available to match different types of expressions.

Wildcard Variables	Match
$\mathbb{a}$ - $\mathbb{h}$	Any constant expression, except matrices
$\mathbb{i}$ - $\mathbb{n}$	Any non-negative literal integer
$\mathbb{o}$ - $\mathbb{q}$	Any expression, except functions
$\mathbb{r}$ - $\mathbb{z}$	Any constant or variable expression

To enter $\mathbb{x}$, type **?x** or use the variable palette's pop-up menu of wildcard variables.

There are three instances in which wildcard variables are used:

- The Substitute manipulation
- The Transform manipulation (and other transformation rules)
- Working statements of functions

With normal equations, you can Substitute expressions in one variable into different expressions with the same variable. For example, you can Substitute the equation $\sin(2x) = \cos(y) + \cos(z)$ into any equation that contains the target expression $\sin(2x)$. However, $\sin(2x)$ does not match the target expression $\sin(2\theta)$.

You can use wildcard variables to match target expressions of a given "pattern" or form regardless of the particular variables. If you create the trigonometry identity $\sin(2\mathbb{p}) = 2\sin(\mathbb{p})\cos(\mathbb{p})$, you can Substitute this relationship into any equation which contains a target expression of the form sin(2*anything*). For example, $\sin(2(A + B))$ expands to $2\sin(A + B)\cos(A + B)$.

A pattern can contain more than one wildcard variable. If they are the same wildcard, they match only identical expressions. For example, the pattern $2\sin(\mathbb{x})\cos(\mathbb{x})$ does not match the expression $2\sin(\theta)\cos(r)$, but does match $2\sin(\theta)\cos(\theta)$.

Each wildcard variable matches only a particular set of expressions. The expression $\sin(\mathbb{a}\mathbb{x})$ matches $\sin(5x)$ but not $\sin(xy)$. Wildcard variables 𝕒 through 𝕙 only match constants, not variables. Nor does sin(𝕒𝕩) match sin(x) or sin($-x$). Neither of these expressions contain a leading constant. In fact, sin(𝕒𝕩) does not even match sin($-5x$) because the negation op encloses the entire expression $5x$.

Wildcard variables can be very particular or very indiscriminate. The expression sin(𝕡) matches the sine of any expression, positive or negative, simple or complex (except for function names not used as functions).

The wildcard variable 𝕩 has no relationship to the variable x. Do not use wildcard variables in name declarations.

For information on using wildcard variables with the Transform manipulation, see the Manipulations section.

For information on using wildcard variables with function statements, see the Propositions section.

Collapsars

Collapsars are used to abbreviate long, unwieldy expressions. Symbolic algebra programs often produce huge, complicated equations that can stretch to several meters. In Theorist, you can compress any expression—large or small—to a "collapsar," which appears as an ellipsis (...).

To collapse an expression, select it, then choose **Collapse** from the **Notebook** menu. To expand a collapsar, select it, then choose **Expose** from the **Notebook** menu. Choose **Expose All** to expose all collapsars in the current theory.

Addition and multiplication expressions, when collapsed, behave a bit differently than other expressions. A collapsed string of terms (or factors) is usually displayed as two ellipses surrounding one term (or factor):

$$......-243y^{5}+......$$

If you select just an ellipsis, and choose **Expose** from the **Notebook** menu, you can "page" through the expression term by term. If the last term is displayed (on the left or right) only one ellipsis is shown:

$$32x^{5}+......$$

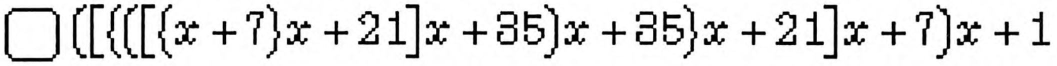
To expand the entire expression, select both ellipses and the displayed term. If you scroll to the first (or last) term, the preceding (or trailing) ellipsis is not shown.

If you choose **Auto Collapsar** from the **Preferences** submenu of the **Manipulate** menu, all new expressions wider than five and a half inches (14 cm) are displayed as collapsars.

To print out a large expression (without printing multiple pages sideways and without printing a collapsar that shows little or none of the underlying expression), Copy and Paste the expression into a comment to see its complete textual representation.

PARENTHESES

Parentheses, brackets, and curly braces are used interchangeably in expressions to mark conceptual groups. For example:



$$\square\ ([\{([\{x+7\}x+21]x+35)x+35\}x+21]x+7)x+1$$

Theorist does not distinguish between the three sets of grouping characters (( ), [ ], and { }). If you type **[x+1}**, (x + 1) is displayed. Unnecessary parentheses are usually deleted from expressions. If you want a specific set of parentheses for final output, export the equation into Expressionist and edit it there. See Appendix B, Using Expressionist with Theorist.

To group an expression in parentheses, select it and type an opening parenthesis.

QUESTION MARK

The question mark is used to signify a "blank space" when you are entering an equation. If left in an expression during a manipulation, it evaluates to an undefined value.

Each undefined value is independently undefined. For example, ? – ? does not evaluate to zero, but to ?. Zero times an undefined value is also undefined, rather than zero. Any arithmetic involving an undefined value results in the undefined value.

If a calculation or manipulation returns a question mark, some item in the manipulation is undefined or is dependent on an undefined value. For example, the Taylor series of sin(x)/x returns a question mark because the function has a removable singularity at x = 0. To get the Taylor series of sin(x)/x, select sin(x), take the Taylor series of that, and then divide out the x by clicking 🔍 (or choosing **Expand** from the **Manipulate** menu).

Manipulations

When you work with Theorist, you enter equations, manipulate equations, and graph equations. This section discusses the various ways that you can manipulate your equations.

Some manipulations are very simple and are usually performed by hand (by dragging, as opposed to using a menu item or palette button). To perform more complicated manipulations, such as factoring a polynomial, you select an expression and then choose an operation from the **Manipulate** menu or click a button from the functions palette.

This section presents:

- An overview of manipulation options
  - Auto Simplify
  - Auto Casing
  - Arbitrary Constants
  - Manipulating in place

- Hand manipulations
  - Isolate
  - Move Over
  - Commute
  - Substitute

- Command manipulations
  - Calculate
  - UnCalculate
  - Simplify
  - Expand (and MiniExpand)
  - Collect
  - Factor
  - Apply
  - Transform
  - Taylor Series
  - Integrate by Parts

Manipulation Options

When you manipulate an expression, you set off a chain reaction that results in one or more new expressions logically derived from the original. Depending on different preferences that you set, new expressions can appear in different forms. In particular, you should be aware of the following ways you can affect the creation of new expressions:

- Auto Simplify
- Auto Casing
- Arbitrary Constants
- Manipulating in place

Auto Simplify

Simplify is a powerful, multifaceted manipulation that reduces the complexity of an expression. If Auto Simplify is on, all other manipulations (except Commute and Apply) simplify an expression before displaying it. Auto Simplify executes the Simplify manipulation repeatedly (up to ten times) until the expression stops changing.

If you want to follow every step of a manipulation, turn Auto Simplify off. There are a few other instances when you will want Auto Simplify off. For example, the Collect manipulation "spreads out" the factors of a product, but Simplify groups them back together again. To rearrange the factors, turn Auto Simplify off and use Collect:

$$\frac{a^2b^3}{x^4y^5}=a^2b^3x^{-4}y^{-5}$$

Then use the Commute and Simplify manipulations to organize the factors differently:

$$\frac{a^2b^3}{x^4y^5}=a^2\,\frac{b^3}{y^5}\,\frac{1}{x^4}$$

Collect also factors out negative one (–1) from expressions, whereas Auto Simplify multiplies it back in.

All examples in this section assume that Auto Simplify is on, unless stated otherwise.

To turn Auto Simplify on or off, choose **Manipulate▸Preferences▸ Auto Simplify**.

Auto Casing

Two manipulations, Isolate and Move Over, can produce more than one possible result. For example, there are two different solutions to $x^2 = 4$: $x = 2$ and $x = -2$. If Auto Casing is on, both cases are displayed in separate case theories. If not, only the primary case is displayed. If Auto Casing is on, up to sixteen separate case theories can be generated by a single manipulation. All examples in this section assume that Auto Casing is on, unless stated otherwise. (See Case Theories in the Propositions section.)

Also, if you integrate an expression (with Simplify, Expand, or Integrate by Parts), and Auto Casing is on, the manipulation can generate one or more arbitrary integer constants.

To turn Auto Casing on or off, choose **Manipulate▸Preferences▸Auto Casing**.

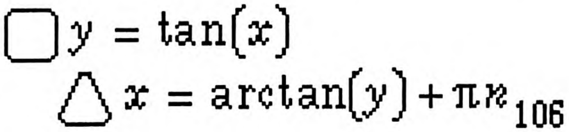
Arbitrary Constants

Some manipulations can generate one or more arbitrary constants. Real numbers are represented by the letter *c* and a numeric subscript. Integers are represented by the letter *n* and a numeric subscript. These subscripts start at 100 and are incremented by one for each new arbitrary constant. To change these defaults, choose **Manipulate▸Preferences▸Arbitrary Constants...**.

If Auto Casing is on, Isolate, Move Over, Simplify, and Expand can all produce arbitrary constants in the following situations.

Symbolically integrating an expression with Simplify or Expand can produce a real arbitrary constant. Using the Integrate by Parts manipulation can also produce a real arbitrary constant.

Isolating the argument of transcendental (e.g., trigonometric, exponential, and hyperbolic) functions produces arbitrary constants if Auto Casing is on. For example, if you use Isolate to find a solution in *x* for the equation $y = \tan(x)$, you create an arbitrary constant:



□ $y = \tan(x)$

△ $x = \arctan(y) + \pi n_{106}$

The constant n_{106} indicates that any *integer* creates a valid statement. That is, the following are possible solutions for *x:*

$$x = \arctan(y) + 5\pi$$

$$x = \arctan(y) + -3\pi$$

But not:

$$x = \arctan(y) + \frac{3}{2}\pi$$

Manipulating in Place

If you hold down the ⌘ and option keys on a Macintosh computer or alt and Ctrl keys on a Windows computer when you execute a manipulation, the new expression (the conclusion) replaces the original. Otherwise, the conclusion appears on a new line below the original expression.

HAND (MOUSE) MANIPULATIONS

The following manipulations are most easily (and most often) performed by hand:

- Isolate
- Move Over
- Commute
- Substitute

Simple manipulations, such as rearranging the terms of an expression, or moving a term of an equation from one side of the equals sign to the other, are easily performed by hand.

Three manipulations, Isolate, Move Over, and Commute, are very similar. In fact, the difference between executing one or another of these manipulations often depends on the exact spot where you release the mouse; a small distance to the left or right is all that distinguishes an Isolate manipulation from a Move Over manipulation.

To invoke any of these manipulations, select an expression (double-click on the expression or click-and-drag over the expression), then hold down the ⌘ key (Macintosh) or Ctrl key (Windows). The mouse pointer arrow turns into a pointing hand. Using this hand, click on the selected expression and drag it to a new location. Theorist highlights the different places where you can release the selection to perform a manipulation. Releasing the mouse performs the manipulation; which manipulation, if any, is determined by *where* you release the mouse.

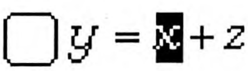
The Commute manipulation highlights possible insertion points with a small rectangle. The Move Over manipulation highlights one whole side of an equation and displays a thin line between the selection rectangle and the other side of the equation. The Isolate manipulation displays a large, highlighted rectangle beyond and outside the equation with a thin line connected to the selection rectangle.

Isolate

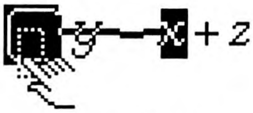
Use Isolate as a simple "solve for" manipulation. Select an expression inside an equation, and drag it with the pointing hand.

To Isolate an expression to the left, drag it over the proposition icon. To Isolate an expression to the right, drag it past the right-hand side of the equation. If you do not drag it far enough, you might execute a Commute or Move Over manipulation rather than an Isolate manipulation.

When you are in a position to make an Isolate manipulation, a thin bar appears connecting the selected expression and the release point.



Selected x



Isolating x

Once you have selected an expression, you can also execute the Isolate manipulation by choosing **Isolate** from the **Manipulate** menu or clicking on the functions palette. The menu item and palette button manipulations always isolate an expression to the left.

Selected Expression	After Isolate Manipulation	Comment
□ $y = 5\boxed{x}+7$	△ $x = \frac{1}{5}[y-7]$	
□ $y = \boxed{5x}+7$	△ $5x = y-7$	
□ $y = \boxed{5x+7}$	△ $5x+7 = y$	
□ $y = \boxed{5}x+7$	△ $5 = \frac{y-7}{x}$	Auto Casing off

Isolate does not always "solve" an equation if there is more than one occurrence of the given expression in your equation.

The Isolate manipulation can solve quadratic polynomials if you select *both* occurrences of the variable or expression to Isolate, and drag them left or right.

When the Auto Casing option is on, Isolate generates multiple case theories (with additional equations, if necessary) and equations with arbitrary constants, if necessary.

Selected Expression	After Isolate Manipulation	Comment
□ $x^2-3\boxed{x} = -2$	△ $x = \frac{1}{3}\left(x^2+2\right)$	Only one *x* isolated
□ $\boxed{x}^2-3\boxed{x} = -2$	○ △ $x = 2$ ○ △ $x = 1$	Two case theories generated
□ $\boxed{a}b = c$	○ □ $b \neq 0$ △ $a = \frac{c}{b}$ ○ □ $b = 0$ △ $0 = c$	Two case theories generated

□ $\tan[x] = y$	△ $x = \arctan[y] + \pi n_{107}$	Generates arbitrary constant
□ $\begin{bmatrix} a & b \\ c & d \\ e & f \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$	△ $d = 4$	Isolating matrix element
□ $y = x^4$	○ △ $x = y^{\frac{1}{4}}$ ○ △ $x = iy^{\frac{1}{4}}$ ○ △ $x = -y^{\frac{1}{4}}$ ○ △ $x = -iy^{\frac{1}{4}}$	
□ $y = x^5$	△ $x = e^{\frac{2}{5}\pi n_{110} i} y^{\frac{1}{5}}$	Exponents greater than four produce arbitrary constants

If Auto Casing is off in these situations, you only get the first answer, with no case theories and no arbitrary constants.

The Isolate manipulation does not work with multiple selections, except for solving quadratic equations as shown.

Move Over

Use the Move Over manipulation to relocate an expression from one side of an equation to the other. As with Isolate, select an expression, then press and hold the ⌘ key (Macintosh) or Ctrl key (Windows). With the pointing hand, drag the expression to the other side of the equation. Do not drag as far as you would to execute an Isolate manipulation. Isolate and Move Over highlight in the same way: a bar is displayed between the selection and possible release points.

Once you have selected an expression, you can also execute a Move Over manipulation by choosing **Manipulate▸Other▸Move Over** or by choosing 🔧 from the 🔨 pop-up subpalette.

Selected Expression	After Move Over Manipulation	Comment
□ $5x + 3 = y$	△ $5x = y - 3$	
□ $5x + 3 = y$	△ $x = \frac{1}{5}[y - 3]$	
□ $5x = y - 3$	△ $0 = -5x + y - 3$	

□ $5x = y - 3$ △ $5 = \frac{y-3}{x}$ Auto Casing off

□ $4x - 2 = 3x - [2 - x]$ △ $0 = 0$

The Move Over manipulation does not work with multiple selections.

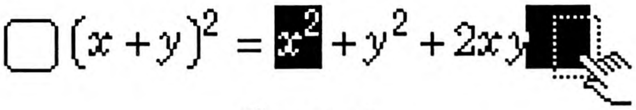
Commute

Use the Commute manipulation to rearrange terms in a sum or factórs in a product. Select one or more expressions, press and hold the ⌘ key (Macintosh) or Ctrl key (Windows), and drag the expressions to the left or the right. If you drag an expression across the equals sign in an equation (or outside an equation), the manipulation is no longer a Commute, but a Move Over or Isolate manipulation.

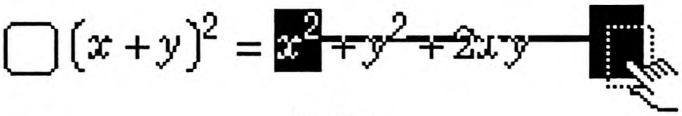
When you click and drag terms of an expression with the pointing hand, all possible release points highlight. This is particularly useful when you are working with algebra and non-commutative operators (e.g., derivatives, summations). Only valid Commute operations highlight a release point. For example, if you try to move the x^2 in the following expression, only one other position is valid (to the left of π), whereas the 4 can move to three valid locations (each of which is highlighted in turn).

$$\square\ 4\frac{\partial}{\partial x}\pi x^2$$

Isolate and Commute highlight differently. Isolate displays a bar between the two highlighted rectangles, and Isolate is farther away from the equation:.



Commute



Isolate

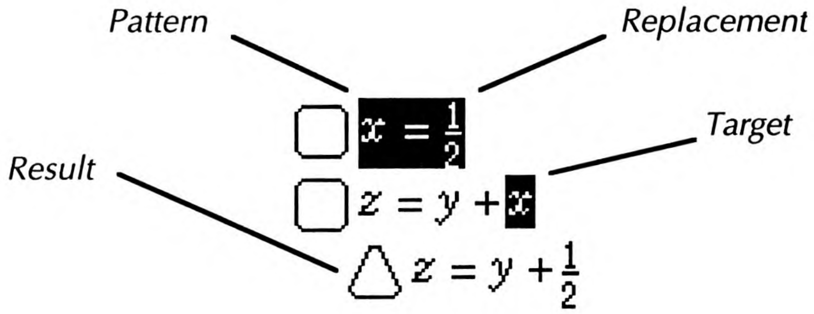
To reverse the order of a series of terms or factors, select the series, and choose **Manipulate▸Other▸Commute** or choose ⇄ from the 🔨 pop-up subpalette. If you select only a single term (e.g., 5, x, $\sqrt{x+y}$), the menu manipulation has no effect. Commute as a hand manipulation allows you to rearrange individual terms and factors. When you Commute an expression using the menu option, the order of the terms is reversed.

Selected Expression	After Commute Manipulation	Comment
□ $3x^2$	△ $3x^2 = x^2 \cdot 3$	By hand only
□ $3x^2+x^2-2x+5$	△ $\cdots = 5-2x+x^2+3x^2$	By palette or menu choice
□ $3x^2+x^2-2x+5$	△ $\cdots = 3x^2+\left(-2x+x^2\right)+5$	By palette or menu choice
□ $5x\frac{\partial}{\partial x}\sin[x]$	□ $5x\frac{\partial}{\partial x}\sin[x]$	Partial derivatives do not commute with their variables
□ $3x\frac{\partial}{\partial x}\sin[x]$	△ $\cdots = x\frac{\partial}{\partial x}\sin[x]\cdot 5$	Constants do commute in differential expressions

The Commute manipulation works with multiple selections, but not by hand—only if you use the menu command or palette.

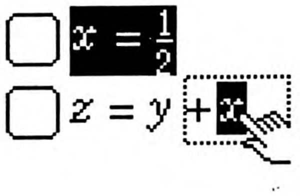
Substitute

The Substitute manipulation is a powerful technique for replacing expressions of a particular form with an equivalent expression of a different form. This manipulation utilizes a substitution equation and a target expression. The substitution equation consists of a pattern expression and a replacement expression.



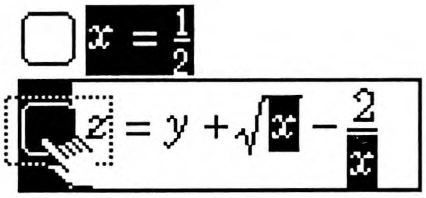
You can invoke the Substitute manipulation in three ways:

❶ Select a substitution equation and drag it over a target expression to replace a single expression.



If you drag a substitution equation over an equation with an expression that matches the replacement expression (rather than the pattern expression), Substitute works in reverse. That is, the replacement expression is used as the pattern expression and vice versa. This reversal only occurs with this form of the Substitute manipulation.

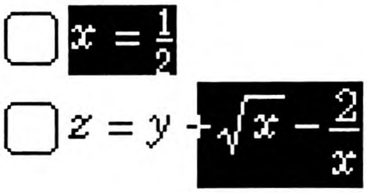
❷ Select a substitution equation and drag it over the icon of another equation to replace all occurrences of the target expression in that equation.



$x = \frac{1}{2}$

$z = y + \sqrt{x} - \frac{2}{x}$

❸ Select one or more substitution equations and one or more (non-equation) target expressions. Then choose **Manipulate▸Other▸Substitute** or choose ↗ from the 🔨 pop-up subpalette to replace all occurrences of each target expression.



$x = \frac{1}{2}$

$z = y + \sqrt{x} - \frac{2}{x}$

Non-equations are always interpreted as targets, while equations are always interpreted as substitutions. To make an equation a target, select both sides of the equation individually, or select the proposition icon.

Note: You do not need to select target expressions with total precision. Selecting the square root of x will work as well as selecting just x.

To select a substitution equation, click on its equals sign. Use the pointing hand (press and hold the ⌘ Macintosh key or Ctrl Windows key) to drag the substitution equation into place. Target expressions highlight when they will accept a substitution. Release the mouse to execute the substitution.

If you use wildcard variables (e.g., 𝓍, 𝓎) in a substitution equation, they match any expression of the type associated with that wildcard variable. Substitution expressions with regular variables only match those explicit variables. To enter wildcard variables, type a question mark, then the variable letter (e.g., **?x**). For more information, see Wildcard Variables in the Expressions section.

The targets and substitution equations must be in the same theory, or in nested theories. The result is displayed in the most deeply nested theory.

If you find you are performing certain substitution manipulations frequently, you may want to create Transform manipulations to handle these replacements. The Transform manipulation is described later in this section.

COMMAND MANIPULATIONS

Command manipulations are invoked by choosing an option from the **Manipulate** menu or clicking a button on the functions palette. Other command manipulations are invoked by choosing an option from the **Manipulate** menu's **Other** submenu or choosing a button from the functions palette's 🔨 pop-up subpalette. Like hand manipulations, the result is displayed below the selected expressions, unless the manipulation is executed in place. To replace the selected expression (with the new expression), hold down the ⌘ and option Macintosh keys or alt and Ctrl Windows keys while executing the manipulation.

This section discusses the following manipulations:

- Calculate
- UnCalculate
- Simplify
- Expand (and MiniExpand)
- Collect
- Factor
- Apply
- Transform
- Taylor Series
- Integrate by Parts

Calculate generates a numeric evaluation of an expression, whereas UnCalculate returns a numeric expression to its symbolic form. Simplify and Expand reduce expressions to simpler, more fundamental forms, whereas Collect and Factor tend to create more compact expressions. With Apply you can change an expression in many ways while maintaining its logical integrity. Taylor Series approximates a function. The Integrate by Parts manipulation is a useful technique for integration.

All command manipulations, with the exception of Apply, can be used on multiple selections.

All examples in this section assume that Auto Simplify is on, unless stated otherwise.

Calculate

Calculate tries to evaluate an expression to a number, like a pocket calculator. It may also produce a matrix or vector of numbers. Calculate uses working definitions of names (if necessary). If an expression can not be fully calculated, because of undefined names or for other reasons, Calculate evaluates all possible subexpressions.

The results of a Calculate manipulation are usually accurate to the full fifteen digits of precision, if possible. You can control the number of digits displayed by choosing the desired accuracy from the **Notebook** menu's **Display Precision** sub-menu. The full fifteen digits are maintained internally.

Selected Expression	After Calculate Manipulation	Comment
$3-\frac{9}{2}$	–1.5	
4π	12.566	Reverse with UnCalculate
sin(1)	0.84147	
$\frac{1}{8}x+7$	$0.125x$ +7	If x is undefined
$\frac{1}{8}x+7$	7.375	If $x = 3$
y	7.375	If $x = 3$ and $y=\frac{1}{8}x+7$

To calculate the partial derivative of an expression with respect to a particular variable, that variable must have an assigned value. The increment value used for the independent variable is determined by a working statement of the form $dx = 0.001$. In the absence of such a definition, Theorist uses 1/10,000*th* of the difference between a defined minimum and maximum (e.g., $x > 0$ and $x \le 100$). If neither is available, an increment value of 0.0001 is used, regardless of the value of the variable.

Definite integrals (i.e., integrals with both upper and lower limits) are calculated numerically. The limits can be any real values including infinity and negative infinity. An integrand must contain a differential to indicate the variable of integration. Integral calculation uses an internally defined increment value. The variable of integration must be a real number.

Selected Expression	After Calculate Manipulation	Comment
$\frac{\partial}{\partial x}y$	$\frac{\partial}{\partial x}y$	If x and y are undefined
$\frac{\partial}{\partial x}y$	0.125	If $x = 3$ and $y=\frac{x}{8}+7$
$\frac{\partial}{\partial x}\sin(1)$	0	

$$\int_0^1 x^3 dx \qquad 0.25$$

$$\int_{-\infty}^{\infty} \frac{dx}{1+x^2} \qquad 3.1416$$

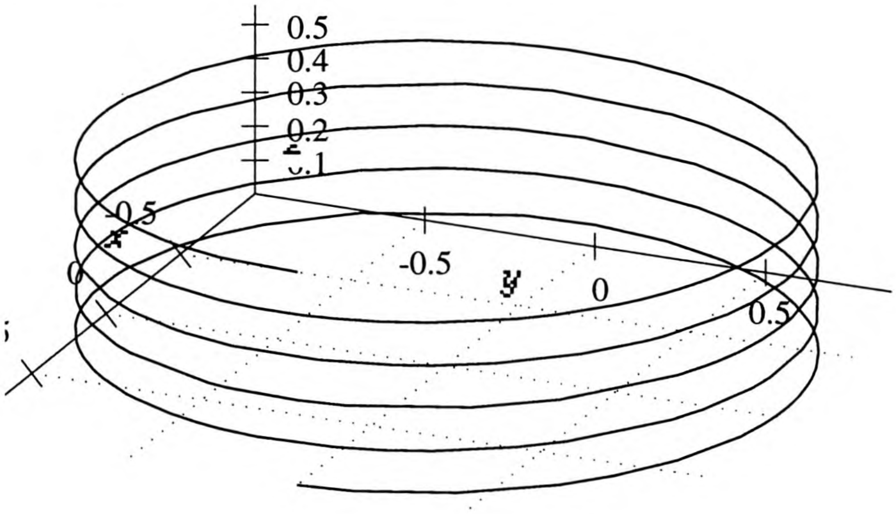
If you want to integrate over a path in space, or over the complex plane, parametrize the path over some real variable and calculate the integral over that variable. For example, consider the spiral curve described by the following working statements:

● $x = \cos(5t)$

● $y = \sin(5t)$

● $z = \frac{t}{10}$

Using these statements you can create a Line plot in a three dimensional graph theory:



Using the Pythagorean theorem, you can calculate the length, *s*, of the curve:

□ $(ds)^2 = (dx)^2 + (dy)^2 + (dz)^2$

With the help of some of the rules in the New Notebook, you can calculate the integral for this path in space:

△ $\int_0^{2\pi} ds = 31.422$

For more information about creating graphs of this type, see the Graph Proposition Elements section.

Before graphing a polynomial with algebraic coefficients (e.g., $x^3 + 3\sqrt{2}\,x^2 + 6x + 2\sqrt{2}$), use Calculate to evaluate the terms. This improves graphing speed.

To calculate an expression, select it, and choose **Calculate** from the **Manipulate** menu or click 🖩 on the functions palette.

UnCalculate

The UnCalculate manipulation attempts to undo the effects of the Calculate manipulation. If Calculate produces an integer such as 6 or 42, UnCalculate has no effect. However, UnCalculate can recreate simple fractions and constants from decimals. UnCalculate reduces to zero "very small" factors and terms (10^{12} times smaller than other values in an expression).

UnCalculate also returns square and cube roots of rational numbers, products of π or π^2 and rational numbers, and e raised to rational powers.

Selected Expression	After UnCalculate Manipulation
0.333333333333333	$\frac{1}{3}$
12.5663706143592	4π
20.0855369231877	e^3

Note: For UnCalculate to regenerate an original value, selected values must be exact to approximately ten to thirteen digits of precision (regardless of the selected display precision), depending on the CPU and FPU.

To UnCalculate an expression, select it, and choose **UnCalculate** from the **Manipulate** menu's **Other** submenu or choose 🖩 from the functions palette's ⚒ pop-up subpalette.

Simplify

Simplify executes a wide range of operations designed to reduce the complexity of an expression. Simplify combines constants, terms, and factors using rational arithmetic, cancels where possible, and arranges expressions into canonical form.

Products are sorted in class order where possible (Constants, Variables, M-Linear Operators, D-Linear Operators), with constants to the left. Sums are sorted in class order, with constants to the right. Imaginary components are sorted so they appear after real components. Powers and subscripts are sorted in numerical order, to arrange polynomials by descending powers. In the absence of other criteria, Simplify sorts in alphabetical order.

Constants are reduced to simple forms, if possible. Powers of powers are distributed out. Derivatives are evaluated if possible, as are simple integrals of linear functions, and linear functions of simple integrals.

Selected Expression	After Simplify Manipulation	Comment
$\sqrt{x}\,\sqrt{x} - \frac{x^2}{x} + \frac{6}{2\cdot 3+0} - 1$	0	
$\frac{1}{\sqrt{2}} - (x-1)$	$-x + \frac{1}{2}\sqrt{2} + 1$	
$5 + x^2 + x^3 + x\cdot 2 + \frac{x^{10}}{2}$	$\frac{1}{2}x^{10} + x^3 + x^2 + 2x + 5$	
$x^3y + 2x^2y^2 + xy^3$	$xy^3 + 2x^2y^2 + x^3y$	
$e^{2\pi i} + \log(1000)$	4	
$\left(x^2\right)^n$	x^{2n}	Reverse with Collect (if Auto Simplify is off)
$\int \left(x^4 + \sin[x]\right)dx$	$-\cos(x) + \frac{1}{5}x^5 + c_{106}$	Generates an arbitrary constant (c_{106})
$\begin{pmatrix} A & B \\ C & D \end{pmatrix}_{(1,2)}$	B	
$\frac{(-x-1)(x-2)}{x+1}$	$-x+2$	

The Simplify manipulation also:

- Propagates negations through powers

  $(-x)^5 \quad \rightarrow \quad -x^5$

  $\sqrt{-x} \quad \rightarrow \quad i\sqrt{x}$

- Removes negations within absolute values

  $|-x| \quad \rightarrow \quad |x|$

- Adjusts function notation to canonical form

  $\sin^2 x \quad \rightarrow \quad (\sin[x])^2$

  $\sin^{-1} x \quad \rightarrow \quad \arcsin(x)$

- Invokes working order statements

  $x^5 \quad \rightarrow \quad 0$ (if $x^5 = 0$ is a working statement)
- Compresses functions of anti-functions

  $\sin(\arcsin(x)) \quad \rightarrow \quad x$
- Interprets element of identity matrix

  $5_{(3,3)} \quad \rightarrow \quad 5$

  $5_{(3,2)} \quad \rightarrow \quad 0$
- Distributes the Adjoint op

  $(x \sin[y])^{\dagger} \quad \rightarrow \quad x^{\dagger} \sin(y^{\dagger})$
- Moves fractional powers of integers out of denominators

  $\frac{1}{\sqrt{3}} \quad \rightarrow \quad \frac{1}{3}\sqrt{3}$
- Evaluates the Evaluate At op
- Executes conditional ops if the conditional expression can be calculated

If the Auto Simplify option is on (choose **Manipulate▸Preferences▸Auto Simplify**), a Simplify manipulation is executed after every manipulation (except Commute and Apply). This is the default. Turn off Auto Simplify if you want to proceed through a manipulation step-by-step, or if you want to rearrange the parts of an un-simplified expression.

Auto Simplify may cause problems when working with integrals. If Auto Simplify is on, the integrand is simplified before figuring the integration. In most cases, this works fine. Occasionally, however, the simplified expression is more difficult to integrate. For example, consider the following integral:

$$\int d \sin(\sin[x])$$

If Auto Simplify is on, Simplify produces the following solution:

$$\int \cos(x)\cos(\sin[x])dx$$

If Auto Simplify is off, you get:

$$\sin(\sin[x])$$

Simplify also applies all transformation rules listed in the current theory as "Upon Simplify." (See the Transform manipulation later in this section.)

To simplify an expression, select it and choose **Simplify** from the **Manipulate** menu or click on the functions palette. You can also select an expression, hold down the ⌘ Macintosh key or Ctrl Windows key, and double-click on the expression. To simplify an expression in place, select it, hold down the ⌘ and option Macintosh keys or alt and Ctrl Windows keys, and double-click on the expression.

Expand and MiniExpand

Expand executes a range of operations designed to increase the complexity of an expression. In many cases it reverses the Collect and Factor manipulations. During its operation, Expand may execute one or more Simplify manipulations.

MiniExpand only works on the outside layer of an expression. Expand is recursive in that it applies a MiniExpand manipulation to all expressions nested within the selected expression. However, Expand is not repetitive. Repeated Expand manipulations may further expand an expression.

$$\text{Expand:} \quad \left([x+1]^2+1\right)^2 \quad \to \quad x^4+4x^3+8x^2+8x+4$$

$$\text{MiniExpand:} \quad \left([x+1]^2+1\right)^2 \quad \to \quad (x+1)^4+2(x+1)^2+1$$

Expand symbolically distributes products, and multiplies out powers of sums, matrices, and derivative and summation operators. It expands matrix sums, inverses and determinants, dot products, and powers of products.

If applied to a rational function (one polynomial divided by another), Expand divides polynomials or distributes a common divisor over a sum. If applied to a fraction with an expanded polynomial in the numerator and a product of polynomials in the denominator, Expand performs a partial-fraction decomposition.

Selected Expression	After Expand Manipulation	Comment
$2x(3y+7)$	$14x+6xy$	Reverse with Collect
$x(x+1)^2(x-1)^2$	x^5-2x^3+x	Reverse with Factor
$\begin{pmatrix}3\ 4\\5\ 6\end{pmatrix}^2\begin{pmatrix}x\\y\end{pmatrix}$	$\begin{pmatrix}4[5x+6y]+3[3x+4y]\\6[5x+6y]+5[3x+4y]\end{pmatrix}$	Needs second Expand (see next)
$\begin{pmatrix}4[5x+6y]+3[3x+4y]\\6[5x+6y]+5[3x+4y]\end{pmatrix}$	$\begin{pmatrix}29x+36y\\45x+56y\end{pmatrix}$	

Selected Expression	After Expand Manipulation	Comment
$\begin{pmatrix}3&4\\5&6\end{pmatrix}+\begin{pmatrix}x&y\\z&0\end{pmatrix}+\frac{a}{2}$	$\begin{pmatrix}\frac{1}{2}a+x+3 & y+4\\ z+5 & \frac{1}{2}a+6\end{pmatrix}$	
$\frac{a}{2}\begin{pmatrix}x&y\\z&0\end{pmatrix}$	$\begin{pmatrix}\frac{1}{2}ax & \frac{1}{2}ay\\ \frac{1}{2}az & 0\end{pmatrix}$	Reverse with Collect
$\left(\frac{\partial}{\partial x}\right)^3 x^{100}$	$970200x^{97}$	
$\left(\frac{\partial}{\partial x}+x\right)^2\sin(x)$	$2x\cos(x)+x^2\sin(x)$	
$\sum_{k=6}^{10} x^k$	$x^{10}+x^9+x^8+x^7+x^6$	
$\prod_{k=1}^{4} x^k$	x^{10}	
$\begin{pmatrix}3&0\\y&7\end{pmatrix}^{-5}$	$\frac{1}{4084101}\begin{pmatrix}16807&0\\-4141y&243\end{pmatrix}$	
$\left\lvert\begin{pmatrix}x&y\\z&a\end{pmatrix}\right\rvert$	$ax-yz$	Determinant
$\frac{1}{4}\left(\frac{\partial}{\partial x},\frac{\partial}{\partial y},\frac{\partial}{\partial z}\right)\bullet\left(x^4,y^4,z^4\right)$	$x^3+y^3+z^3$	
$\left(x^2y^3z^8\right)^5$	$x^{10}y^{15}z^{40}$	Reverse with Factor
$\frac{x^4+2x^3+5x^2-x+1}{x^2+1}$	$\frac{-3x-3}{x^2+1}+x^2+2x+4$	Reverse with Collect
$\frac{x+7}{x^2-1}$	$\frac{7}{x^2-1}+\frac{x}{x^2-1}$	Reverse with Collect
$\frac{x+7}{(x+1)(x-1)}$	$-3\frac{1}{x+1}+4\frac{1}{x-1}$	Reverse with Collect

The Expand manipulation also performs the following actions:

- Distributes powers over products and quotients

$$\left(\frac{ab}{c}\right)^7 \quad\rightarrow\quad \frac{a^7b^7}{c^7}$$

- Distributes the absolute value op

$$\left|\frac{ab}{c}\right| \quad \rightarrow \quad \frac{|a||b|}{|c|}$$

- Distributes logs of products

$$\log(abc) \quad \rightarrow \quad \log(a) + \log(b) + \log(c)$$

- Distributes an integral over matrix elements

$$\int \begin{pmatrix} x & y \\ a & b \end{pmatrix} \quad \rightarrow \quad \begin{pmatrix} [\int x] & [\int y] \\ [\int a] & [\int b] \end{pmatrix}$$

Expand also applies all transformation rules listed in the current theory as "Upon Expand." (See the Transform manipulation later in this section.)

To Expand an expression, select it and choose **Expand** from the **Manipulate** menu or click 🔧 on the functions palette. To MiniExpand an expression, select it, and choose **MiniExpand** from the **Manipulate** menu's **Other** submenu or choose 🔨 from the functions palette's 🔨 pop-up subpalette.

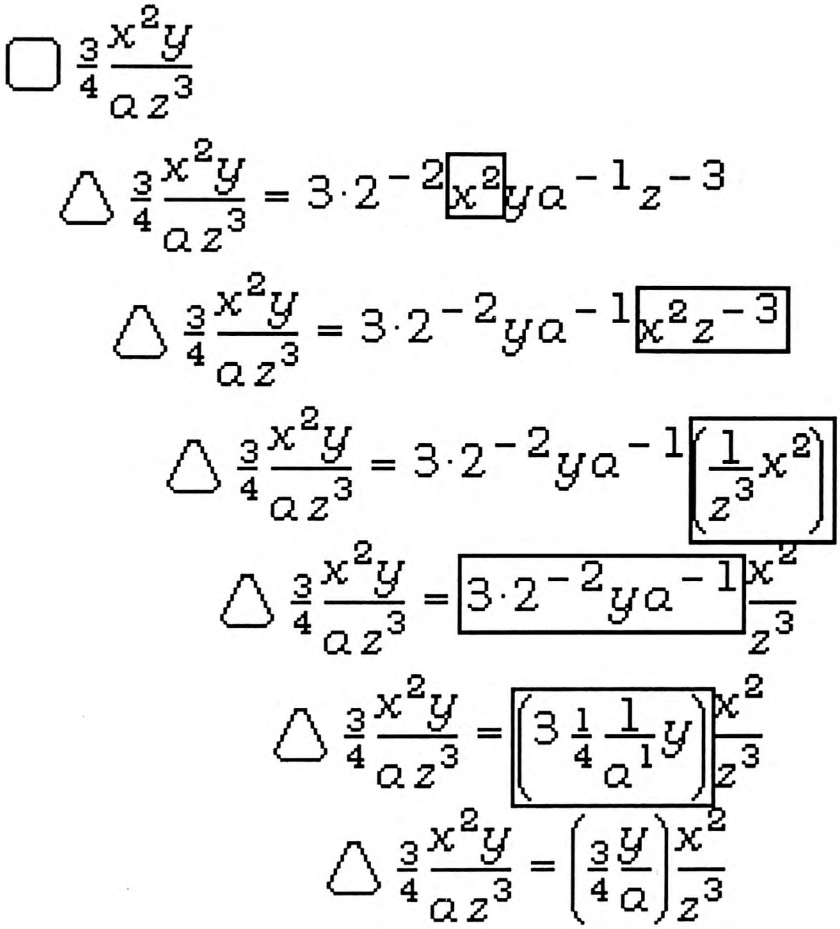
Collect

Collect separates out common terms of a sum and orders the remaining terms as a polynomial in a given variable, grouping coefficients by powers in the polynomial variable. Applied to a sum of fractions, Collect adds the fractions together and collects all terms over a common denominator. Collect also separates out common factors of a matrix.

Selected Expression	After Collect Manipulation	Comment
x^5	$x\,x\,x\,x\,x$	Auto Simplify off; Reverse with Simplify
$7x^4 + 21x^3 + 21x^2 + 7x$	$7(x^3 + 3x^2 + 3x + 1)x$	Reverse with Expand
$xy^3z + x^3y^2z^2 + x^2y^7z^3$	$(y + xy^5z^2 + x^2z)xy^2z$	Reverse with Expand
$xy^3z + x^3y^2z^2 + x^2y^7z^2$	$(y + [x^2 + xy^5]z)xy^2z$	Reverse with Expand
$\frac{1}{x} + \frac{1}{y} + \frac{1}{z}$	$\frac{xy + (x+y)z}{xyz}$	Reverse with Expand

$xy^3z+\frac{x^3y^2}{z^2}+x^2y^7z^3$	$\frac{(x^2+xy^5z^5+yz^3)xy^2}{z^2}$	Reverse with Expand
$\begin{pmatrix}12x^2 & 0\\ 9x^6 & 6x\end{pmatrix}$	$3x\begin{pmatrix}4x & 0\\ 3x^5 & 2\end{pmatrix}$	Reverse with Expand
$\frac{\partial}{\partial x}y+\frac{\partial}{\partial x}z$	$\frac{\partial}{\partial x}(1y+z)$	Reverse with Simplify

Collect "spreads out" factors in a product so you can rearrange and regroup them in a new order. (Turn Auto Simplify off, to use Collect in this way.) For example, you can use the following manipulations to rearrange a fractional product:



□ $\frac{3}{4}\frac{x^2y}{az^3}$

Collect Expression

△ $\frac{3}{4}\frac{x^2y}{az^3}=3\cdot 2^{-2}\boxed{x^2}ya^{-1}z^{-3}$

Commute x^2 by hand

△ $\frac{3}{4}\frac{x^2y}{az^3}=3\cdot 2^{-2}ya^{-1}\boxed{x^2z^{-3}}$

Simplify part of expression

△ $\frac{3}{4}\frac{x^2y}{az^3}=3\cdot 2^{-2}ya^{-1}\boxed{\left(\frac{1}{z^3}x^2\right)}$

Simplify again

△ $\frac{3}{4}\frac{x^2y}{az^3}=\boxed{3\cdot 2^{-2}ya^{-1}}\frac{x^2}{z^3}$

Simplify other part

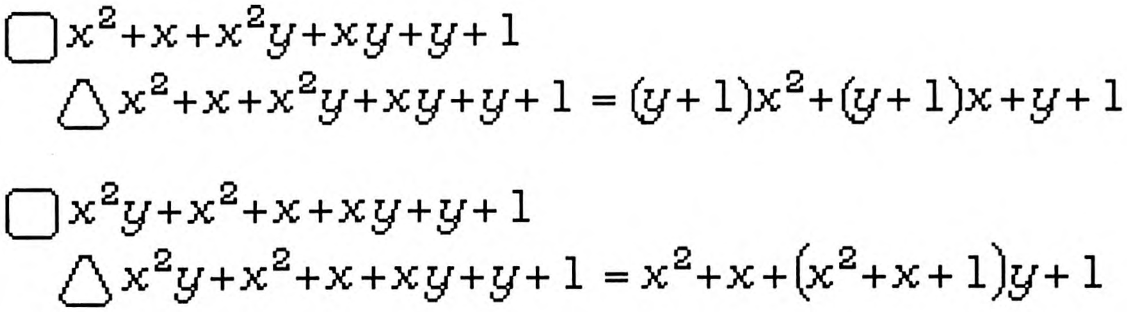
△ $\frac{3}{4}\frac{x^2y}{az^3}=\boxed{\left(3\frac{1}{4}\frac{1}{a^1}y\right)}\frac{x^2}{z^3}$

Simplify again

△ $\frac{3}{4}\frac{x^2y}{az^3}=\left(\frac{3}{4}\frac{y}{a}\right)\frac{x^2}{z^3}$

Auto Simplify must also be off in order to use Collect to distribute power sums ($x^{a+b} \rightarrow x^ax^b$) and integer powers ($x^3 \rightarrow xxx$).

The polynomial variable determines the grouping of terms when you invoke the Collect manipulation. You can specify the polynomial variable of an expression with more than one variable by arranging the terms of the polynomial. The right-most variable of the first term is used as the polynomial variable. The following two equations are equivalent; in the first, x is the polynomial variable, in the second, y is the polynomial variable:



□ $x^2+x+x^2y+xy+y+1$

△ $x^2+x+x^2y+xy+y+1 = (y+1)x^2+(y+1)x+y+1$

□ $x^2y+x^2+x+xy+y+1$

△ $x^2y+x^2+x+xy+y+1 = x^2+x+\left(x^2+x+1\right)y+1$

Collect also applies the inverse of all transformation rules listed in the current theory as "Upon Expand." (See the Transform manipulation, later in this section.)

To Collect an expression, select it and choose **Collect** from the **Manipulate** menu or click on the functions palette.

Factor

Factor executes several operations designed to dismantle an expression into its most fundamental terms. Integers are converted into their prime factors, and sums are factored as polynomials, if possible. Products with common factors are joined together. In general, Factor breaks down an expression more completely than Collect. For example, Collect separates out the common terms of a polynomial whereas Factor determines roots of a polynomial. In most cases, you can reverse the effects of Factor with Expand.

Factor can manipulate all polynomials with numeric coefficients (one variable), and some polynomials with symbolic coefficients (or polynomials with multiple variables). Quadratic polynomials are factored symbolically using the quadratic equation.

Selected Expression	After Factor Manipulation	Comment
-720	$(-1)\cdot 2^4\cdot 3^2\cdot 5$	Reverse with Expand or Simplify
$-x^6$	$(-1)x^6$	
x^2-1	$(x+1)(x-1)$	Reverse with Expand
x^2y^2	$(xy)^2$	Reverse with Expand
$x^2-11x-26$	$(x+2)(x-13)$	Reverse with Expand

If you select a cubic or quartic polynomial for factoring, a dialog asks whether you want to proceed numerically or symbolically. Theorist uses the Cubic formula and the Quartic formula to symbolically factor third-order and fourth-order polynomials. These manipulations can be extremely time consuming, especially if the intermediate expressions do not cancel. On the other hand, numerically factoring such polynomials is usually very fast.

Fifth order and higher polynomials are always factored numerically. The roots are often real or complex numbers. These numbers are usually accurate to thirteen digits.

You can specify the polynomial variable of an expression with more than one variable by arranging the terms of the polynomial. The right-most variable of the first term is used as the polynomial variable. For example, $x^2+ 2xy + y^2$ is factored in terms of x, whereas $y^2 + 2xy +x^2$ (or $2xy + y^2 +x^2$) is factored in terms of y.

To Factor an expression, select it and choose **Factor** from the **Manipulate** menu or click on the functions palette.

Apply

Apply is a generic manipulation that provides a technique for doing almost anything to both sides of an equation, or both the numerator and denominator of a fraction. Apply can also be used to multiply an expression by a term and its inverse (n and $1/n$), or add and subtract the same value from an expression ($+n$ and $-n$). Apply never changes the validity of an expression or equation, just the structure.

To use Apply, select an expression or equation and choose **Apply** from the **Manipulate** menu or click on the functions palette. There are four modes for using Apply, but in each instance, the manipulation creates a conclusion with two selected empty expressions (indicated by question marks).

Mode 1 Using Apply with an Equation

In this mode, you can add or subtract the same expression from both sides of an equation. For example, to complete the square of a polynomial, select the equation (click on the equals sign), invoke the Apply manipulation, type **+1**, then use Simplify and Factor:

$\square\ 0 = x^2 - 4x + 3$

$\triangle\ 0 + 1 = (x^2 - 4x + 3) + 1$ *Applied & added 1*

$\triangle\ 1 = x^2 - 4x + 4$ *Simplified*

$\triangle\ 1 = (x - 2)^2$ *right side Factored*

As another example of using Apply with an equation, consider calculating the integral of the following expressions. Select the equation, invoke Apply, type **$** (to enter an integral), then use the Simplify manipulation:

□ $\frac{dy}{y} = \sin(x)dx$

△ $\int \frac{dy}{y} = \int \sin(x)dx$ *Applied & typed $*

△ $\ln(y) = -\cos(x)$ *Simplified*

Mode 2
Using Apply with a Single Term

In this mode, you can add *and* subtract the same expression from a selected term. For example, you can turn –2 into two –1s:

(boxes show selections prior to manipulations)

□ $x^2+y^2\boxed{-2}$

△ $x^2+y^2-2 = x^2+y^2\boxed{-2+1}-1$ *Applied & typed 1*

△ $x^2+y^2-2 = \boxed{x^2}+y^2-1-1$ *Simplified*

△ $x^2+y^2-2 = \boxed{y^2-1}+x^2-1$ *Commuted by hand*

△ $x^2+y^2-2 = (y^2-1)+\boxed{x^2-1}$ *Simplified*

△ $x^2+y^2-2 = (y^2-1)+(x^2-1)$ *Simplified*

Mode 3
Using Apply with a Fraction

In this mode you can multiply both the numerator and denominator of a fraction by the same expression.

□ $\frac{\cos(x)}{1+\sin(x)}$

△ $\frac{\cos(x)}{1+\sin(x)} = \frac{(1-\sin[x])\cos(x)}{(1-\sin[x])(1+\sin[x])}$ *Applied & typed* (1 – sin(*x*))

△ $\frac{\cos(x)}{1+\sin(x)} = \frac{(1-\sin[x])\cos(x)}{-(\sin[x])^2+1}$ *Expanded denominator*

△ $\frac{\cos(x)}{1+\sin(x)} = \frac{(1-\sin[x])\cos(x)}{-(-[\cos(x)]^2+1)+1}$ *Transformed* (sin[*x*])²

When you Transform a dialog offers two possible transformations; accept the simpler form.

△ $\frac{\cos(x)}{1+\sin(x)} = \frac{(1-\sin[x])\cos(x)}{(\cos[x])^2}$ *Simplified denominator*

△ $\frac{\cos(x)}{1+\sin(x)} = \frac{-\sin(x)+1}{\cos(x)}$ *Simplified fraction*

Mode 4 Using Apply with a Product

In this mode you can multiply a term by an expression and the inverse of that expression.

$$\square\, 2\left(\sqrt{x}-1\right)$$

$$\triangle\, 2\left(\sqrt{x}-1\right)=2\left(\sqrt{x}-1\right)\left(\sqrt{x}+1\right)\frac{1}{\sqrt{x}+1}$$

Applied & typed sqrt(*x*) + 1

$$\triangle\, 2\left(\sqrt{x}-1\right)=2\left(x-1\right)\frac{1}{\sqrt{x}+1}$$

Expanded parenthetic terms

$$\triangle\, 2\left(\sqrt{x}-1\right)=2\,\frac{x-1}{\sqrt{x}+1}$$

Simplified

Apply does not work with multiple selections.

Transform

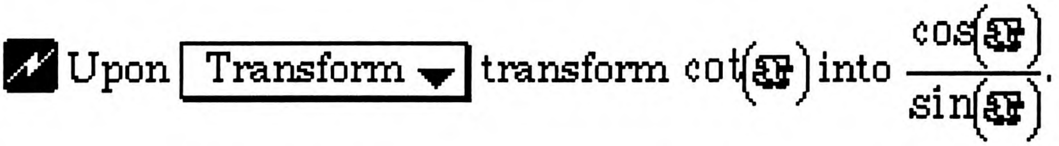
The transform manipulation is designed to be a flexible and extensible way to invoke predefined or user-defined algebra rules. Transform searches through your notebook for transformation rules that match a given algebraic pattern, and then changes those situations to a different expression based upon a replacement pattern.

To invoke Transform, simply select the expression you want to transform and choose **Transform** from the **Manipulate** menu or click on the functions palette. For instance, if you transform the expression $\cot(\sqrt{y})$ Theorist gives you the result

$$\triangle\, \cot\left(\sqrt{y}\right)=\frac{\cos\left(\sqrt{y}\right)}{\sin\left(\sqrt{y}\right)}$$

Transform

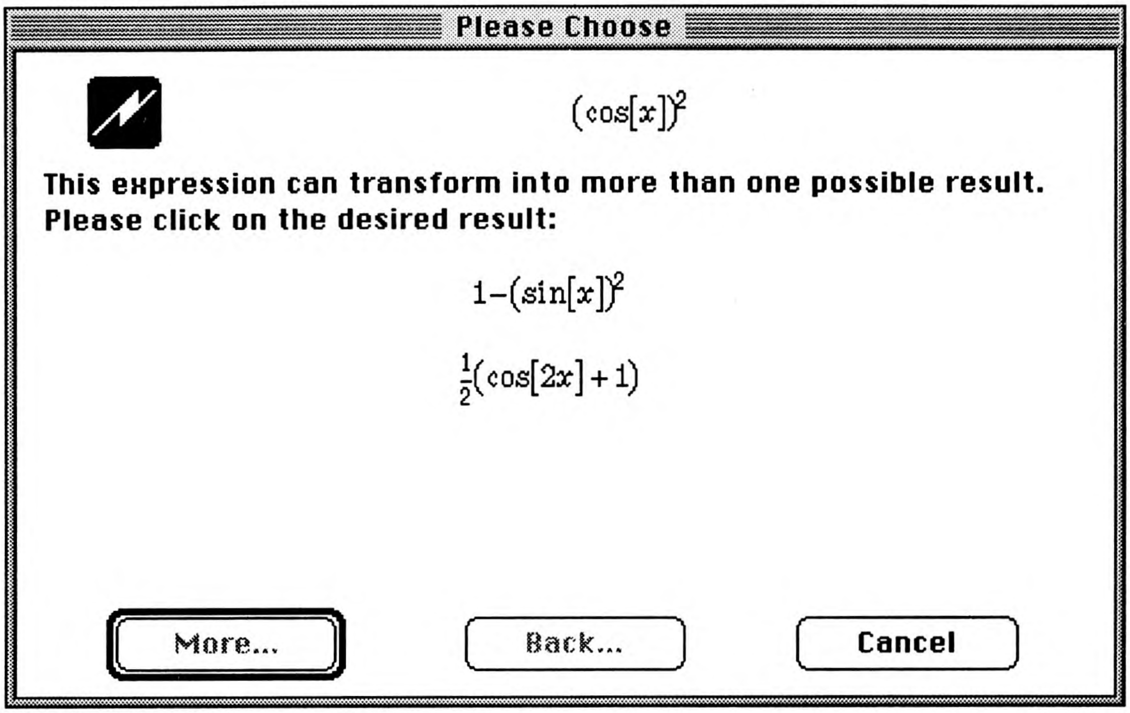
because your notebook contains the following rule.



Upon Transform ▾ transform $\cot(\mathfrak{x})$ into $\dfrac{\cos(\mathfrak{x})}{\sin(\mathfrak{x})}$.

The wildcard $\mathfrak{x}$ represents any expression of a given type. In this case, $\mathfrak{x}$ represents sqrt(y). (See the Wildcard Variables section within the Expressions section of this manual.)

In many cases, there will be more than one rule that matches a given expression. For example, if you try to transform $(\cos[x])^2$ you will get the following dialog because your notebook has more than one rule which matches this pattern.



$(\cos[x])^2$

This expression can transform into more than one possible result. Please click on the desired result:

$1-(\sin[x])^2$

$\frac{1}{2}(\cos[2x]+1)$

When you are presented with this dialog, click on the alternative you want to appear in your notebook.

By default, your notebook has several dozen transformation rules in it, each describing some sort of algebraic identity. You can open the Declarations section of your notebook and tinker with the insides if you feel brave enough to change your rules of algebra. For more power, you can add more rules, or for greater speed, you can remove them. You can derive rules yourself, or you can get them from notebooks in the Mathematics directory, or from colleagues.

Transform seeks to identify expressions of particular forms and turn them into different forms. For example, a transformation rule such as:

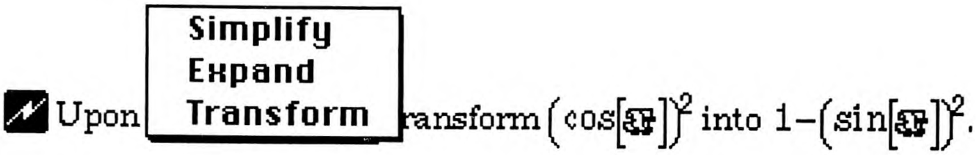
Upon Transform ▾ transform $(\cos[\mathfrak{x}])^2$ into $1-(\sin[\mathfrak{x}])^2$.

...turns any selected expression of the form $(\cos[\mathfrak{x}])^2$ into an expression of the form $1-(\sin[\mathfrak{x}])^2$, where $\mathfrak{x}$ can represent almost any expression.

Use wildcard variables (e.g., $\mathfrak{x}$, $\mathfrak{y}$) to represent expressions in transformation rules. Wildcard variables match any variable of a given type. The type of variable is determined by the letter used. (See the Wildcard Variables section in the Expressions section of this manual.)

A Transform manipulation works very much like a Substitute manipulation except the substitution equation is not an equation. The substitution equation is contained in the transformation rule. You select only the target expression, rather than the target *and* the substitution equation.

You can also make transformation rules execute during Simplify, Expand, and Collect manipulations. Choose the desired manipulation from the rule's pop-up menu:



Upon ransform $(\cos[\,])^2$ into $1-(\sin[\,])^2$.

Collect reverses the action of transformation rules that are set to execute "Upon Expand."

In some situations, more than one transformation rule that executes with the Simplify or Expand manipulation will match a selected expression. In these cases, rather than displaying a dialog box, the manipulation executes the first matching rule. The manipulation searches the current theory from top to bottom, then the parent theory from top to bottom, and so on.

Transformation rules that execute "Upon Simplify" are carried out *after* executing the Simplify manipulation, whereas transformation rules that execute "Upon Expand" are carried out *before* executing the Expand manipulation.

The Transform manipulation is not recursive. It executes only once on the outermost layer of an expression. If nested expressions can still be transformed, repeat the manipulation. To make a Transform manipulation recursive, change the pop-up menu to Simplify or Expand. Both of these manipulations execute recursively.

Taylor Series

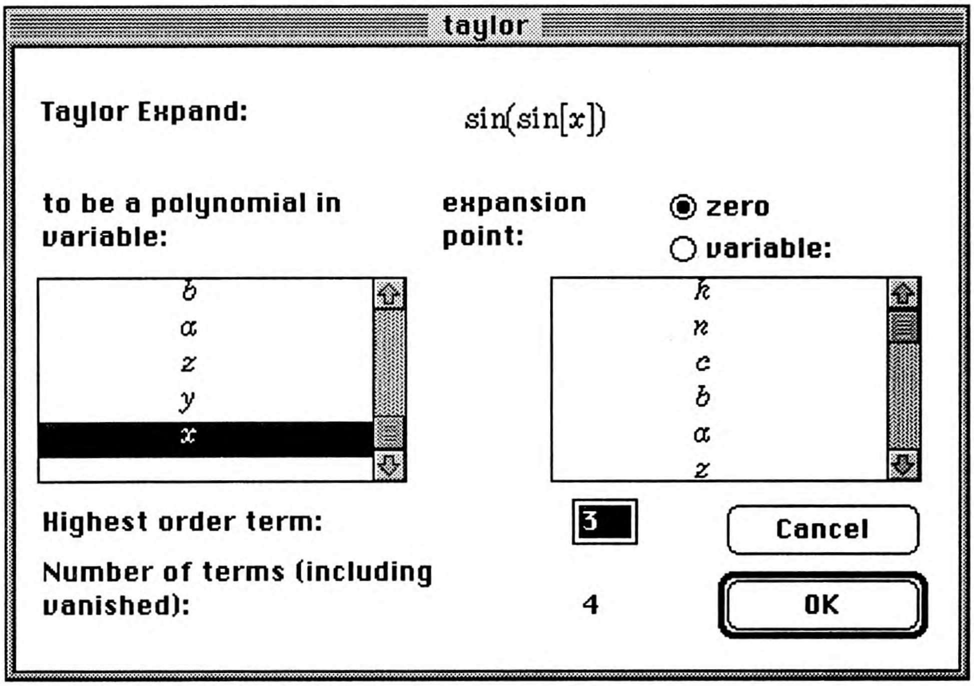
The Taylor Series manipulation is specifically designed to generate the Taylor series of a continuous function. A Taylor series is a polynomial that approximates such a function. A single point of the function, usually zero, is designated as the expansion point. At that point, the Taylor series approximation is exact. The approximation deteriorates as you move away from that point. The higher the order of the polynomial, the better the approximation.

A Taylor series can have terms up to the 99^{th} order (e.g., x^{99}). However, computing a Taylor series of this complexity may be inordinately time-consuming.

The expansion point for a Taylor series can be zero, another constant, or a variable. If you define it as a variable, you can generate distinct Taylor expansions in alternate case theories.

To create a Taylor series, select a function and choose from the pop-up or choose **Manipulate▸Other▸Taylor Series**. A dialog box asks you to verify the variable to use, the highest order term in the expansion, and the point of expansion. You can create several Taylor expansions at the same time by selecting multiple functions and then choosing **Taylor Series**.

The Taylor Series manipulation presents the following dialog:



taylor

Taylor Expand: $\sin(\sin[x])$

to be a polynomial in variable: x

expansion point: ● zero ○ variable:

Highest order term: 3

Number of terms (including vanished): 4

A Taylor series is generated by taking repeated derivatives of an expression, once for each order. Therefore, it is reasonable to generate an arbitrarily large Taylor series for polynomials, certain trigonometric functions (sin and cos), certain hyperbolic functions (sinh and cosh), and some exponential expressions (e^x). But most functions, including some trigonometric functions (tan, sec, tanh, sech), rational functions (a fraction with polynomials in the numerator and denominator) algebraically explode.

To create a multidimensional Taylor series, declare the variables independent, select an expression, and create a Taylor expansion once for each variable.

For example, to create a Taylor series in two dimensions, use the following steps. Create an equation where one variable is a function of two independent variables:

□ $z = 1.5^x \sin(y)$

Declare the variables x and y independent of each other:

The variables (x, y) are independent of each other ▾.

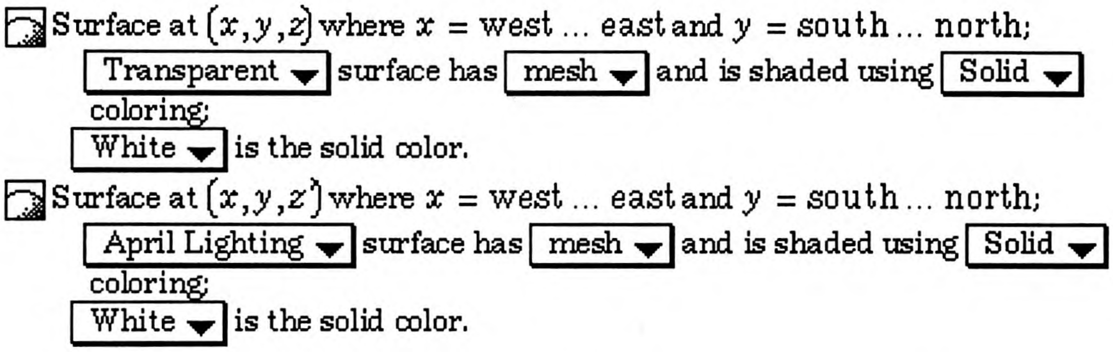
Taylor expand the right-hand side of the equation twice. For the first one, select x as the polynomial variable (in the left-hand scroll box of the dialog box). For the second, designate y as the polynomial variable.

If you choose third-order expansions (the default), the Taylor series approximation of the expression is:

$$z = \frac{1}{6}(\ln[1.5])^3\left(-\frac{1}{6}y^3 + y\right)x^3 + \frac{1}{2}(\ln[1.5])^2\left(-\frac{1}{6}y^3 + y\right)x^2$$
$$+\left(-\frac{1}{6}y^3 + y\right) + \ln(1.5)x\left(-\frac{1}{6}y^3 + y\right)$$

You will have let Theorist know about some variables.

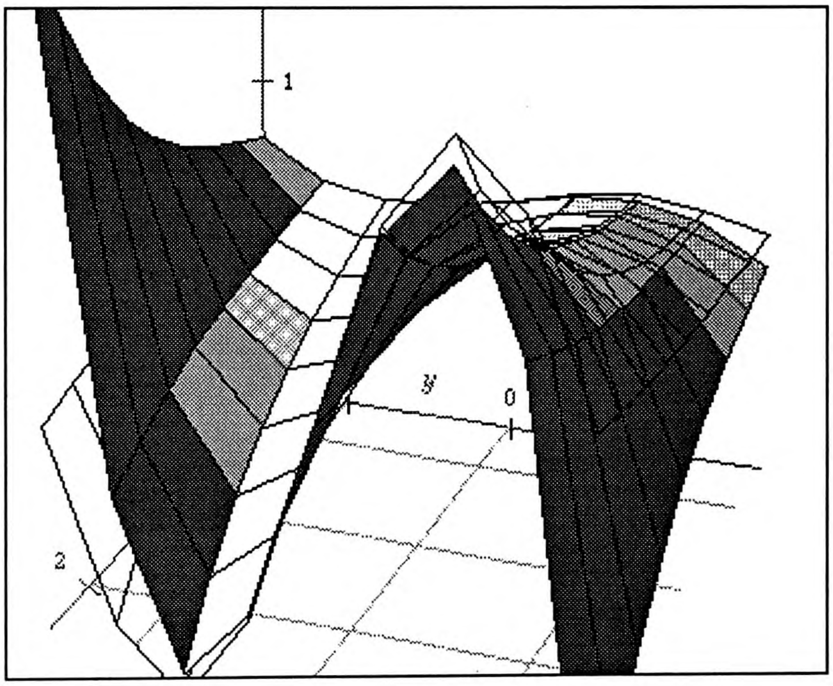
To graphically compare the original equation to the Taylor expansion, set the approximation equal to z', and create a graph with a Surface plot for each equation. To create a graph as shown in the following figure, create an Illuminated 3D graph from the original equation. Then select the expansion equation and choose **Graph▸Additional▸Add Surface Plot**. Finally, adjust the first plot proposition to display a transparent surface, sand the second to display illuminated, so the propositions appear as:



Surface at (x, y, z) where x = west … east and y = south … north;
Transparent ▾ surface has mesh ▾ and is shaded using Solid ▾ coloring;
White ▾ is the solid color.

Surface at (x, y, z') where x = west … east and y = south … north;
April Lighting ▾ surface has mesh ▾ and is shaded using Solid ▾ coloring;
White ▾ is the solid color.

The graph itself, with two Surface plots, should appear as follows (after you rotate it for a better view). Note how the Taylor series approximation (shown in gray) is exact at (0,0), but veers away as x and y increase.



Integrate by Parts

The Integrate by Parts manipulation is a useful technique for integration. If an integrand is the product of two expressions, one of which is easy to integrate, and the other is easy to differentiate, integration by parts can lead to a solution. The rule it uses is:

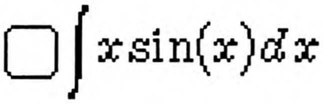
$$\int udv = uv - \left(\int vdu\right)$$

When you integrate by parts, you always get another integral, but the new integral may be easier to solve or may provide some leverage for solving the original integral.

To use the Integrate by Parts manipulation, select the *dv* part of the expression, and choose ∫udv from the 🔨 pop-up or choose **Manipulate▸Other▸Int. by Parts**.

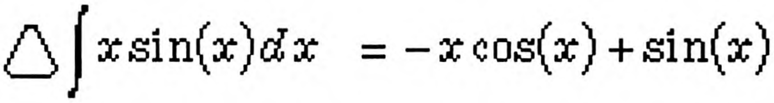
Integrate by Parts is commonly used where you have a low power of *x* multiplied by a transcendental function that is easy to integrate.

For example, consider the following integral.



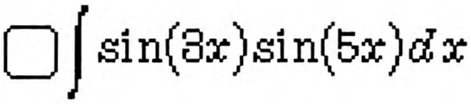
□ $\int x\sin(x)dx$

Select sin(*x*)*dx* and Integrate by Parts.



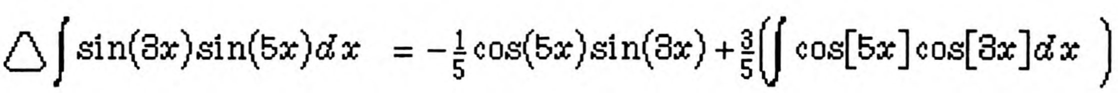
△ $\int x\sin(x)dx \quad = -x\cos(x)+\sin(x)$

As another example of using Integrate by Parts, consider the following integral.



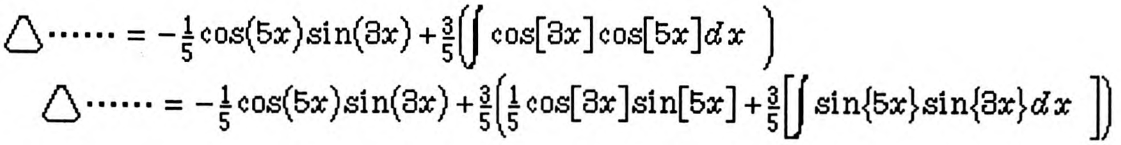
□ $\int \sin(3x)\sin(5x)dx$

To find a solution for this integral, select sin(5*x*) *dx,* and Integrate by Parts.



△ $\int \sin(3x)\sin(5x)dx \quad = -\frac{1}{5}\cos(5x)\sin(3x)+\frac{3}{5}\left(\int \cos[5x]\cos[3x]dx\right)$

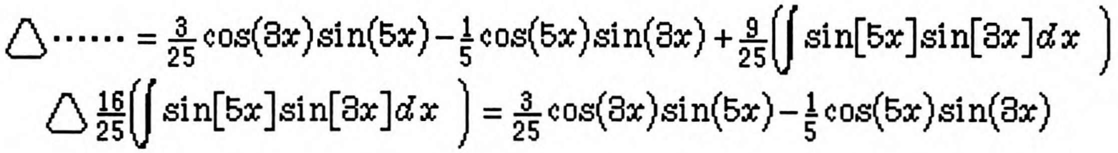
Commute the two cosine terms of the integral on the right, so that you can select cos(5x)dx, and Integrate by Parts again.



$$\triangle\cdots\cdots = -\tfrac{1}{5}\cos(5x)\sin(3x)+\tfrac{3}{5}\left(\int \cos[3x]\cos[5x]dx\ \right)$$

$$\triangle\cdots\cdots = -\tfrac{1}{5}\cos(5x)\sin(3x)+\tfrac{3}{5}\left(\tfrac{1}{5}\cos[3x]\sin[5x]+\tfrac{3}{5}\left[\int \sin\{5x\}\sin\{3x\}dx\ \right]\right)$$

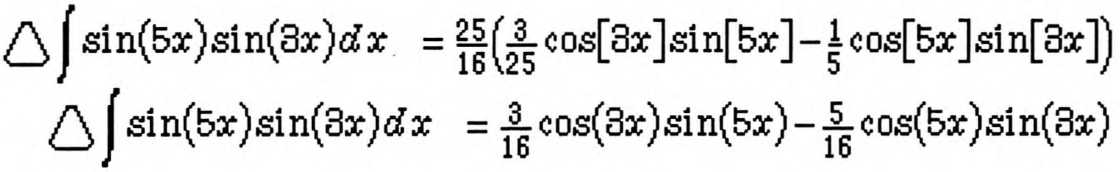
Expand the entire right side, and use the Move Over manipulation to consolidate the integral on the left side.



$$\triangle\cdots\cdots = \tfrac{3}{25}\cos(3x)\sin(5x)-\tfrac{1}{5}\cos(5x)\sin(3x)+\tfrac{9}{25}\left(\int \sin[5x]\sin[3x]dx\ \right)$$

$$\triangle\ \tfrac{16}{25}\left(\int \sin[5x]\sin[3x]dx\ \right) = \tfrac{3}{25}\cos(3x)\sin(5x)-\tfrac{1}{5}\cos(5x)\sin(3x)$$

Solve for the integral by selecting the entire integral expression (click on the integral sign) and dragging it over the proposition icon. Expand the right side one more time to see the relationship more clearly.



$$\triangle\int \sin(5x)\sin(3x)dx\ = \tfrac{25}{16}\left(\tfrac{3}{25}\cos[3x]\sin[5x]-\tfrac{1}{5}\cos[5x]\sin[3x]\right)$$

$$\triangle\int \sin(5x)\sin(3x)dx\ = \tfrac{3}{16}\cos(3x)\sin(5x)-\tfrac{5}{16}\cos(5x)\sin(3x)$$

Tables

INTRODUCTION

A table is a model of a smooth, continuous function defined by function values at regular intervals. It is the computerized equivalent of the printed look-up tables of logarithms or trigonometric functions that were in common use before pocket calculators incorporated these functions. With Theorist, a table for any kind of function can be easily generated. Alternatively, you can create your own tables by importing data from an external source or entering the numbers by hand.

With Theorist, tables are much more efficient at storing large quantities of numbers than expressions. Whereas matrices of over 100 rows can get unwieldy, tables can have tens of thousands of entries without problems. Memory size is the only limit of table size.

You can create a table from an expression, a graph, or from external data. Examples of how to create a table using each method are described in the next few sections.

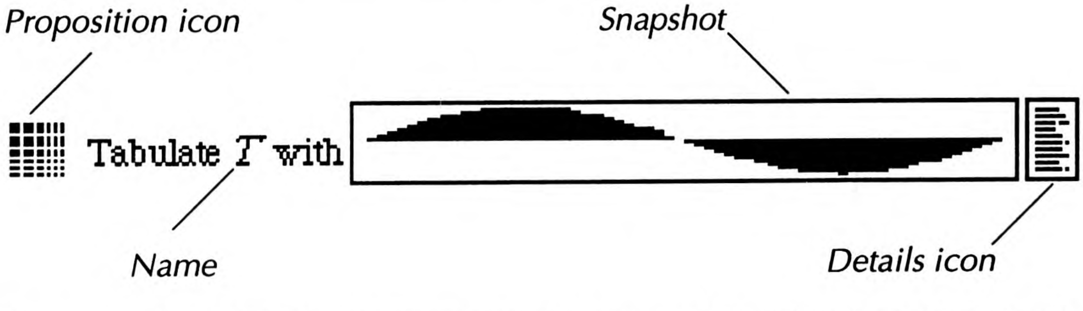
Table Types

Theorist has two table types, Real-valued and Complex-valued.

Real tables contain a list of real values for the function. Complex tables similarly contain a list of complex values. Both contain information about the allowable domain, and the correspondence between the domain and range values.

Table Proposition Elements

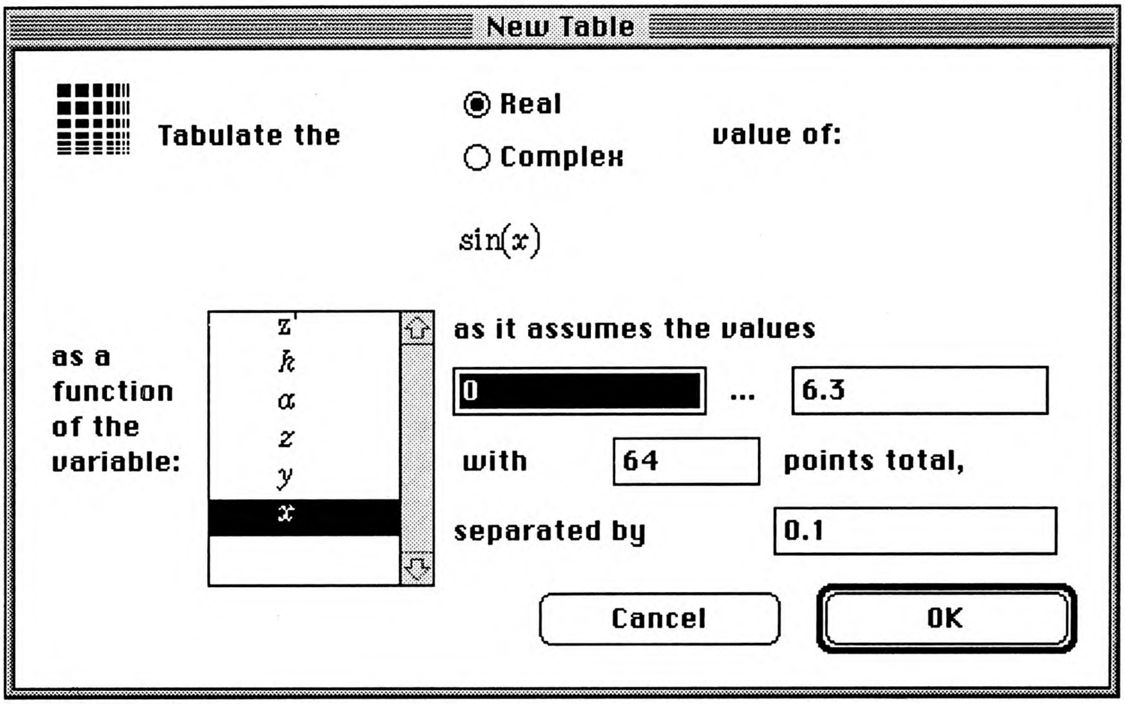
Theorist tables store numerical values in a convenient condensed format. A table proposition in a Theorist notebook consists of a proposition icon, table name, snapshot, and details icon.



Snapshot

The snapshot depicts the numerical values stored in the table. In the example above you can see the table looks like it contains values corresponding to a single cycle of a sine wave. The wider the snapshot is, the more values are in the table. (It is a logarithmic relationship, so you can see the snapshot no matter how large or small the table is.)

You can make a table like the one above. Make a new notebook and enter the expression “sin(x)”. Select it (the Select Out command helps), choose **Manipulate▸Table▸Generate...** .



New Table

Tabulate the ● Real ○ Complex value of:

sin(x)

as a function of the variable: z', k, α, z, y, x

as it assumes the values 0 ... 6.3

with 64 points total,

separated by 0.1

The table creation dialog allows you to specify the type of table (real or complex), the domain of x values, the number of points to calculate and store, and the variable that advances over the domain of the table. Notice your expression to be evaluated, sin(x) in this case, is shown in the dialog. Do not change anything; just click **OK** or press return to accept the default settings. A table proposition is created.

Name

Note the outlined question mark. Entering a letter or word allows you to define a function using the table. Enter T as the name of the function associated with the table. Clarify the Notebook and define T as a function.

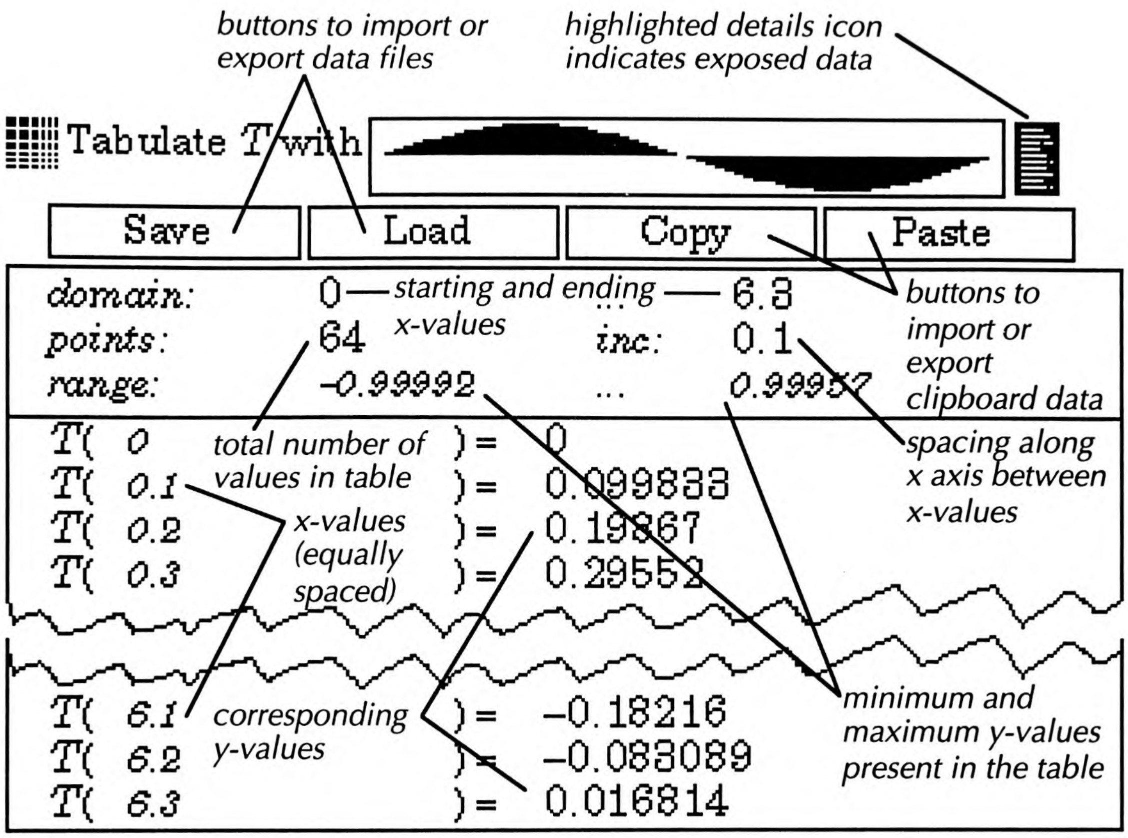
This table proposition acts like an equation that defines the function T, just like a working statement.

Details Icon

▤

The table details icon button at the right of the proposition allows you to see the details (data) stored in the table. (It behaves in a manner similar to the graph details button.) Because tables usually contain a large number of points, the data is always hidden until you expose it with the details button.

Click on the table details icon to see the data in the table.



```
domain:   0   ...   6.3
points:   64   inc:   0.1
range:    -0.99992   ...   0.99957
T( 0   ) = 0
T( 0.1 ) = 0.099833
T( 0.2 ) = 0.19867
T( 0.3 ) = 0.29552
T( 6.1 ) = -0.18216
T( 6.2 ) = -0.083089
T( 6.3 ) = 0.016814
```

buttons to import or export data files

highlighted details icon indicates exposed data

buttons to import or export clipboard data

starting and ending x-values

total number of values in table

spacing along x axis between x-values

x-values (equally spaced)

corresponding y-values

minimum and maximum y-values present in the table

Data Transfer Buttons

The four buttons immediately below the snapshot allow you to export or import data to or from external files or the clipboard. Subsequent sections explain in detail data transfer with these buttons.

Numbers

The *domain* numbers define the *x* value distribution of the data points. The *range* numbers show the minimum and maximum *y*-values present in the table. (In this discussion we use *x* and *y* to refer to the domain and range; in practice you can use any variable.) The increment number (labeled "*inc*") defines the spacing along the *x* axis between *x*-values. The *points* number shows the total number of data points in the table.

Numbers in the table not shown in italics can be edited. This is described further in a following section, Editing Tables.

Click on the table details button to hide the table details.

Complex Tables

Complex tables look very much like real tables. If your computer displays colors the table snapshot will be in color. The real part is shown as the height, as with real tables. The colors tell you the complex phase, using the same rules as Complex 3D graphs:

yellow = positive real (the lightest color)
blue = negative real (the darkest color)
strawberry red = positive imaginary
sea green = negative imaginary

Creating Other Tables

You can create a table from any expression anywhere in your notebook; simply select the expression and choose **Manipulate▸Table▸Generate...**. For example, you could select the number 5 and end up with a table full of 5's (because the value of 5 does not change as x changes). If you do not select anything, and there is no graph, then the expression will be ? and you will generate a table of unknown ? values (into which you can place values).

See the end of the Installation section for instructions on how to increase the application memory size.

Large tables require a great deal of memory. For instance, handling a table of 100,000 points uses over one megabyte of RAM (not counting overhead and everything else in your notebook and 1500K or more for Theorist itself). You may need to double that because a copy of the table may be stored in Theorist's undo buffer if it is deleted. *If you plan to use large tables, you must increase Theorist's application memory size on the Macintosh.*

EDITING TABLES

You can edit the values in a table by clicking, dragging, and typing. Just click on an entry and start typing. Pressing tab moves the selection to the next value. Note that you can not modify the numbers displayed in italics; each is fixed by other numbers elsewhere.

The domain may also be changed by editing the values specified in the table details. This is useful to remove extraneous data points or to map data from one domain into another domain.

Type '**N** for ∞.

Only numbers, not arbitrary expressions, are allowed as table entries. The numbers can be positive, negative, zero, or can be any of the three special values ∞, -∞, or **?**. The numbers carry the full fifteen digits of precision that regular real numbers in Theorist carry.

Multiple selections inside the same table are not allowed. If you want to generate many table values at once according to some mathematical scheme, make a new table using the **Manipulate▸Table▸Generate...** menu command.

TABLE DATA FORMATS

Exported Data

The data exported for a real table is stored as in the following example:

```
0
0.0998334166468281523
0.198669330795061215
0.295520206661339575
0.389418342308650492
...
```

There are no spaces or extraneous characters; numbers are separated by return characters on lines by themselves. For the most part the numbers have nineteen digits of precision, with any superfluous trailing zeros or decimal points removed (as in the first value above).

The data exported for a complex table is stored as in the following example:

```
1                        0
0.995004165278025766     0.0998334166468281523
0.980066577841241631     0.198669330795061215
0.955336489125606019     0.295520206661339575
0.921060994002885082     0.389418342308650492
...
```

One tab separates the real and imaginary parts, and the lines are separated by carriage returns again. You can paste this into a spreadsheet.

The x (domain) values are not stored in the data upon export.

Imported Data

Data being read into a table can have more latitude. The data file must have one, two or three columns of numbers, as explained above. The columns can be separated by tabs, commas, semicolons, spaces or almost any other non-numeric character. Other than that, Theorist ignores non-numeric characters such as letters (except for E's in floating point numbers), dollar signs, punctuation, spaces, and the like.

If you try to read in a two-column file into a real table, or a three-column file into a complex table, the first column will be interpreted as domain (x) values and should be (approximately) equally spaced.

Graphs

Graphing mathematical relationships is a powerful technique for visualizing and understanding the interactions of functions and variables. In Theorist, a graph is a two-dimensional space (or a representation of a three-dimensional space) for drawing plots. You can also animate any graph and display a family of related plots in a short movie.

This section discusses the fundamentals of two- and three-dimensional graphs and their plots:

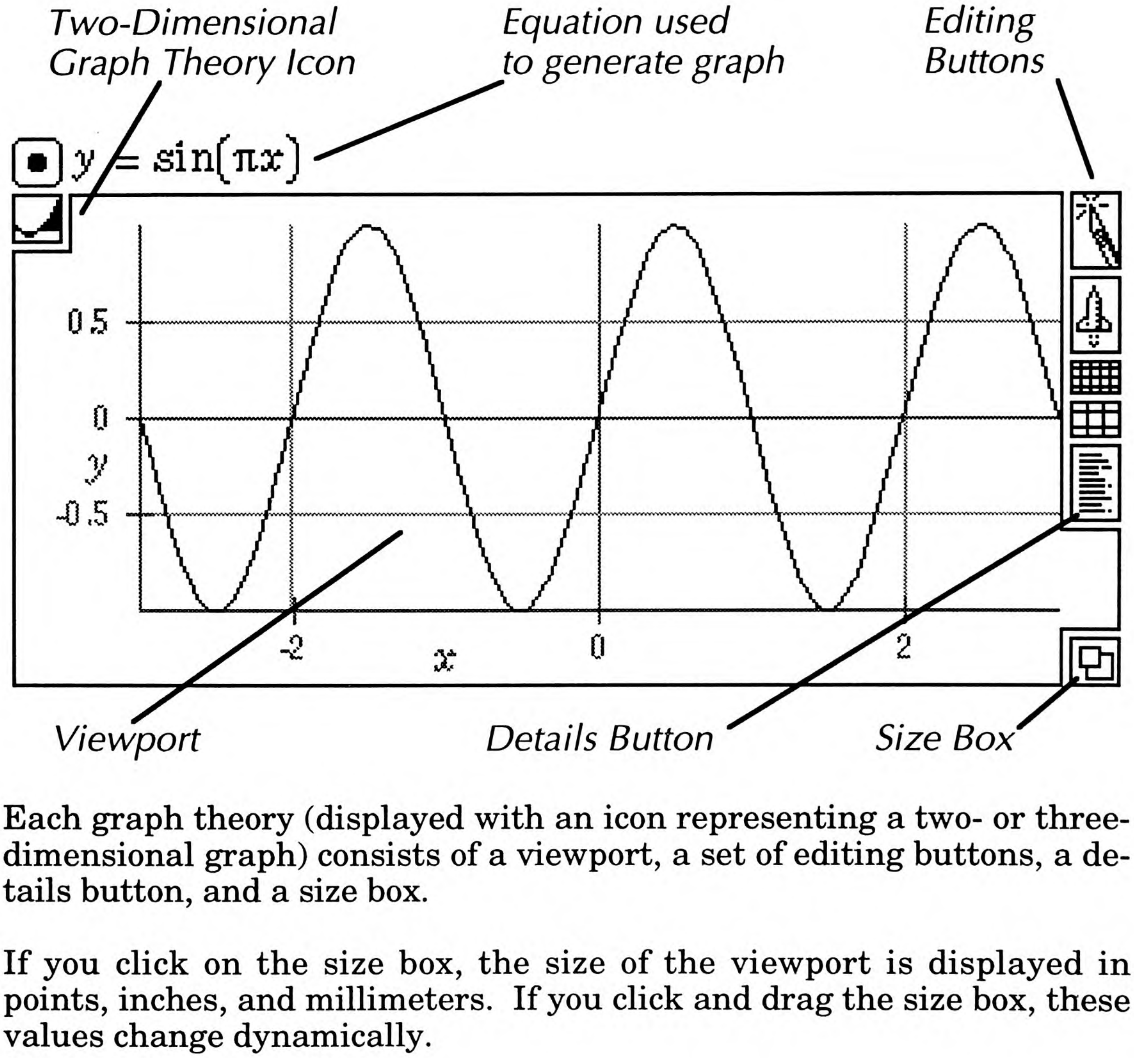
Graph theories
Two-Dimensional graph theories
Three-Dimensional graph theories
Graph proposition elements
Animation
Printing and Exporting

GRAPH THEORIES

A graph theory provides a framework for displaying Plot propositions. When you create a graph, you actually create a graph theory and a set of associated plot propositions contained within the graph theory. All definitions and settings within a graph theory (e.g., boundary values, plot resolution) are true only for that graph. Graphs, like case theories, often look outside for information when necessary, but if a sufficient definition is available internally, it takes precedence over an external definition.

This section introduces two- and three-dimensional graph theories and the different graphs that create them.

The following figure displays an equation and a two-dimensional graph created from the equation.



Each graph theory (displayed with an icon representing a two- or three-dimensional graph) consists of a viewport, a set of editing buttons, a details button, and a size box.

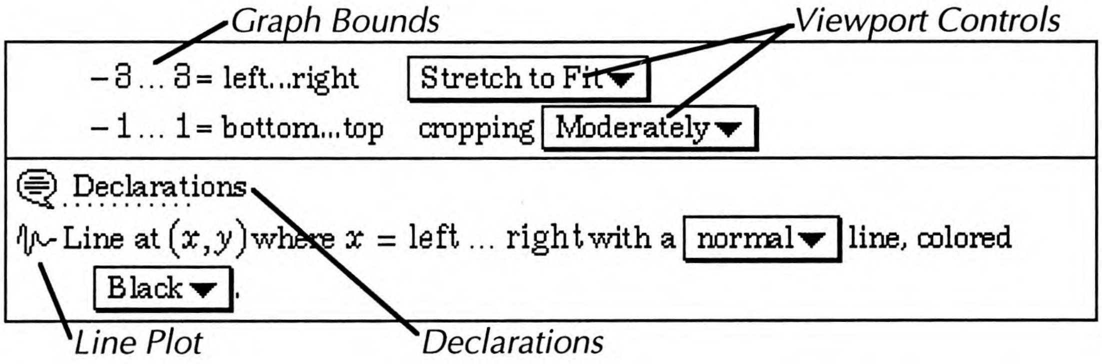
If you click on the size box, the size of the viewport is displayed in points, inches, and millimeters. If you click and drag the size box, these values change dynamically.

Click on the details button to open a box showing the graph details. Each statement in the details box is a proposition describing some element of the graph. You can alter the display of a graph with the editing buttons, or by en tering values directly into the details box.

Graph details can contain any kind and any number of propositions. However, graph details usually contain information about the viewport, declared definitions, plots, axes, and grids. If you click on the details button, a box opens displaying the graph details.

The details are like a fold-down control panel on a television. You can quickly create a graph and easily adjust your view with the editing buttons and the open hand pointer, but if you want to fine tune a graph's appearance or make a unique or special graph, adjust the details.

Following are the graph details for a simple two-dimensional graph with a single line plot.



$-3 \ldots 3$ = left…right Stretch to Fit ▼

$-1 \ldots 1$ = bottom…top cropping Moderately ▼

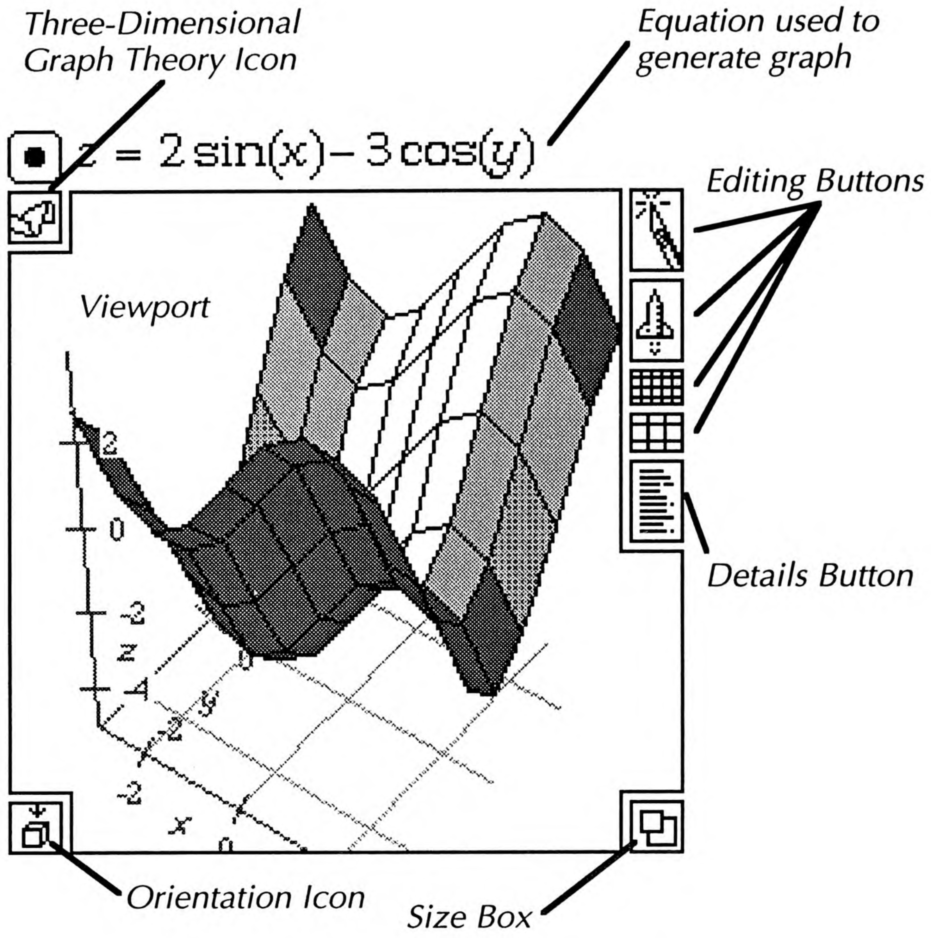
Declarations

Line at (x,y) where x = left … right with a normal ▼ line, colored Black ▼.

There are also axis and grid propositions collapsed under the Declarations comment. Expose the Declarations to see them.

Two-dimensional and three-dimensional graph theories are very similar. Three-dimensional graph theories include an *orientation* icon in the lower left-hand corner, which indicates the side of the graph shown in the viewport. Three-dimensional graph theories most often display three-dimensional plots (e.g., surfaces), although they can also display Line, Scatter, or Contour plots.

The following figure displays an equation and a three-dimensional graph created from the equation.



Plots in a graph theory are propositions. They are displayed in a graph's viewport. Plot propositions include Line, Scatter, Surface, and Contour plots, as well as the Axes and Grid Lines.

When you point to the viewport with the mouse, the arrow becomes an open hand. Use this pointer on two- and three-dimensional graphs to click-and-drag the displayed image. This is the most direct way to change your view of a graph. The editing buttons at the right of the viewport also give you quick ways for changing your view. They perform the following functions:

Knife	Cut out a small portion of a plot to view it in greater detail.
Rocket	Zoom out (or in, using the Option key) by a factor of two.
Higher Resolution	Increase the number of data points (to create a smoother plot).
Lower Resolution	Decrease the number of data points (to create a rougher but faster-drawn plot).
Details	Show (or hide) the graph details.

Plots, Axes, and Grid Lines are all graphic objects drawn into a graph's viewport. The details box contains the propositions for each of these objects as well as a set of definitions used by the graph. These propositions, like all Theorist propositions, are not programming commands; their order does not affect their function. Each is an independent description of a graphical object.

Creating Graph Theories

To create a graph theory, select one or more equations or tables that specify relationships between variables. Then choose the type of plot you want to use to display this relationship. The submenus of the Graph menu provide numerous ways of creating different types of plots.

It is important to determine which variables should be *independent* (ranging freely between the boundary values) and which should be *dependent* (functions of the independent variables).

Items on the Graph submenus let you plot:

One dependent variable against one independent variable
One dependent variable against two independent variables
Other combinations of variables
Discrete points stored in a matrix or table

y = f(x)
1 Dependent
1 Independent

Use one of the items on the y = f(x) submenu to construct a graph that displays the relationship between two variables where one is a function of the other (one independent variable, and one dependent variable). There are six such graphs: Linear, SemiLog, SemiLog-X, Log-Log, Polar, and Complex 3D.

Graph	Graph Theory	Description
Linear;	2 D	The independent variable is displayed on the horizontal (*x*) axis and the dependent variable is displayed on the vertical (*y*) axis. Both scales are linear.
SemiLog	2 D	The independent variable is displayed on the horizontal (*x*) axis (linear) and the dependent variable is displayed on the vertical (*y*) axis (logarithmic).
SemiLog-X	2 D	The independent variable is displayed on the horizontal (*x*) axis (logarithmic) and the dependent variable is displayed on the vertical (*y*) axis (linear).
Log-Log	2 D	The independent variable is displayed on the horizontal (*x*) axis and the dependent variable is displayed on the vertical (*y*) axis. Both scales are logarithmic.

Polar	2 D	The independent variable is displayed along the circumference (θ) of a circle and the dependent variable is displayed along the radius (*r*) of the circle.
Complex 3D	3 D	The real component of the independent variable is displayed along the West-East axis and the imaginary component is displayed along the South-North axis. The absolute value of the dependent variable is displayed along the Bottom-Top axis and the complex phase is displayed as the surface color.

z = f(x, y)
1 Dependent
2 Independent

Use an item from the z = f(x, y) submenu to construct a graph that displays the relationship between three variables where one is a function of the other two (two independent variables, and one dependent variable). There are six such graphs: Density 2D, Contour 2D, Color 3D, Illuminated 3D, Spherical 3D, and Cylindrical 3D.

Graph	Graph Theory	Description
Density 2D	2 D	The independent variables are displayed along the horizontal (*x*) and vertical (*y*) axes. The dependent variable is displayed as a white-to-black shading (white = 0, or less; black =1, or more)
Contour 2D	2 D	The independent variables are displayed along the horizontal (*x*) and vertical (*y*) axes. The dependent variable is displayed as a collection of labeled contour lines.
Color 3D	3 D	The independent variables are displayed along the West-East and South-North axes. The dependent variable is displayed as a surface along the Bottom-Top axis using a specified range of colors depending on the elevation.

Illuminated 3D	3 D	The independent variables are displayed along the West-East and South-North axes. The dependent variable is displayed as a surface along the Bottom-Top axis. The surface is solid white illuminated by a light source over your left shoulder.
Spherical 3D	3 D	The independent variables are displayed as the latitude, measured from the top rather than the equator (θ), and longitude (φ). The dependent variable is displayed as the distance from the center origin.
Cylindrical 3D	3 D	The independent variables are displayed as the longitude (θ) and height. The dependent variable is displayed as the radius (*r*).

Other

Use one of the items on the Other submenu to construct a graph that displays a special relationship between variables.

x = f(t), y = g(t) Parametric
2 Dependent
1 Independent

Use this item on the Other submenu to construct a graph that displays the relationship between three variables where two are functions of a third variable The result is a two-dimensional parametric graph.

Graph	Graph Theory	Description
Parametric	2 D	The independent variable is not explicitly displayed, but varies along the length of the curve. The dependent variables are displayed along the left-right (*x*) and bottom-top (*y*) axes.

x = f(t), y = g(t). z = h(t) Space Curve
3 Dependent
1 Independent

Use this item on the Other submenu to construct a graph that displays the relationship between four variables where three are functions of a fourth variable. The result is a three-dimensional parametric graph.

Graph	Graph Theory	Description
Space Curve	3 D	The independent variable is not explicitly displayed, but varies along the length of the curve. The dependent variables are displayed along the west-east (x), north-south (y), and bottom-top (z) axes.

f(x, y) = g(x, y)
Implicit
2 Dependent
1 Independent

Use this item on the Other submenu to construct a graph that displays the relationship between two independent variables. The result is a two-dimensional graph with an implicit plot showing a contour line where the equation holds true.

Graph	Graph Theory	Description
Implicit	2 D	The independent variables are displayed along the horizontal (x) and vertical (y) axes. There are no dependent variables.

Because graphs theories *are* theories, they can contain other theories. See the file "Spherical Family" on the Theorist distribution disks for an example of using case theories within graph theories.

Adding Plots

You can move plots between graph theories without difficulty, using the copy and paste commands. However, it's often easier to select an equation and choose the appropriate menu option on the Graph menu to add plots, axes or grid lines to an existing graph theory.

To add a plot (or axis or set of grid lines) to an existing graph theory, select an equation (and a graph theory if there is more than one), and choose **Add Line Plot**, **Add Surface Plot**, **Add Contour Plot**, or **Add Scatter Plot** (or **Add Axes** or **Add Grid Lines**) from the **Graph▸Additional** submenu.

Two-Dimensional Graphs

Two-dimensional graphs consist of a two-dimensional graph theory (the shell) with one or more two- or three-dimensional plots (Line, Scatter, Surface, or Contour) and associated axes, grid lines, and name declarations. This section presents:

- A list of graphs which create two-dimensional graph theories
- Scrolling a two-dimensional graph for a better view
- Icons for editing two-dimensional graph theories
- Graph details of two-dimensional graph theories
- Descriptions of plots
- Techniques for finding roots of a line plot or two contour plots

Creating 2D Graphs

Choose one of the following types of graphs to generate a two-dimensional graph theory.

Graph Type	Submenu
Parametric x = f(t), y = g(t)	Other
Linear	y = f(x) *or* Scatter
SemiLog	y = f(x) *or* Scatter
SemiLog-X	y = f(x) *or* Scatter
Log-Log	y = f(x) *or* Scatter
Polar	y = f(x) *or* Scatter
Density	z = f(x, y)
Contour	z = f(x, y)
Implicit f(x,y) = g(x,y)	Other

Parametric, Linear, SemiLog and Polar graphs all generate Line plots from the selected function and are drawn with a Normal line. This line can be adjusted with the proposition's line style pop-up menu to be normal, heavy, dotted, or dashed. Its color can be changed with the other pop-up menu. Density graphs generate Surface plots, and Implicit graphs and Contour graphs generate Contour plots.

Default bounds of –3 and 3 are often used when a graph is created. (In many cases other default bounds are generated.) These bounds set the domain of the independent variable(s). Before creating a graph, you can explicitly set one or both boundary values with an inequality (e.g.,

$x > 0$, $x \leq 100$). If only one bound is set, the other bound is guessed by Theorist. You can also edit these bounds in the details once the graph is created.

Scrolling

The most direct way to change your view of a two-dimensional graph theory is to use the mouse. When the pointer is over the viewport, it turns into an open hand icon. Click, drag, and release the image as if it were a piece of paper and the image redraws with new boundary values.

The manner in which it scrolls depends upon the graph's resolution setting (see below) and the redraw setting in the Graph Preferences (explained later). If the graph resolution is low, it redraws as you drag and always looks right. If the graph resolution is high, you will drag an image of the middle of the graph around, crudely imitating the way in which it finally looks when you release the mouse button.

You can also use the editing icons or change values in the graph details box to change the view on your graph.

Editing Icons

Each 2-D graph proposition is bordered by its own set of editing icons for changing various characteristics of the graph.

Knife

Knife Cursor

Super Knife Cursor

Click on the Knife icon to *pick up* the knife (the cursor becomes a knife). Click and drag across a rectangular portion of the graph to look at just that part. The portion of the graph you select can be most of the plot or just a tiny portion. Using the Knife changes the values of the graphs bounds.

Double-click the Knife icon to activate the Super Knife, which stays "in your hand" for repeated knife-zooming until you click the Knife icon to make it go away. While active, Undo undoes *all* of your slices and returns the graph to its original view.

If, while slicing with either knife, you change your mind or make a mistake, you can abort the slice by dragging the knife back to its original "incision" location and release the mouse button. (Even if you do not do it perfectly, Theorist assumes your very small slice means you want to abort the operation.)

Note that you can also use the knife to zoom outward. Start by clicking inside the graph viewport near one corner then drag toward the opposite corner. Keep dragging past the corner until you think you have enclosed what you want to see.

Rocket

Click on the Rocketship icon to *zoom out* (from the center of the viewport) by a factor of two. After a moment, the graph redraws. To zoom by more than a factor of two, click the icon repeatedly and quickly. To zoom by a factor of 1000, click ten times.

Be sure the keys are held when you click and release the mouse button.

Hold the option key (Macintosh) or alt *and* Ctrl keys (Windows) and click on the Rocketship to *zoom in* by a factor of two. To view a small portion of the graph, it is faster to use the Knife. Zooming in and out is reflected in the values of the graph bounds.

Resolution Icons

▦ ⊞

Click on the dense grid icon to increase the data points used to create the plots in the graph by (approximately) a factor of two. Click on the open grid icon to (approximately) halve the number of data points. The more data points used, the smoother the curved line(s), and the longer it takes to generate the image. Plot resolution is not reflected in the graph details.

Note: higher resolution settings eat memory!

When you click on a resolution icon, the current value and the destination value appear briefly to the left of the icon (e.g., "8 -> 16"). Resolution values are 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, and 1024. (The default is eight.) For Line Plots, Axes, and Grid Lines, the value indicates the approximate number of line segments used to create a 90-degree curve. For Surface Plots and Contour Plots, the value indicates the number of panels on each side of the displayed plot.

Details Box

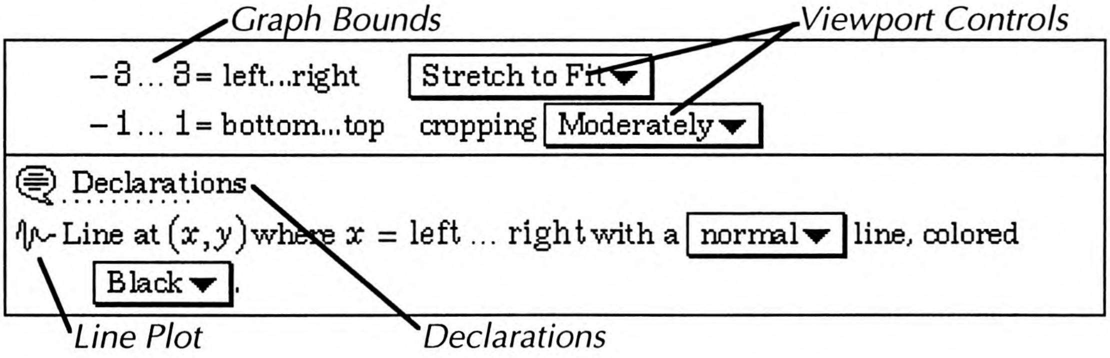
Click on the details box to open or close the graph details. The graph details section contains information about all graphical elements in a graph theory, and how they are displayed in the graph viewport. (Graph details are explained in depth below.)

Size Box

Click and drag on the size box to change the size and shape of the viewport. Larger graphs consume more memory; more pixels are required to record and display the image. Graph theories with larger viewports take slightly longer to display than smaller theories with the same image. As you drag the size box, the new dimensions are displayed. The size of the viewport is not reflected in the graph details.

Graph Details

The details box allows you to fine tune (and significantly alter) the appearance of a graph theory. The graph details section contains information about all graphical elements in a graph theory, and how they are displayed in the graph viewport.



−3 … 3 = left…right Stretch to Fit ▾

−1 … 1 = bottom…top cropping Moderately ▾

Declarations

Line at (x,y) where x = left … right with a normal ▾ line, colored Black ▾.

Plots

Each plot (and every other graphical item) is displayed with an icon followed by a sentence describing the plot's attributes. These sentences contain pop-up menus and expressions that you can use to change the plot. For example, Line plots can have different styles and colors;

Surface plots can have various optical properties and coloring schemes. Line, Contour, and Scatter plots are described in detail later in this section.

Once created, plot propositions can be adjusted, edited, and moved around like any other propositions. To move a plot, select it by its proposition icon and cut (or copy) and paste it into another graph theory. Plot propositions are meaningless outside graph theories.

If you don't select an equation you must choose graph axis variables from a dialog.

To add a line plot to a graph, select a graph and an equation (if there are more than one in the notebook), and choose **Graph▸Additional▸Add Line Plot**. A line plot is generated and added to the graph details. Added lines use different colors or styles, depending on whether the computer is using colors or black and white.

Graph Bounds

The graph bounds determine the logical height and width (in the units your data is in) of the graphing area. These data values are scaled to fit within the viewport. The actual graphing area (i.e., the on-screen rectangle defined by the left, right, bottom, and top values) is somewhat smaller than the viewport, and depends upon the two viewport controls (see below). Edit the boundary values as you would any other expression. Set the values to different numbers or to names or expressions that evaluate to numerical values.

The values displayed in the graph bounds change when you use the open hand pointer or any of the editing icons (except when you change the resolution). For example, zooming out using the Rocketship icon is the same as increasing the boundary values. (Strictly speaking, one click of the Rocketship icon is equivalent to doubling the difference between each pair of boundary values.) Similarly, selecting part of a graph with the Knife icon also changes the boundary values.

Viewport Controls

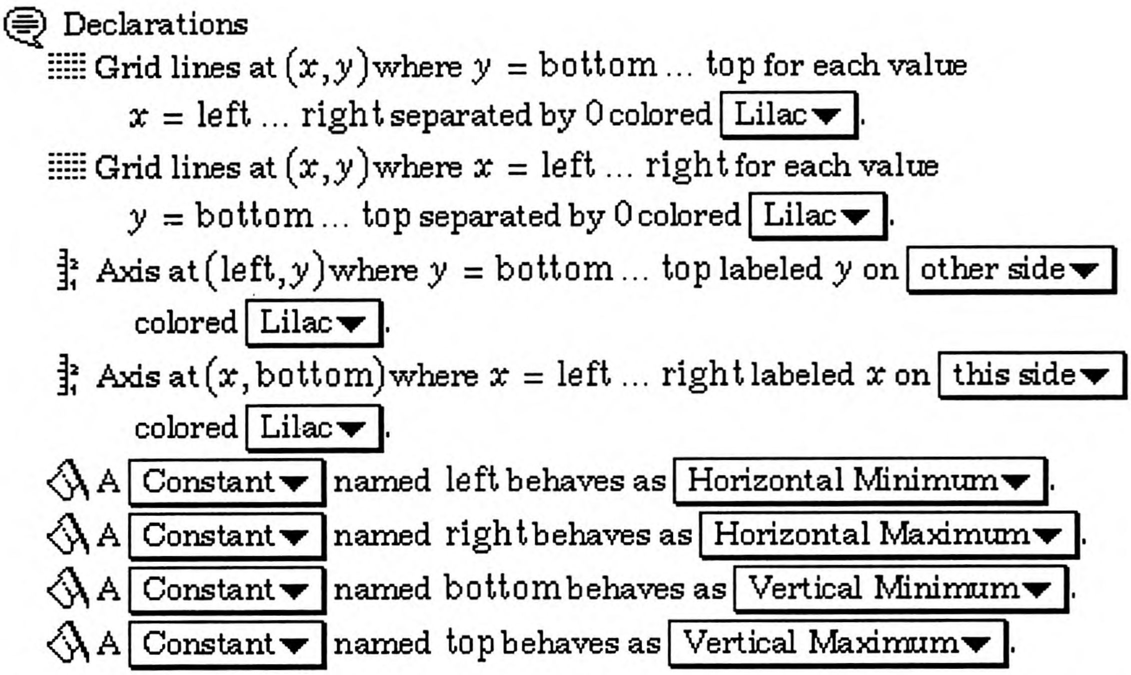
To the right of the boundary values are two pop-up menus. The first determines the *aspect ratio* and the second the degree to which the display is *cropped.*

The aspect ratio is either “True Proportions” or “Stretch to Fit.” True Proportions indicates that both axes display the same unit measure. For example, if one horizontal unit equals 1.3cm, the same is true for a vertical unit.) Stretch to Fit adjusts the scale of the axes to fit them into the height and width of the viewport. For example, this lets you graph megahertz against microkelvins without having a long, thin graph. The default display is Stretch to Fit, unless True Proportions is more appropriate for the values derived from generating the graph. Polar graphs always default to True Proportions.

The more tightly cropped the display the closer the graph bounds are to the edges of the viewport. The default is Moderately cropped. The possible settings are: Very Tightly, Tightly, Moderately Tightly, Moderately, Moderately Wide, Wide, and Very Wide. If the display is Very Tightly cropped, the graph bounds are at the same location as the viewport bounding box.

Declarations

"Declarations" is a comment proposition (collapsed by default) containing plots and definitions used to construct the graph. These include the constants bottom, top, left, and right (and sometimes north, south, east, and west). These names define the bounds of the graphing area in the viewport. Most plots use some or all of these names. The following exposed Declarations has names created by default for a Linear graph:



Declarations

Grid lines at (x,y) where y = bottom ... top for each value x = left ... right separated by 0 colored Lilac ▾.

Grid lines at (x,y) where x = left ... right for each value y = bottom ... top separated by 0 colored Lilac ▾.

Axis at (left,y) where y = bottom ... top labeled y on other side ▾ colored Lilac ▾.

Axis at (x,bottom) where x = left ... right labeled x on this side ▾ colored Lilac ▾.

A Constant ▾ named left behaves as Horizontal Minimum ▾.

A Constant ▾ named right behaves as Horizontal Maximum ▾.

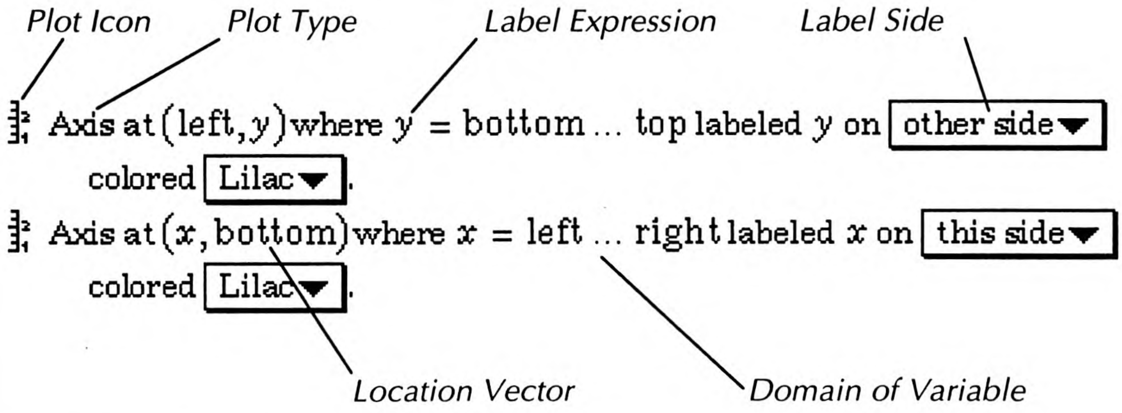
A Constant ▾ named bottom behaves as Vertical Minimum ▾.

A Constant ▾ named top behaves as Vertical Maximum ▾.

As you can see, the exposed Declarations also show Grid and Axis plot propositions, explained below.

Axes

Each Axis is displayed with an icon and a sentence describing its placement and labeling. Axes (and Grid Lines) are actually simple plots. They calculate the location of a series of points and generate a line connecting those points. This set of points is defined by a location vector. All other plots work on the same principle. By default, each two-dimensional graph theory contains a pair of axes:



Axis at (left,y) where y = bottom ... top labeled y on other side ▾ colored Lilac ▾.

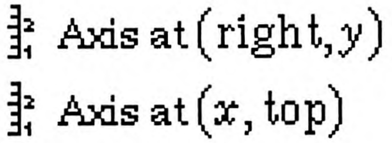
Axis at (x,bottom) where x = left ... right labeled x on this side ▾ colored Lilac ▾.

The location vector for a default axis usually consists of one of the equation variables, and a constant defined for that particular graph theory. The value of the constants "bottom" and "left" (which are assigned values in the graph bounds) are independent of the equation variables. In this example, the variable x takes on a series of values (between the graph bounds: left ... right). Each of these values is paired with the constant value for bottom. In this way a line is generated along the lower edge of the graphing area. Another line is generated on the far left of this area by the second Axis plot proposition.

Axes are generated with a set of tick marks that display a set of values of the label expression. The label expression is displayed on one side or the other of the axis between the first and second tick marks. By default, labels are taken from the equation. You can change this name to a simple expression containing the named variable (e.g., $3x$, $x/2$). If you want to display a different name (e.g., Speed), make a variable of the name and use it.

You can move axes from one side of the graph to the other (by changing the location vector), add axes to the right or top of the graphing area, and even change the location vector to be dependent on an equation.

To add a pair of Axes to a graph, select a graph (if there is more than one in the notebook), and choose **Graph▸Additional▸Add Axes**. A duplicate set of Axes are generated and added to the graph details. To make a box of axes, change the two constants bottom and left to top and right:



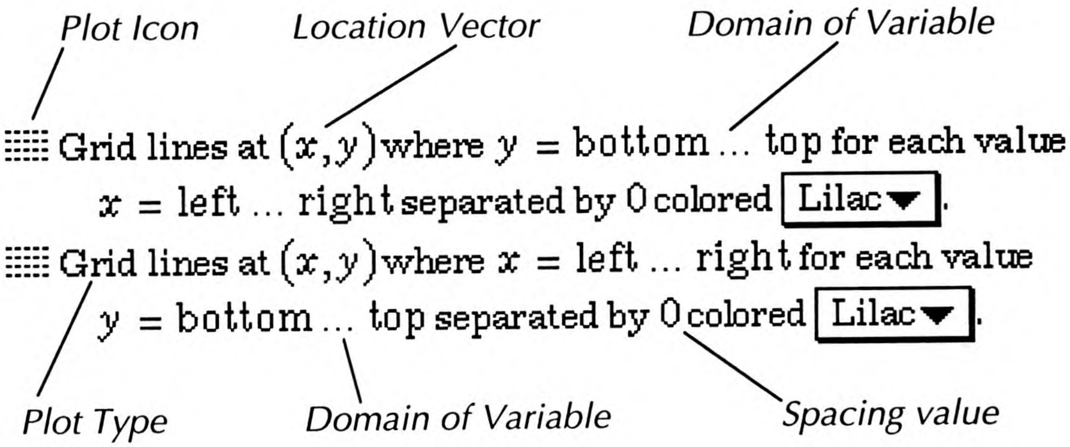
Axis at (right, y)

Axis at (x, top)

Axis and grid line propositions are like plots, and can be copied, moved, and edited in the same way.

Grid Lines

Each set of (usually parallel) Grid Lines is displayed with an icon and a sentence describing its placement. Grid Lines are also simple plot propositions. Each two-dimensional graph theory usually contains two Grid Line plot propositions:



Grid lines at (x, y) where y = bottom ... top for each value x = left ... right separated by 0 colored Lilac▼.

Grid lines at (x, y) where x = left ... right for each value y = bottom ... top separated by 0 colored Lilac▼.

As with Axes propositions, Grid Lines utilize a location vector to determine where the lines are drawn. The first proposition of this pair creates a set of vertical lines; the second creates the horizontal lines.

The spacing value determines the distance between grid lines. If the spacing value is zero (the default), one grid line is drawn to match each tick mark on the axis (if you are using default Axes). To draw a grid line at every unit, change this value to 1. For lines every half unit, set the spacing value to $\frac{1}{2}$.

Grid lines for major increments are thicker or darker than other grid lines. Major increments are at zero and at other large, round numbers. Zero is always a major increment.

To add a set of grid lines to a graph, select a graph (if there is more than one in the current notebook), and choose **Graph▸Additional▸Add Grid Lines**. A pair of grid lines are generated and added to the graph details.

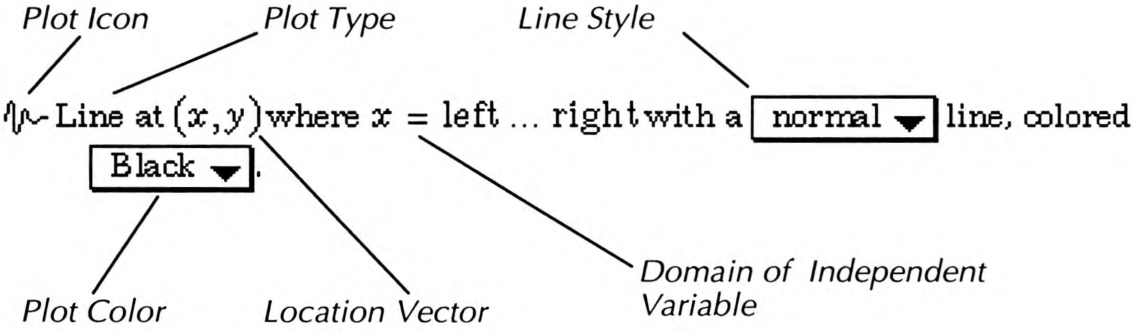
Grid Line propositions are like plots, and can be copied, moved, and edited in the same way..

Plot Types

Theorist can put any plot in a two-dimensional graph, but only Line, Contour, Density, and Scatter plots use 2-D graphs by default.

Line Plots

Line plot propositions contain the following information:

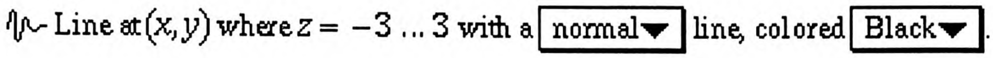


Line at (x,y) where x = left ... right with a normal ▼ line, colored Black ▼.

The location vector and the domain of the independent variable determine the point-by-point location of the plot. The first value in the location vector determines the horizontal (left ... right) location of each point, and the second value determines the vertical (bottom ... top) location. In this example, the boundary values (as set in the first line of the graph details) are used to generate a series of x values. The y values for each point are generated by plugging successive x values into the original equation.

You can replace the location vector with any expression that evaluates to a two-element vector with real values. For instance, in polar graphs the vector usually looks like "FromPolar(r, θ)."

Parametric plots work slightly differently in that the domain is set to specific numerical values not affected by the graph's boundary values:

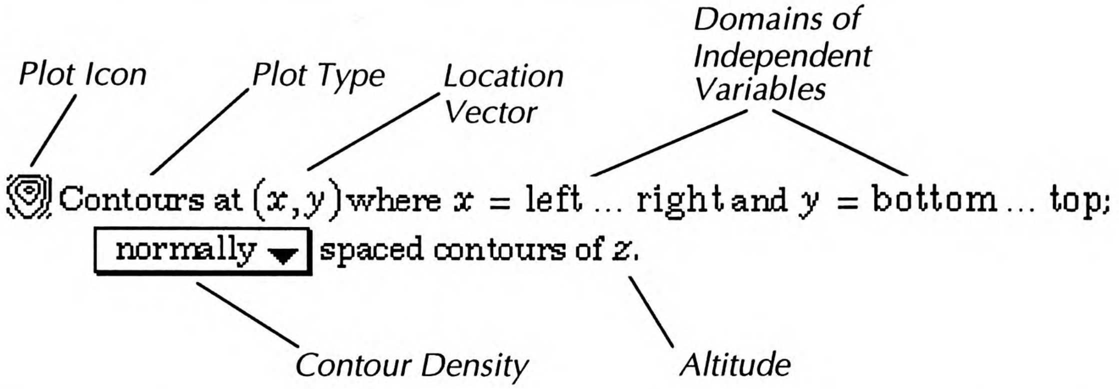


Line at (x,y) where $z = -3 \dots 3$ with a normal▼ line, colored Black▼.

If you selected a table to make the graph, Theorist automatically picks up its domain.

Contour Plots

Contour plots contain the following information:



Contours at (x,y) where x = left ... right and y = bottom ... top; normally ▼ spaced contours of z.

Contour plots work somewhat like Line plots. The first value in the location vector determines the horizontal location of a point and the second determines the vertical location, and each domain specifies a series of values to use for each. A standard Contour plot generates several lines as determined by the original equation, the density pop-up menu and the altitude value. Each contour is generated by plugging successive values of the location vector into the original equation, calculating a value for the altitude.

The altitude value (in this case z), can be any expression dependent on one or both of the domain variables. If the expression is not dependent on these values, no plot lines are drawn. With the pop-up menu set to "normally," between eight and twenty lines are generated between the minimum and maximum altitude values. The other menu settings halve or double the number of contour lines drawn. The menu choices are: at zero only (which specifies a single line), very sparsely, sparsely, normally, densely, very densely.

The spacing of contours is always some power of ten times two, five, or ten. For example, contours may be placed every 0.02 units, 0.05 units, 0.1 units, 0.2 units, or 0.5 units, and so on.

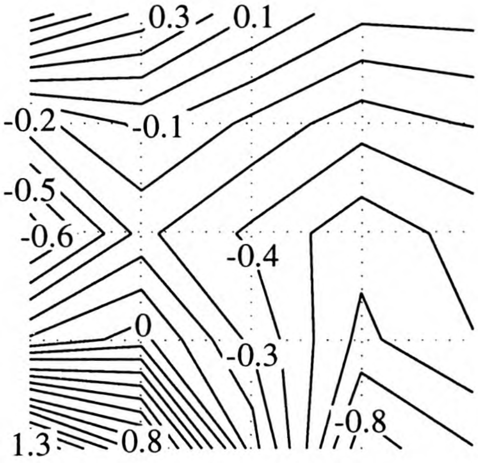
For predictable contour spacing, you must explicitly set a range of values for the altitude. Edit the altitude expression so it reads, z = minimum ... maximum. This is essential if you intend to animate a graph with a Contour plot. If the range is not specified, the minimum and maximum are taken from the altitude values themselves so Theorist may choose different contour spacings for each frame. If you specify explicit minimum and maximum this way, and the calculated altitude value is above or below the specified range, a few contour lines are drawn, but most are not.

Because the ratio between two and five is greater than two, choosing the next menu setting might *not* change the number of contour lines. For example, if the range of altitude values is $z = 0 \ldots 1$, the two settings "sparsely" and "very sparsely" draw the same number of lines.

Contour lines are labeled with their altitude values. No more than one label is generated for each altitude value. If there are multiple hills in different parts of a Contour plot, and each hill passes through the same altitude contours, no more than one of each contour is labeled. Some altitude values are not labeled at all if the labels are too close to each other. For these reasons, there may be contours whose altitude values are not immediately apparent.

Contour plots are drawn in panels, like Surface plots. Each panel contains a set of straight lines that connect with other lines from neighboring panels at the edges. The graph's resolution determines the number of panels, as with Surface plots. The resolution value indicates the number of panels in each direction. Therefore, the default display of a Contour plot generated from a particular equation is not as smooth as a Line plot generated from a similar equation, even though the resolution values are the same.

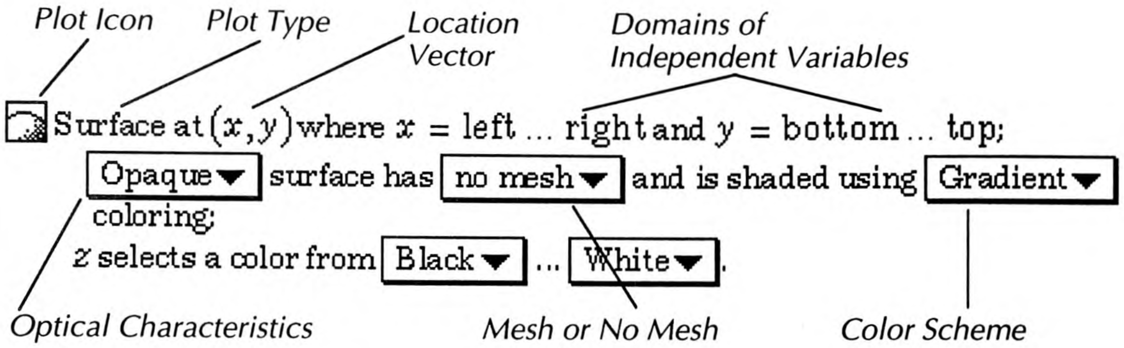
The following Contour plot has a resolution value of four (four panels on each side for a total of sixteen panels). A set of Grid Lines has been added to the graph delimiting each panel. Within each panel the lines of the plot are straight.



This shows you how a contour plot works. If you increase the resolution, the lines usually smooth out and look the way you would expect.

Density Plots

Density plots contain the following information:



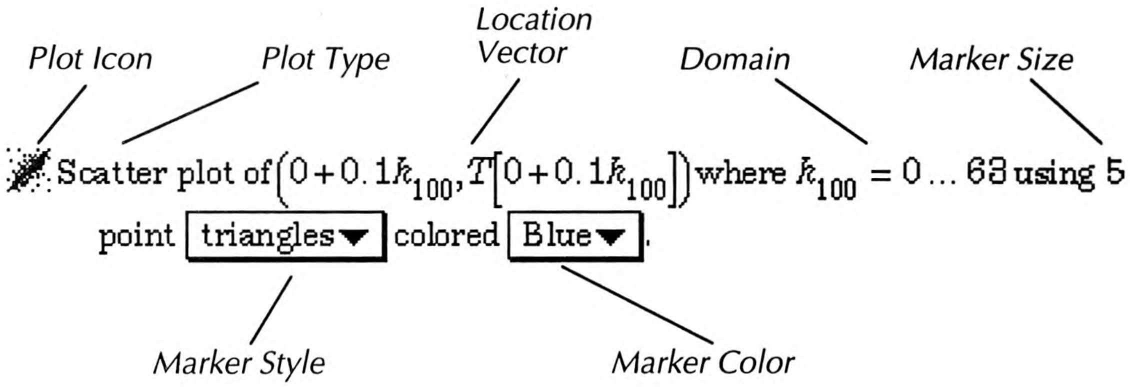
Surface at (x,y) where x = left ... right and y = bottom ... top;
Opaque▼ surface has no mesh▼ and is shaded using Gradient▼ coloring;
z selects a color from Black▼ ... White▼.

A two-dimensional density graph uses a Surface plot, which is normally displayed in three-dimensional graph theories. The plot is displayed with an opaque surface, with no mesh. The z values are displayed as a range of grays from black-to-white. All pop-up menus can be adjusted.

See the section on three-dimensional graphs, later in this section, for more details about Surface plots.

Scatter Plots

Scatter plots contain the following information:



Scatter plot of $(0+0.1k_{100}, T[0+0.1k_{100}])$ where $k_{100} = 0 \ldots 63$ using 5
point triangles▼ colored Blue▼.

When creating scatter plots Theorist generates arbitrary variables.

A scatter plot is a set of dots or markers that show individual data points. It is best for situations where you have to display individual data values instead of a continuous curve or surface.

There are 72 points per inch.

To change the style of the markers in the scatter plot, select the desired shape from the marker style pop-up menu in the graph details. The color of the markers can be changed in a similar manner. By typing in a number to replace the "5" on the second line of the scatter plot proposition, you can alter the size of the marker's diameter, in points. If you enter 0, the markers disappear. If you enter 1, the markers change to single dots, regardless of the marker style. If you enter any other expression, its numerical value is interpreted as the size. The expression can be a simple number or it can be an expression that depends upon the variable so that it changes size from one marker to the next.

Use matrices to make a scatter plot whenever the points do not come at regularly spaced intervals along the independent axis.

You can make other kinds of graphs by modifying the scatter plot directly. The range variable assumes integer values from the range expression; for each integer value, a marker is drawn. To draw a single marker you can delete the range expression and leave it as the ? character. The vector is the two-dimensional or three-dimensional location of the center of the marker. The vector can be any expression. This is useful for some animations.

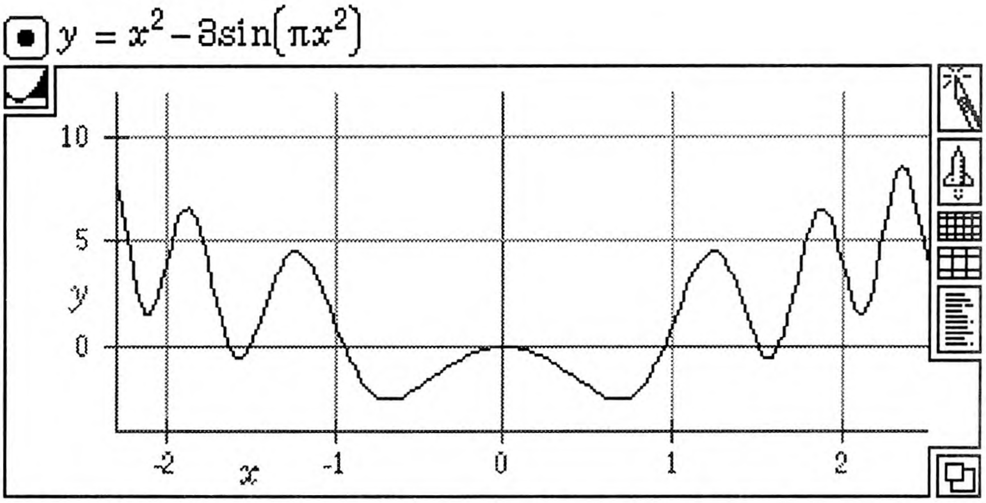
2D Graph Types

This section describes each type of graph that generates a two-dimensional graph theory. To create any one of these graphs, choose a command from one of the Graph menu's submenus. There are fourteen two-dimensional graphs: Parametric, Implicit, Density, and Contour, and line and scatter versions of Linear, SemiLog, SemiLog-X, Log-Log, and Polar.

Linear

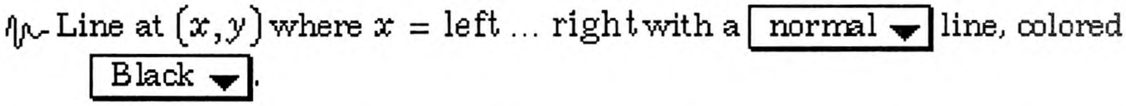
Alternately, you can select a table named in the form *y*(*x*).

To create a Linear graph, select an equation of the form $y = f(x)$ and choose **Graph▸y = f(x)▸Linear**:



● $y = x^2 - 3\sin\left(\pi x^2\right)$

The description of this plot is shown in the graph details:



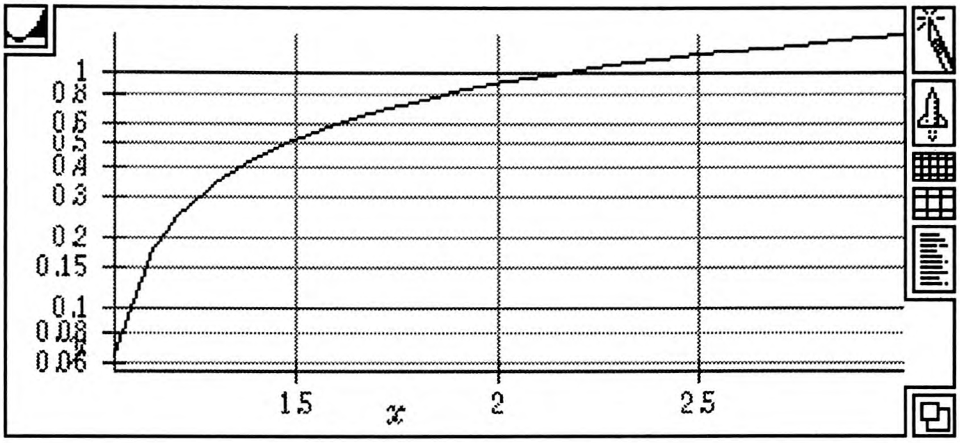
Line at (x, y) where x = left ... right with a normal ▾ line, colored Black ▾.

The bounds of this graph were edited to be –2.3 ... 2.5 = left ... right and –4 ... 12 = bottom ... top.

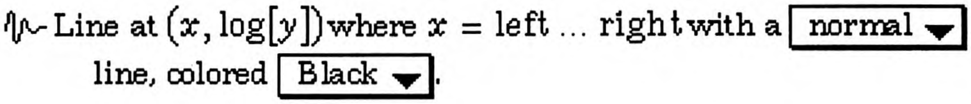
SemiLog
Alternately, you can select a table named in the form *y(x)*.

To create a SemiLog graph, select an equation of the form $y = f(x)$ and choose **Graph▸y = f(x)▸SemiLog**:

● $y = \log\left(x^3\right)$



The description of this plot is shown in the graph details:

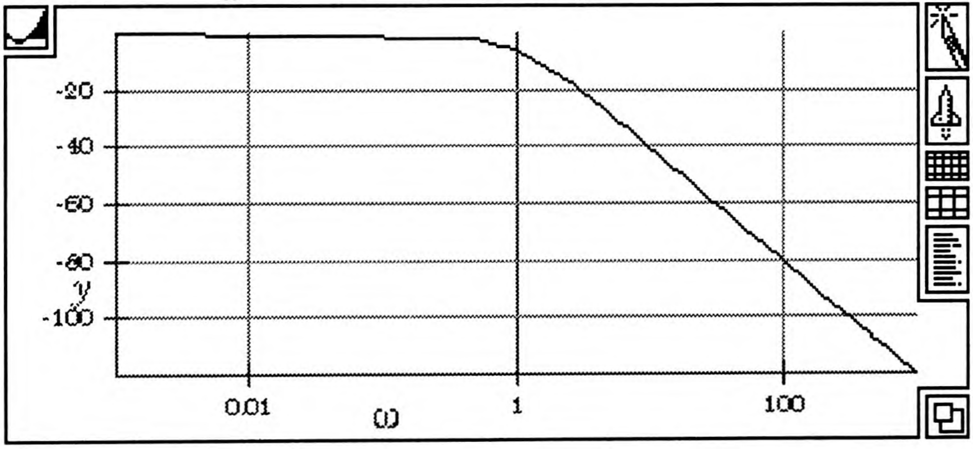


Line at $(x, \log[y])$ where x = left … right with a normal ▾ line, colored Black ▾.

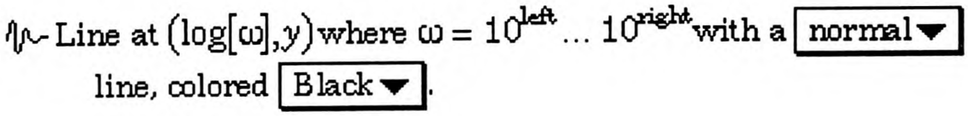
SemiLog-X
Alternately, you can select a table named in the form *y(x)*.

To create a SemiLog-X graph, select an equation of the form $y = f(x)$ and choose **Graph▸y = f(x)▸SemiLog-X**:

● $$y = 20\log\left(\left|\frac{1}{\left[1+2\zeta\left\{i\frac{\omega}{\omega_n}\right\}\right]+\left[i\frac{\omega}{\omega_n}\right]^2}\right|\right)$$



The description of this plot is shown in the graph details:

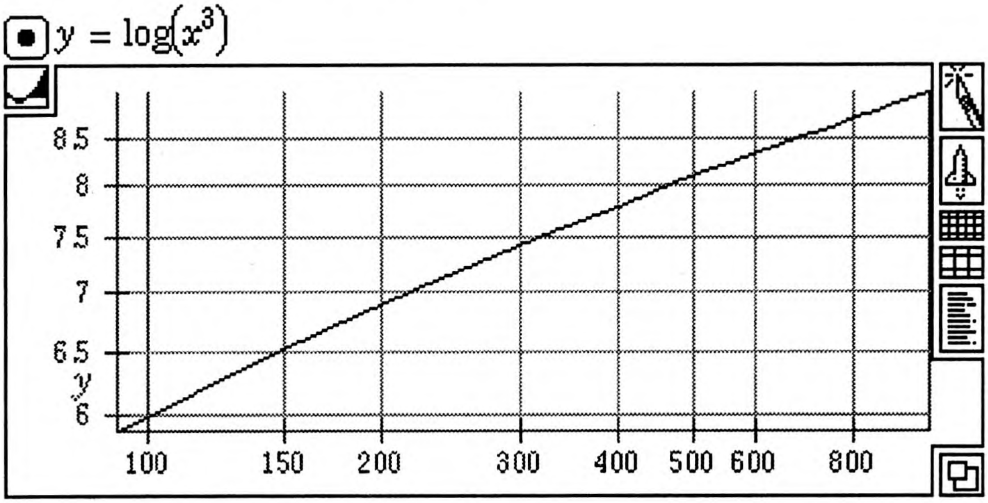


Line at $(\log[\omega], y)$ where $\omega = 10^{\text{left}} \ldots 10^{\text{right}}$ with a normal ▾ line, colored Black ▾.

Log-Log

Alternately, you can select a table named in the form *y(x)*.

To create a Log-Log graph, select an equation of the form $y = f(x)$ and choose **Graph▸y = f(x)▸Log-Log**:



● $y = \log\left(x^3\right)$

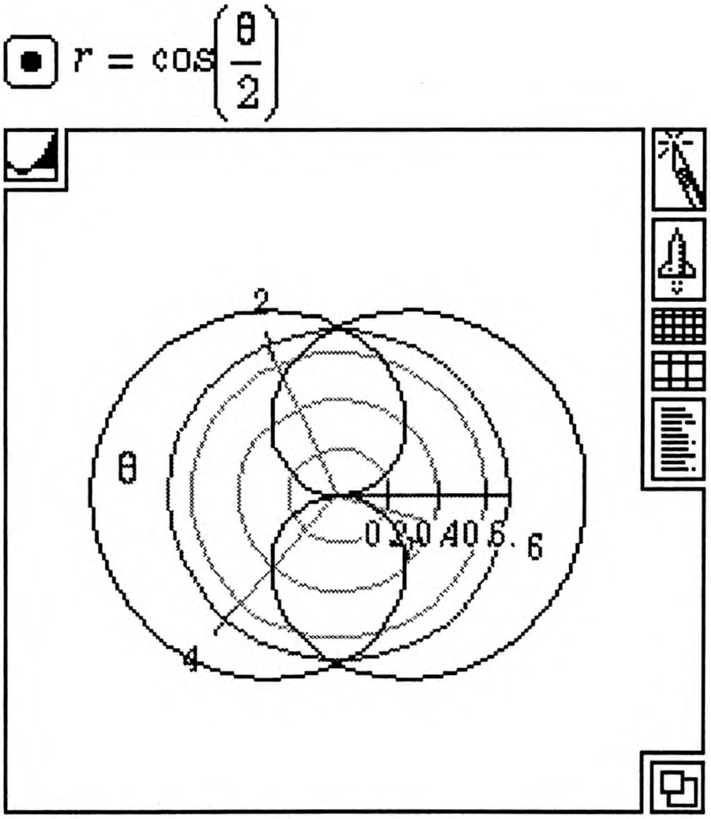
The description of this plot is shown in the graph details:

Line at $(\log[x], \log[y])$ where $x = 10^{\text{left}} \ldots 10^{\text{right}}$ with a normal▼ line, colored Black▼.

Polar

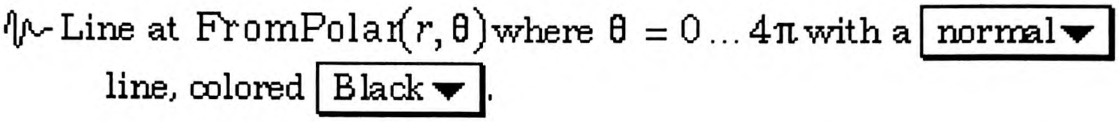
Alternately, you can select a table named in the form *y(x)*.

To create a Polar graph, select an equation of the form $y = f(x)$ and choose **Graph▸y = f(x)▸Polar**:



● $r = \cos\left(\frac{\theta}{2}\right)$

The description of this plot is shown in the graph details:



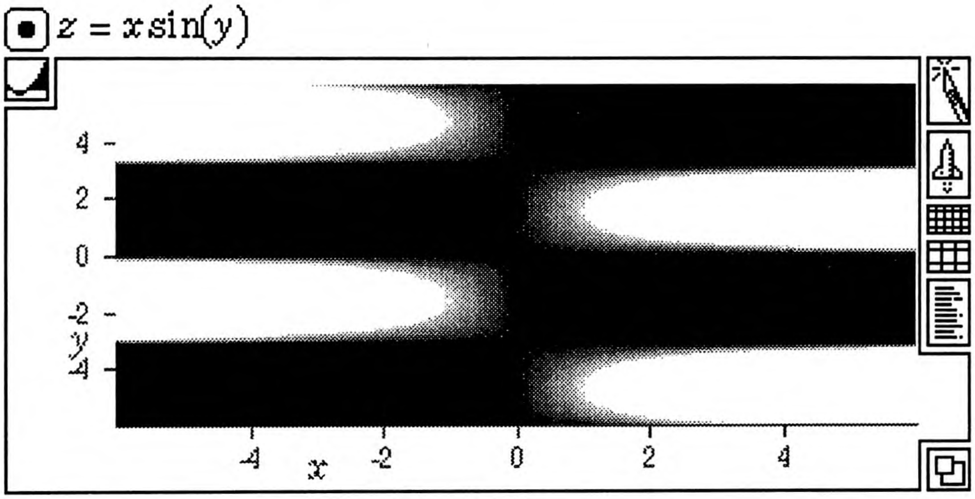
Line at FromPolar(r, θ) where θ = 0 ... 4π with a normal▼ line, colored Black▼.

For this graph, the domain of the independent variable (θ) was increased from the default of 0 ... 2π to 0 ... 4π.

Polar plots use the predefined function FromPolar to translate data-point values from polar to rectangular coordinates. The first time you create a polar graph, a dialog box asks you to confirm the inclusion of this function in the current notebook. The name declaration for FromPolar is placed in the Declarations comment at the top of the current notebook and is used by all subsequent Polar graphs. See the Expressions section for more information about predefined names and behaviors..

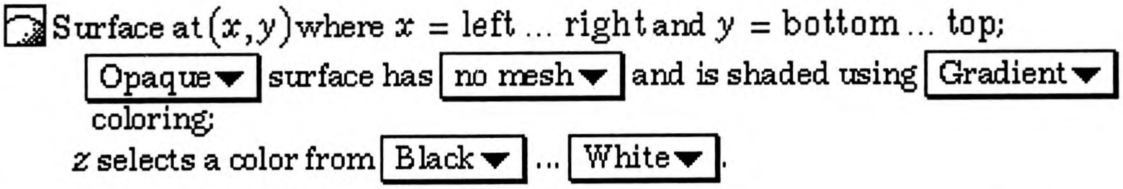
Density

To create a Density graph, select an equation of the form $z = f(x, y)$ and choose **Graph▸z = f(x, y)▸Density 2D**:



● $z = x\sin(y)$

The description of this plot is shown in the graph details:



Surface at (x, y) where x = left ... right and y = bottom ... top;
Opaque▼ surface has no mesh▼ and is shaded using Gradient▼ coloring;
z selects a color from Black▼ ... White▼.

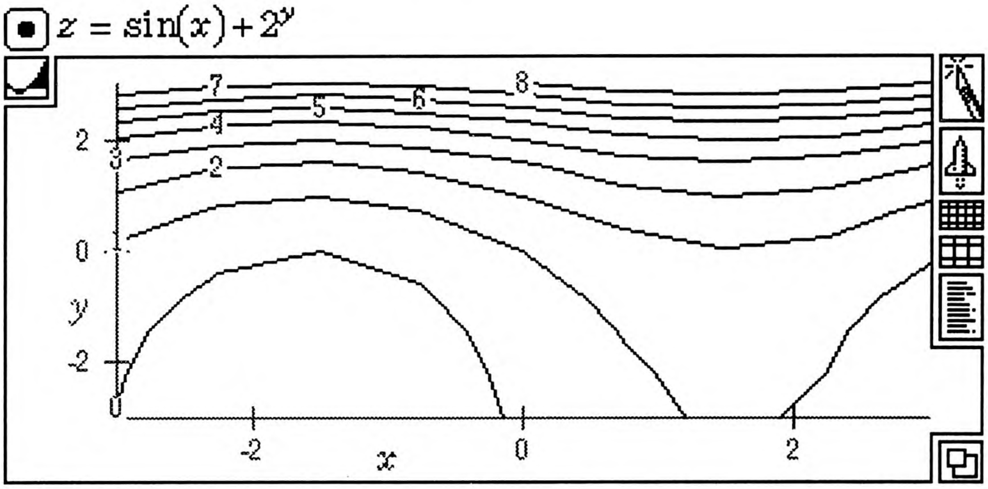
A density graph consists of a two-dimensional graph theory, a Surface plot, and a pair of Axes. Surface plots are usually displayed in three-dimensional graph theories but density plots are an exception. The plot is displayed with an opaque surface, with no mesh. The z values are displayed as a range of grays from black-to-white. All pop-up menus can be adjusted.

For this graph, the resolution was increased from the default of eight to 128 by clicking on the higher-resolution icon.

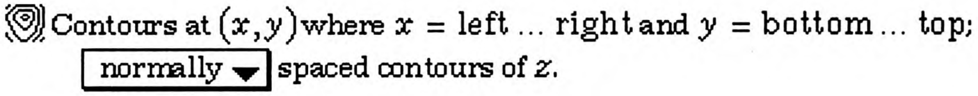
See the section on three-dimensional graphs, later in this section, for more details about Surface plots.

Contour

To create a Contour graph, select an equation of the form $z = f(x, y)$ and choose **Graph▸z = f(x, y)▸Contour 2D**:



The description of this plot is shown in the graph details:

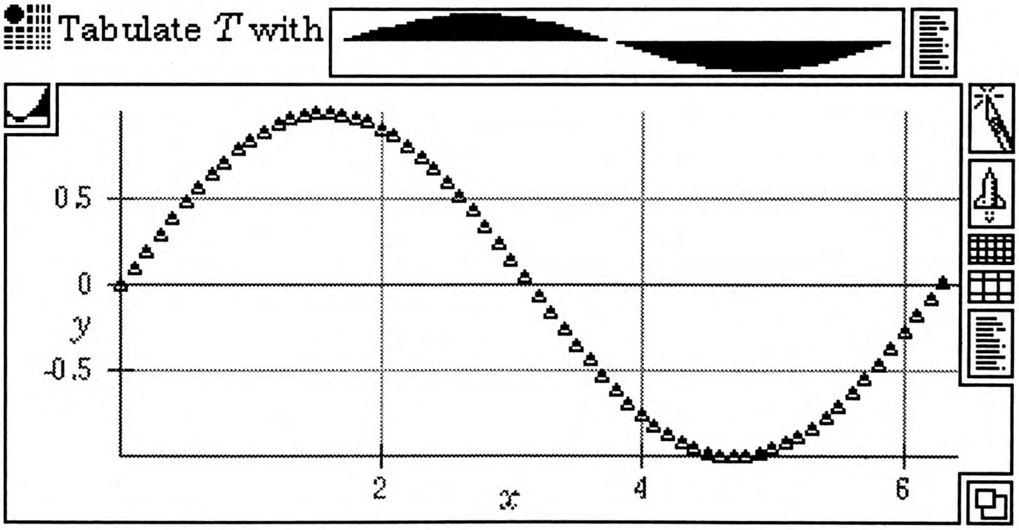

Contours at (x,y) where x = left ... right and y = bottom ... top;
normally ▾ spaced contours of z.

Contour plots display the relationship between three variables in two dimensions. The dependent variable (z) is displayed as contour lines of equal value as on a topographical map. Using the pop-up menu, contours can be placed at zero only, very sparsely, sparsely, normally (the default), densely, or very densely.

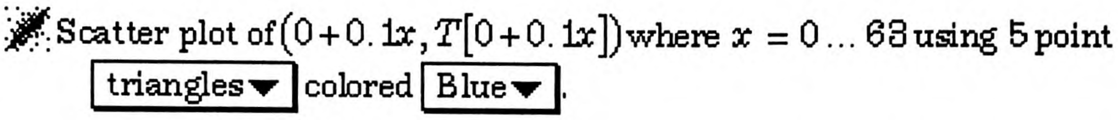
Scatter

Alternately, you can select a table named in the form *y(x)*.

To create a Scatter plot select a matrix or table and choose an item from the **Graph** menu's **Scatter** submenu.



The description of this plot is shown in the graph details:



Scatter plot of $(0+0.1x, T[0+0.1x])$ where $x = 0 \ldots 63$ using 5 point triangles▼ colored Blue▼.

The items on the Scatter submenu let you choose from different coordinate systems.

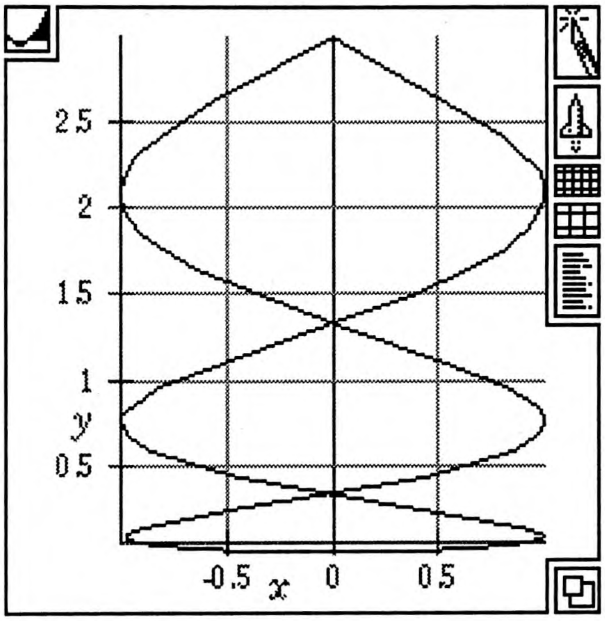
Parametric

Alternately, you can select a table in place of either or both equations; they should be named in the form *x(t)* and *y(t)* as appropriate.

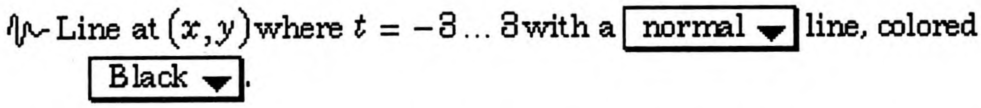
To create a Parametric graph, select two equations of the form $x = f(t)$, $y = g(t)$ and choose **Graph▸Other▸x = f(t), y = g(t) Parametric**.

● $x = \sin(\pi t)$

● $y = \dfrac{t^2}{3}$



The description of this plot is shown in the graph details:



Line at (x, y) where $t = -3 \ldots 3$ with a normal ▼ line, colored Black ▼.

In this graph, x and y are both dependent variables; t is the independent variable. No boundary values were explicitly indicated for t; defaults of –3 and 3 were used to generate the graph.

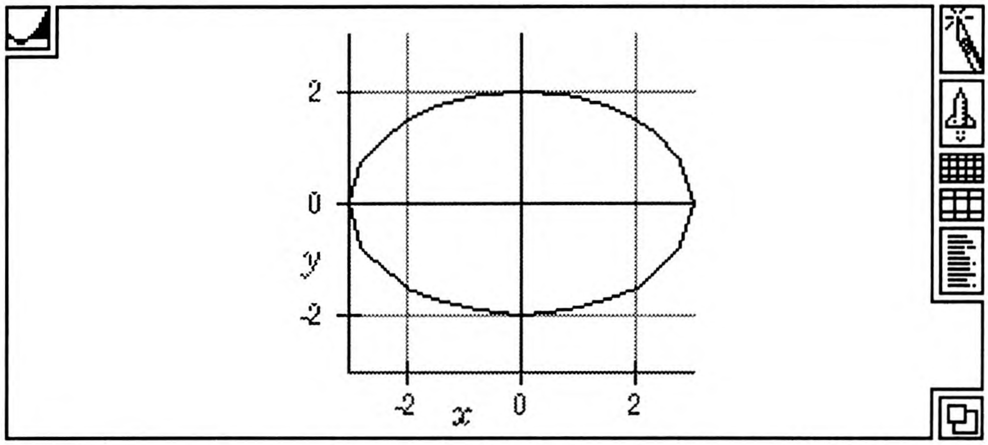
The dependent variable in the first selected equation becomes the first variable in the location vector and is displayed along the horizontal axis.

Parametric graphs are displayed after the second selected equation.

Implicit

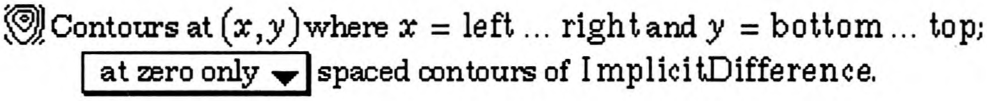
Implicit graphs are used for equations in two variables which can not be resolved into the standard explicit form. To create an Implicit graph, select an equation of the form f(*x*,*y*) = g(*x*,*y*) and choose **Graph▸Other▸f(x,y) = g(x,y) Implicit**:

□ $\frac{x^2}{9} + \frac{y^2}{4} = 1$



This graph's aspect ratio was changed to true proportions.

The description of this plot is shown in the graph details:



Contours at (x,y) where x = left ... right and y = bottom ... top;
at zero only ▾ spaced contours of ImplicitDifference.

Notice the graph is not as smooth as with a typical graph, such as a graph of sin(*x*). This is because implicit equations are much more difficult to calculate.

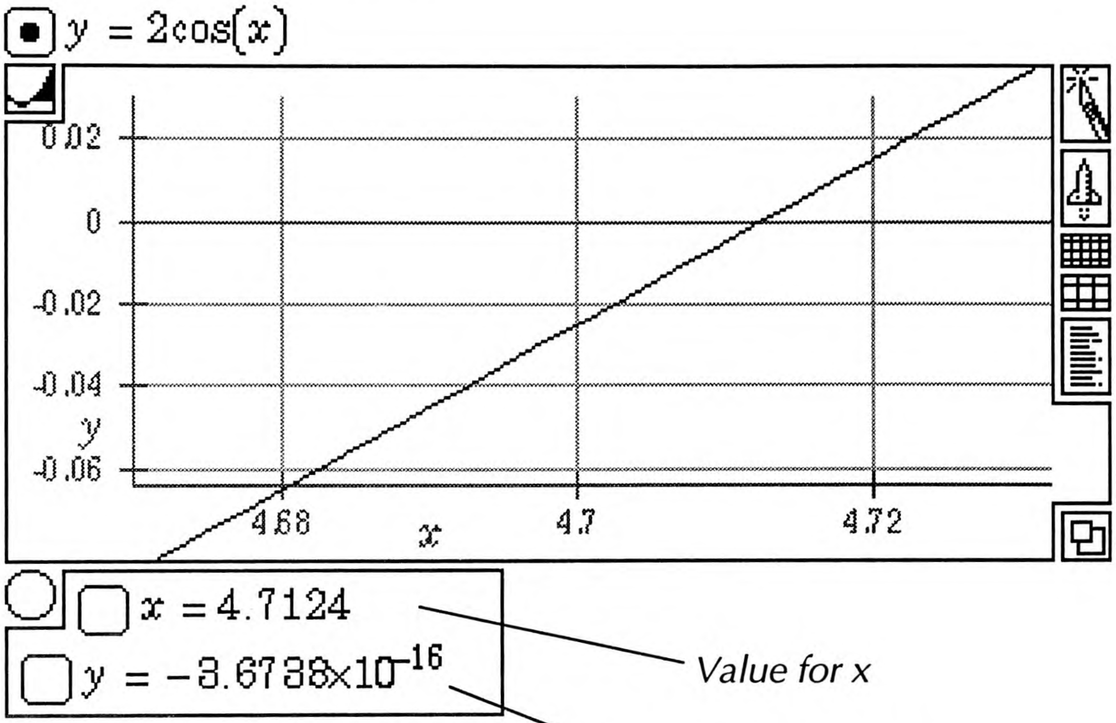
With explicit equations, Theorist chooses different values for *x* and calculates the value of *y* to draw a line. The result is an (*x*, *y*) pair guaranteed to be on the line exactly. It is a one-step operation where forty to seventy evaluations generally get as much smoothness as is necessary.

With implicit equations, Theorist chooses *pairs* of values (*x*, *y*). It evaluates both sides of the equation for each data point (*x*, *y*) to see how close to zero the result is. Rarely scoring a direct hit, Theorist then does linear interpolation to estimate where the line lies. The resulting line segments are merely rough sketches of where the equation is true. By default, eighty-one (*x*, *y*) pairs are evaluated, yielding the equivalent of nine data points vertically and nine horizontally. To get a resolution approaching that of typical line plots at normal resolution, you must increase the implicit graph to a much higher resolution, which is time-consuming to calculate.

Implicit graphs do not guess the domain and range as well as some of the other kinds of graphs. This may require you to manipulate the graph using the Rocketship or the Knife (or editing the bounds values in the graph details).

Finding Roots

To find the root of a Line plot displayed in a two-dimensional graph theory, zoom in (with the Rocketship or the Knife) on a portion of the graph that contains only one crossing of the horizontal (x) axis. Then select **Find Graph Root** from the **Manipulate** menu. A new case theory appears immediately following the graph theory displaying values for the two variables:



● $y = 2\cos(x)$

○ □ $x = 4.7124$

□ $y = -3.6738\times10^{-16}$

Value for x

Associated value for y (closest value to zero)

If there is more than one root (value for x) visible in the viewport, one is selected arbitrarily, or you receive an error message.

You can evaluate several roots by focusing on different portions of the graph in succession. You generate a new case theory each time you choose Find Graph Root.

Simultaneous Equations

You can also use Find Graph Root for simultaneous numerical equation solving. By creating two zero Contour plots, and finding a crossing of the two plots, you can solve two equations in two unknowns. For example, consider finding solutions for the two equations:

$$\frac{\log(y)}{x} = 2y\sin(x) \qquad\qquad e^x y = 2x^2$$

Use the Move Over manipulation on the first equation to create an equation that equals zero. Then rename the zero “zero”:

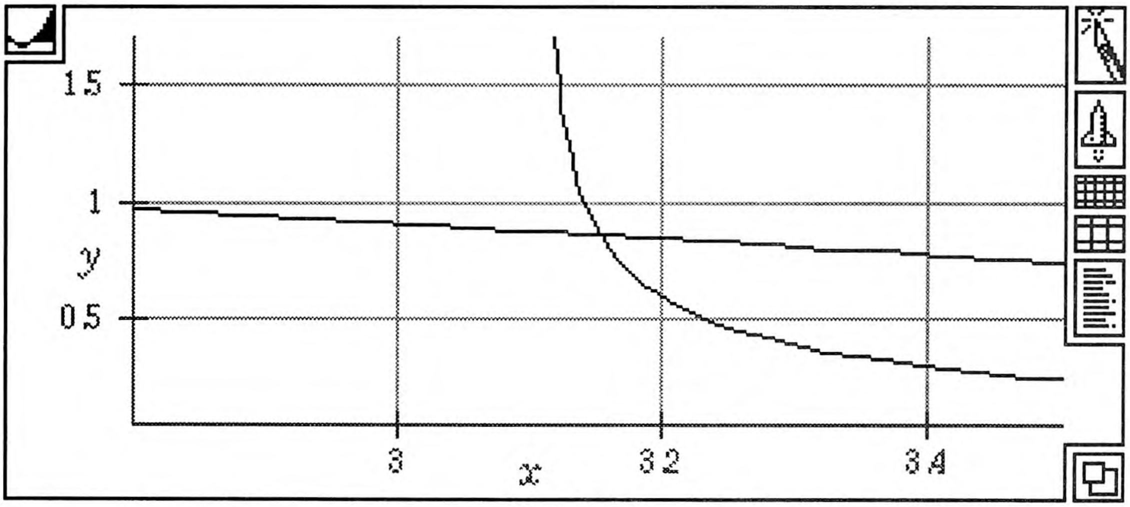
□ $$\frac{\log(y)}{x} = 2y\sin(x)$$

△ $$0 = -\frac{\log(y)}{x} + 2y\sin(x)$$ *MoveOver*

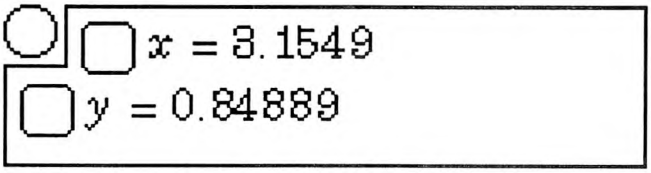
□ $$\mathrm{zero} = -\frac{\log(y)}{x} + 2y\sin(x)$$

Make zero user-defined.

Create an Implicit graph from the $e^x y = 2x^2$ equation. Declare x and y to be the graph variables. Then select the equation for zero and choose the **Add Contour Plot** item from the **Graph▸Additional** submenu. Match up the x and y graph variables and make zero the Contour variable. Open the graph details and adjust the second Contour plot’s pop-up menu to “at zero only” (like the first Contour plot). Then use the Rocket, the open hand pointer, and the Knife to close in on a point where the two plots cross.



When a single crossing point is in view, choose **Find Graph Root** from the **Manipulate** menu. A case theory is generated with values for x and y.


○ □ $x = 3.1549$

□ $y = 0.84889$

THREE-DIMENSIONAL GRAPHS

Three-dimensional graphs consist of a three-dimensional graph theory (the shell) and one or more two- or three-dimensional plots (Line, Scatter, Surface, or Contour) and associated Axes and Grid Lines. This section presents:

- A list of graphs which generate three-dimensional graph theories
- Rotating a three-dimensional graph for a better view
- Icons for editing three-dimensional graph theories
- Graph details of three-dimensional graph theories
- A description of the Surface plot proposition and its variations

Creating 3D Graphs

Choose one of the seven following types of graphs to generate a three-dimensional graph theory.

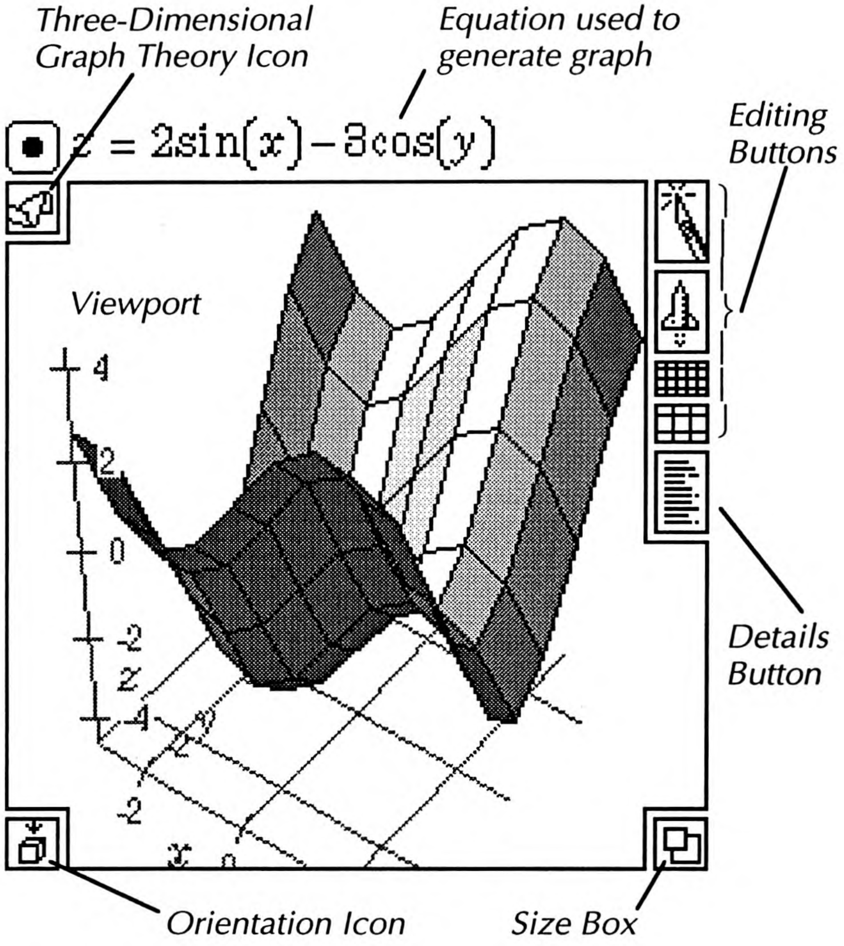
Graph Type	Submenu
Complex 3D	$y = f(x)$
Color 3D	$z = f(x, y)$
Illuminated 3D	$z = f(x, y)$
Spherical 3D	$z = f(x, y)$
Cylindrical 3D	$z = f(x, y)$
Space Curve	Other
Scatter 3D	Scatter

Complex 3D, Color 3D, Illuminated 3D, Spherical 3D, and Cylindrical 3D graphs all create Surface plots generated from the selected function. Space Curve creates a line plot. Scatter 3D creates a scatter plot.

You can specify the domain of the independent variables before creating a graph (e.g., $x > 0$, $x < 10$), or let the program generate default values. These values are displayed at the top of the details box and can be edited easily after creating a graph.

Overview

Most three-dimensional graph controls are similar to the two-dimensional graph controls. However, the open hand pointer works quite differently, and an extra icon, the "orientation icon," is displayed in the lower left-hand corner of the graph theory.


Three-Dimensional Graph Theory Icon
Equation used to generate graph
$z = 2\sin(x) - 3\cos(y)$
Editing Buttons
Viewport
4
2
0
-2
-4
z
y
2
-2
x
0
Details Button
Orientation Icon
Size Box

Rotating

To *rotate* (rather than slide) a three-dimensional graph, click in the viewport with the open hand pointer, and drag the image left, right, up, or down. The portion of three-dimensional space displayed in the viewport rotates as if you had your hand on a globe. As you drag, the graph either redraws smoothly with the new orientation, or a cube or rectangular box shows the bounds of the graph as you drag, and the graph redraws with the new orientation when you release the mouse. Whether Theorist smoothly redraws the graph or just draws an outline depends on the current resolution and the resolution setting in the graph preferences dialog. After a rotation, unlike clicking and dragging in a two-dimensional graph theory, the graph's boundary values do not change.

The outline box line thickness increases at each multiple of 15 degrees of rotation (in latitude and longitude). If you rotate the box to a position that is a multiple of 15 degrees longitude *and* 15 degrees latitude, the line thickness increases even more.

Rotational freedom is limited to rotating the display around its vertical axis, and rotating the vertical axis towards and away from you. Hold down the option key (Macintosh) or alt and Ctrl keys (Windows) to tilt the vertical axis sideways and rotate the graph more freely. In this mode, you can rotate the display in the line of sight by dragging in a wide circle. Do this in a direction opposite to the direction you want to rotate the model. (This technique works on the same principal as rolling a tennis ball on a table top with the palm of your hand. The direction of rotation is the opposite of the effected rotation.) Click on the Orientation icon to return the display to its original orientation.

Momentum occurs in the background when you have left Theorist to use other programs. You may want to stop a rotation by clicking on the graph before putting Theorist in the background.

If you click to grab a graph, move the hand, and release the mouse button while the hand is still, the graph redraws in the new orientation. However, if you click and grab a graph, move the hand, and release the mouse button before you have stopped moving the mouse, the graph is continually redrawn along the desired rotation path. The grab-and-stop method is like slowly moving a globe to position it at a specific location, whereas the grab-and-swipe method is like giving the globe a good swipe to watch it spin. The amount of momentum is determined by how short or long the "swipe" is—grab a corner and swipe to the opposite corner for lots of momentum or grab in the middle with a short swipe for a less radical rotation. (The grab-and-spin method may be more entertaining, the grab-and-stop method more enlightening.)

Rotational momentum described here occurs only when the graph resolution is set at or lower than the setting in the Graph Prefs. To change the graph bounds of a three-dimensional graph, use the Knife or Rocketship icons as you would for a two-dimensional graph, or edit the boundary values in the graph details.

Editing Icons

Each 3-D graph proposition is bordered by its own set of editing icons for changing various characteristics of the graph.

Knife

Knife Cursor

Click on the Knife icon to *pick up* the Knife (the cursor becomes a knife). Click and drag across a selection of surface panels to look at just that part of a plot. The portion of the graph you select can be most of the surface or just a tiny portion, but it must be at least two panels. It is easiest to use the knife if your view is of the top (or bottom) of a plot, or perpendicular to the portion of the surface you want to select. Using the Knife changes the values of the graph bounds.

Super Knife Cursor

As with 2-D graphs, you can double-click the Knife icon to activate the Super Knife, which stays "in your hand" for repeated knife-zooming until you click the Knife icon to make it go away. While active, Undo undoes *all* of your slices and returns the graph to its original view.

If, while slicing with either knife, you change your mind or make a mistake, you can abort the slice by dragging the knife back to its original "incision" panel and release the mouse button. (If only one panel is selected, Theorist assumes your very small slice means you want to abort the operation.)

In higher resolutions, the knife may select more than one panel at a time. That is, it selects in squares of four, sixteen, sixty-four, or more. For this reason, use a lower graph resolution to make an exact knife selection.

Rocket

Click on the Rocketship icon to *zoom out* (from the center of the viewport) by a factor of two. Hold down the option key (Macintosh) or alt and Ctrl keys (Windows) and click on the Rocketship to *zoom in* by a factor of two. After a moment, the graph redraws. To zoom in or out by more than a factor of two, click the icon repeatedly. To zoom out by a factor of 1000, click on the icon ten times. Zooming in and out is reflected in the values of the graph bounds.

Resolution Icons

Click on the dense grid to increase the data points used to create the plots in the graph. Click on the open grid to reduce the number of data points. The more data points used, the smoother the plot surface(s), but the greater the memory consumption, and the longer it takes to generate the image.

Note: higher resolution settings eat memory!

When you click on a resolution icon, the current value and the destination value appear briefly to the left of the icon (e.g., 8 -> 16). Resolution values are 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, and 1024. (The default is 8.) For Surface Plots and Contour Plots, the value indicates the number of panels on each side of the displayed plot. For Line Plots, Axes, and Grid Lines, the value indicates the approximate number of line segments used to create a 90 degree curve. Plot resolution is not reflected in the graph details. Therefore, with a typical three-dimensional surface plot, increasing resolution quadruples the number of panels and reducing resolution quarters the number of panels.

Details Box

Click on the details box to open or close the graph details. The graph details section contains information about all graphical elements in a graph theory, and how they are displayed in the graph viewport.

Size Box

Click and drag on the size box to change the shape of the viewport. Larger graphs require more memory (especially when animated), and take slightly longer to generate. Changing the size of the viewport is not reflected in the graph details.

Orientation Icon

Click on the orientation icon at any time to return to the original orientation and sizing of a display.

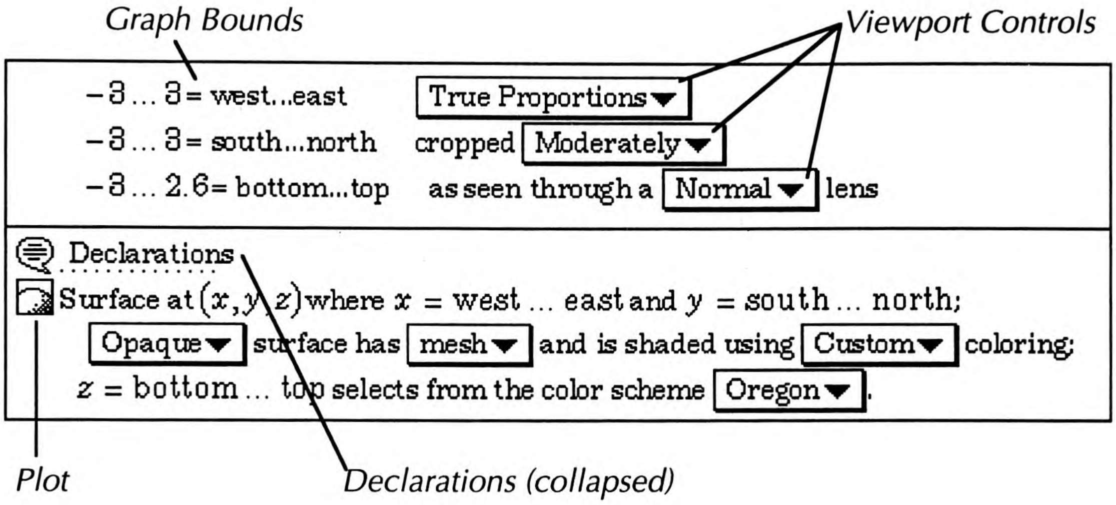
The orientation icon displays the side of the boundary volume shown in the viewport:

These appear different in smaller Notebook Font sizes.

front	*back*
right	*top*
left	*bottom*

Graph Details

The details box allows you to fine tune (and significantly alter) the appearance of a graph theory. The graph details section contains information about all graphical elements in a graph theory, and how they are displayed in the graph viewport:



−3 … 3 = west…east   True Proportions▾
−3 … 3 = south…north   cropped Moderately▾
−3 … 2.6 = bottom…top   as seen through a Normal▾ lens

Declarations

Surface at (x,y,z) where x = west … east and y = south … north;
Opaque▾ surface has mesh▾ and is shaded using Custom▾ coloring;
z = bottom … top selects from the color scheme Oregon▾.

Plots

Each plot (and every other graphical item) is displayed with an icon followed by a sentence describing the attributes of the plot. These sentences contain pop-up menus that you can use to change the plot's attributes.

Once created, plot propositions can be adjusted, edited, and moved around like any other proposition. To move a graph element, select a plot and Cut (or Copy) and Paste it into another graph theory. Plot propositions are meaningless outside graph theories.

You can move plots among three-dimensional graph theories without difficulty. However, it is often easier to select an equation and choose the appropriate menu item from the Additional submenu to add Line, Scatter, or Contour plots, Axes or Grid Lines, or even additional Surface plots.

To add a plot (or axis or set of grid lines) to a graph theory, select an equation (and a graph theory if there is more than one), and choose an item from the **Graph▸Additional** submenu.

Graph Bounds

The graph bounds determine the logical height, width, and depth of the graphing area. Edit the boundary values as you would any other expression. Set the values to different numbers or to names or expressions that evaluate to numerical values. Using the Rocketship or Knife icon adjusts the boundary values. Rotating a three-dimensional graph does not change the boundary values and is not reflected in the graph details.

Viewport Controls

To the right of the boundary values are three pop-up menus. The first determines the *aspect ratio,* the second the degree to which the display is *cropped*, and the third the type of *lens* used to view the graph.

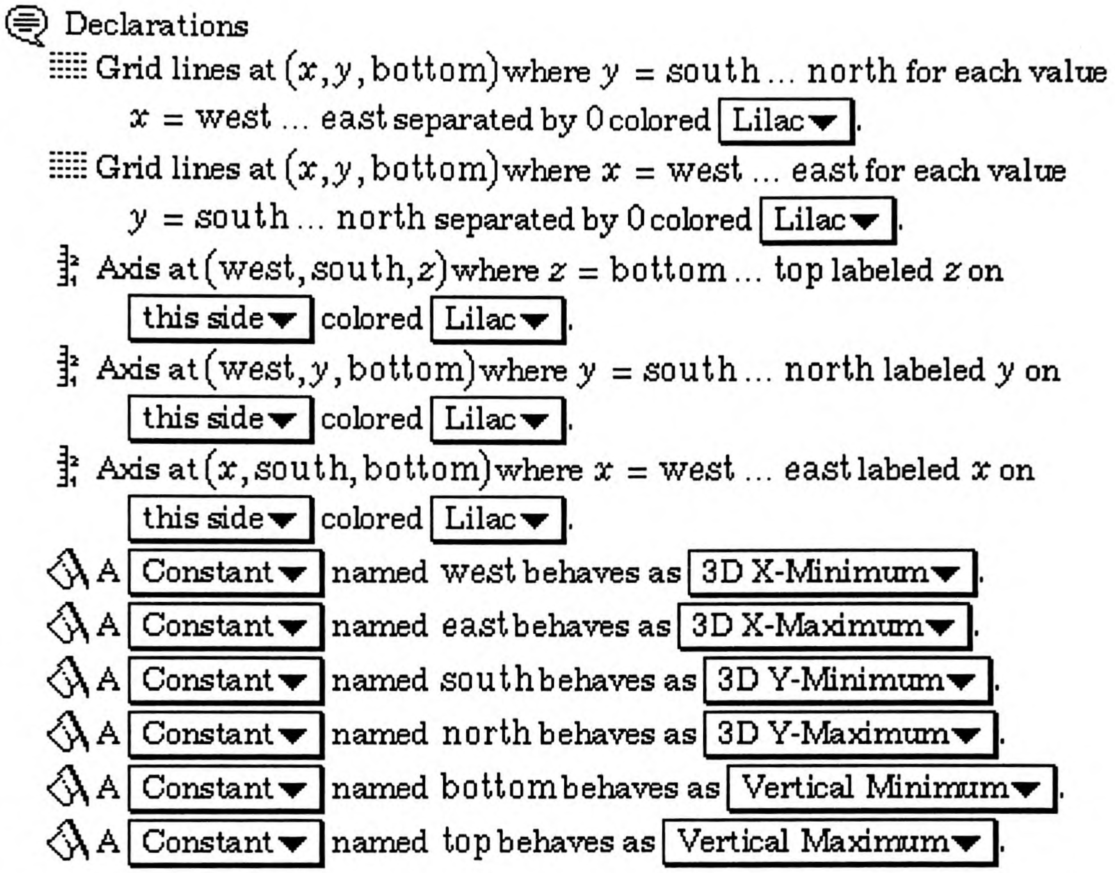
The aspect ratio is either "True Proportions" or "Stretch to Fit." True Proportions indicates that all three axes display the same unit measure. This would not work well if you were plotting different units on the different axes. For example, if you drew a graph of a person's height (in millimeters) versus her age (in years), it would be considerably taller than wide and the plot would be hard to see. Stretch to Fit adjusts the scale of the axes to fit an exact cube. This does not work well if the three dimensions all have the same units (e.g., centimeters) or if a graph uses spherical coordinates. For example, a sphere might end up flattened like a pancake or narrowed into a rod, unless all three sets of bounds are separated by the same distance. The default display is Stretch to Fit except for Spherical 3D, Cylindrical 3D, and Complex 3D graphs, which are always created using True Proportions.

To adjust how tightly a graph is displayed, choose an item from the cropping pop-up menu. A tightly cropped view displays a close up of a three-dimensional graph which fills the viewport and "crops" off parts which are not in view. A loosely cropped view shows the entire graph, all axis labels, and a great deal of white space. The default is Moderately cropped (a small portion of the graph is usually outside the viewport). The possible settings are: Very Tightly, Tightly, Moderately Tightly, Moderately, Moderately Wide, Wide, and Very Wide.

The lens setting specifies the amount of foreshortening or perspective. A Wide Angle lens exaggerates the three-dimensional effect. The Telephoto setting reduces this effect, and Infinitely Distant creates an isometric projection (no foreshortening). The possible settings are: Very Wide Angle, Wide Angle, Slight Wide Angle, Normal, Slight Telephoto, Telephoto, and Infinitely Distant. The wider angle lenses usually work best with wider cropping.

Declarations

"Declarations" is a comment proposition (collapsed by default) containing plots and definitions used to construct the graph. These include the constants bottom, top, north, south, east, and west (and sometimes others). These names define the bounds of the graphing area in the viewport. Most plots use some or all of these names. The following exposed Declarations has names created by default for a Color 3D graph:



Declarations

Grid lines at (x, y, bottom) where y = south ... north for each value x = west ... east separated by 0 colored Lilac ▼.

Grid lines at (x, y, bottom) where x = west ... east for each value y = south ... north separated by 0 colored Lilac ▼.

Axis at $(\text{west}, \text{south}, z)$ where z = bottom ... top labeled z on this side ▼ colored Lilac ▼.

Axis at $(\text{west}, y, \text{bottom})$ where y = south ... north labeled y on this side ▼ colored Lilac ▼.

Axis at $(x, \text{south}, \text{bottom})$ where x = west ... east labeled x on this side ▼ colored Lilac ▼.

A Constant ▼ named west behaves as 3D X-Minimum ▼.

A Constant ▼ named east behaves as 3D X-Maximum ▼.

A Constant ▼ named south behaves as 3D Y-Minimum ▼.

A Constant ▼ named north behaves as 3D Y-Maximum ▼.

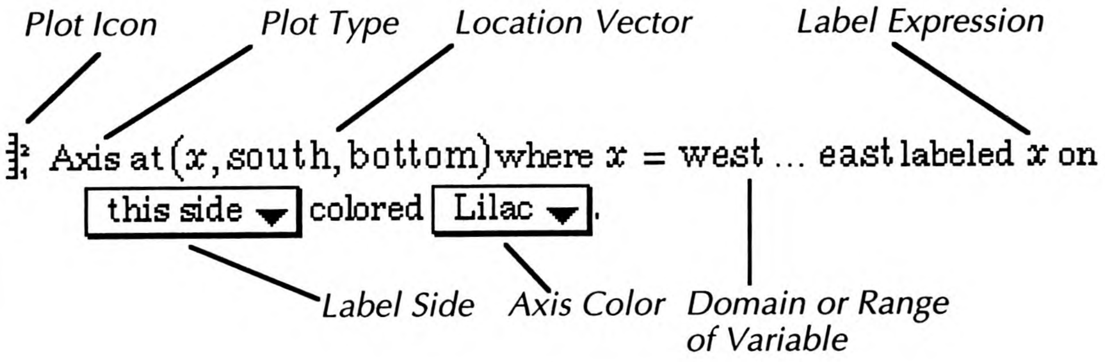
A Constant ▼ named bottom behaves as Vertical Minimum ▼.

A Constant ▼ named top behaves as Vertical Maximum ▼.

As you can see, the exposed Declarations also show Grid and Axis plot propositions, explained below.

Axes

Each Axis is displayed with an icon and a sentence describing its placement and labeling. Axes are simple plot propositions that use certain constants to generate a frame of reference for the section of three space shown in the viewport. They calculate the location of a series of points and generate a line connecting those points. This set of points is defined by a location vector. By default, each three-dimensional graph contains three axes.



Axis at $(x, \text{south}, \text{bottom})$ where x = west ... east labeled x on this side ▼ colored Lilac ▼.

Axes for three-dimensional graphs are generated the same way as two-dimensional axes. Two constants are grouped with one of the equation variables in a location vector to generate a series of points that form a straight line. Spherical graphs generate two curved axes and one straight axis, whereas Cylindrical graphs generate one curved axis and two straight ones.

Axes are generated with a set of tick marks that display a set of values of the equation variable. This expression is displayed on one side or the other of the axis between the first and second tick marks.

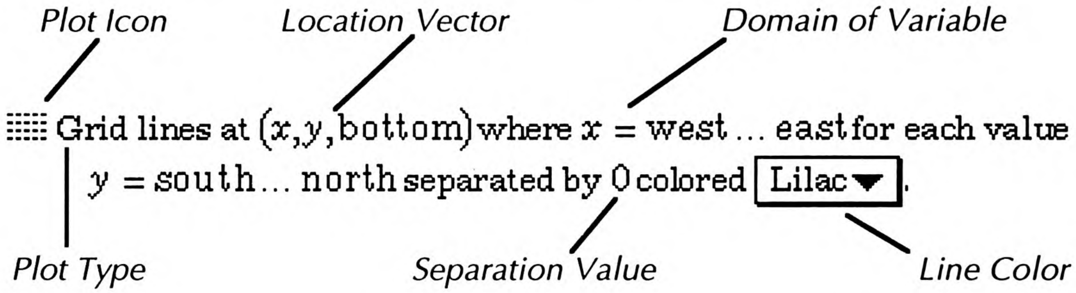
You can move Axes from one edge of the graphing area to another (by changing the location vector), add Axes to the other edges of the graphing area, and even change the location vector to be dependent on an equation.

As you zoom in and out with a graph (using the Rocketship or Knife or by typing new bounds), the axes and grids follow to demarcate the space being viewed. You can change this behavior by changing these expressions.

Axis propositions can be copied, moved, and edited like other propositions.

Grid Lines

Each set of (usually parallel) grid lines is displayed with an icon and a sentence describing its placement. Grid Lines are also simple plot propositions. By default, all three-dimensional graphs, other than spherical and cylindrical graphs, contain two Grid Line plot propositions. Spherical and Cylindrical graphs do not generate Grid Lines.



Grid lines at (x,y,bottom) where x = west … east for each value
y = south … north separated by 0 colored Lilac▼.

Grid Lines for rectangular three-dimensional graphs are displayed at the bottom of the segment of three space displayed in the viewport. As with axis plot propositions, Grid Lines can be moved any where in this space, added, or deleted.

Grid Lines utilize a location vector to determine where the lines are drawn. By default, the graph bounds are used to determine the location of the grid lines. The above example makes a set of grid lines parallel to the x axis, but each at a different y value.

You can also set the range of x values to any expression that evaluates to a range of two real numbers. For example, if you set the range for the first of these plots to x =west +1 … east –1, the proposition generates a set of lines parallel to the x axis that start one unit in from each side.

The spacing value determines the distance between grid lines. If the spacing value is zero (the default), one grid line is drawn to match each tick mark on the axis (if you are using default Axes). To draw a grid line at every unit, change this value to 1. For lines every five units, set the spacing value to 5.

On color monitors, grid lines appear as light purple. On black and white monitors (and when printed on PostScript printers), grid lines are drawn as thin and/or dotted lines. Grid lines for major increments are thicker or darker than other grid lines. Major increments are at zero and at other large, round numbers. Zero is always a major increment.

Grid propositions can be copied, moved, and edited like other propositions.

Plot Types

Theorist can put any plot in a three-dimensional graph. Complex 3D, Color 3D, Illuminated 3D, Spherical 3D, and Cylindrical 3D generate Surface plots. Space Curve graphs generate line plots. Scatter 3D graphs generate scatter plots. Though there is no command to generate a three-dimensional Contour graph, you can add a Contour plot to a three-dimensional graph.

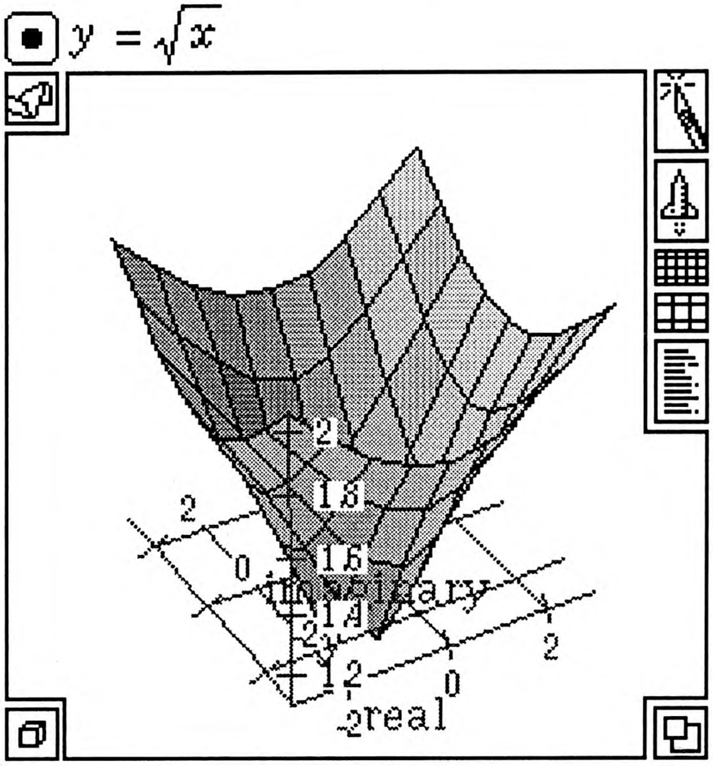
These plots work as described previously in the Two-Dimensional Graphing section, the main difference being the three-element location vector. Because of their significance in 3-D graphs, Surface plots are described in detail in the following Surface Plot Propositions section.

3D Graph Types

Each graph initially generates a specific variant of Surface plot as described in the following sections.

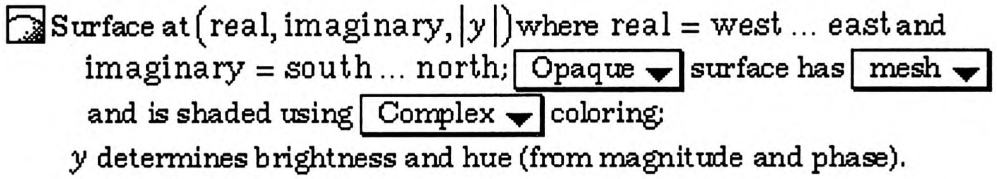
Complex 3D

To create a Complex 3D graph, select an equation of the form *y* = *f*(*x*) and choose **Graph▸y = f(x)▸Complex 3D**:



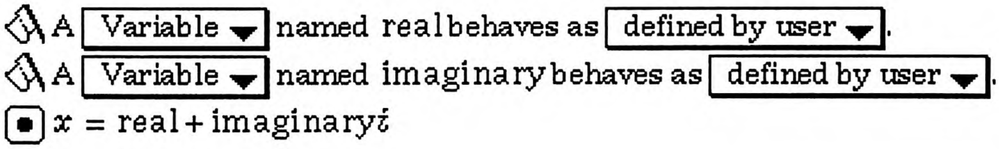
● $y = \sqrt{x}$

The default description of this plot is shown in the graph details:



Surface at (real, imaginary, $|y|$) where real = west ... east and imaginary = south ... north; Opaque ▾ surface has mesh ▾ and is shaded using Complex ▾ coloring; y determines brightness and hue (from magnitude and phase).

Complex 3D graphs generate a Surface plot specifically designed to display complex numbers, using the color scheme Complex. The location vector consists of the two variables real and imaginary, and the absolute value of the dependent variable, *y*. A Complex 3D graph splits the independent variable into its real and imaginary components and treats each of these as independent variables. The two variables, real and imaginary, are defined for this graph only and contained in the graph details under the Declarations proposition:



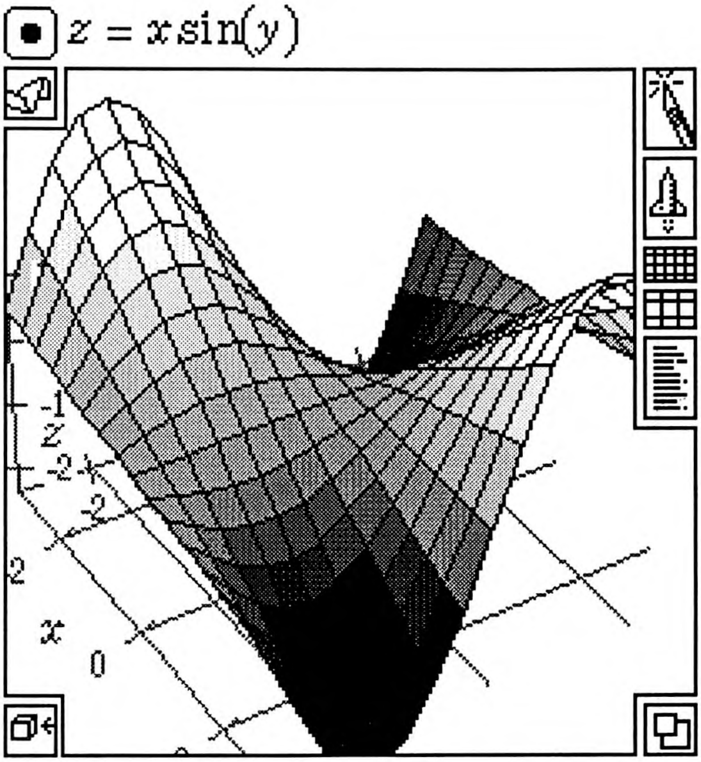
A Variable ▾ named real behaves as defined by user ▾.

A Variable ▾ named imaginary behaves as defined by user ▾.

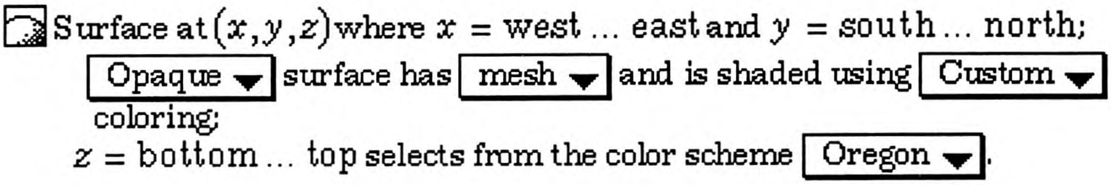
● x = real + imaginary$i$

Color 3D

To create a Color 3D graph, select an equation of the form $z = f(x, y)$ and choose **Graph▸z = f(x, y)▸Color 3D**:



The default description of this plot is shown in the graph details:



Surface at (x,y,z) where x = west ... east and y = south ... north;
Opaque ▾ surface has mesh ▾ and is shaded using Custom ▾ coloring;
z = bottom ... top selects from the color scheme Oregon ▾.

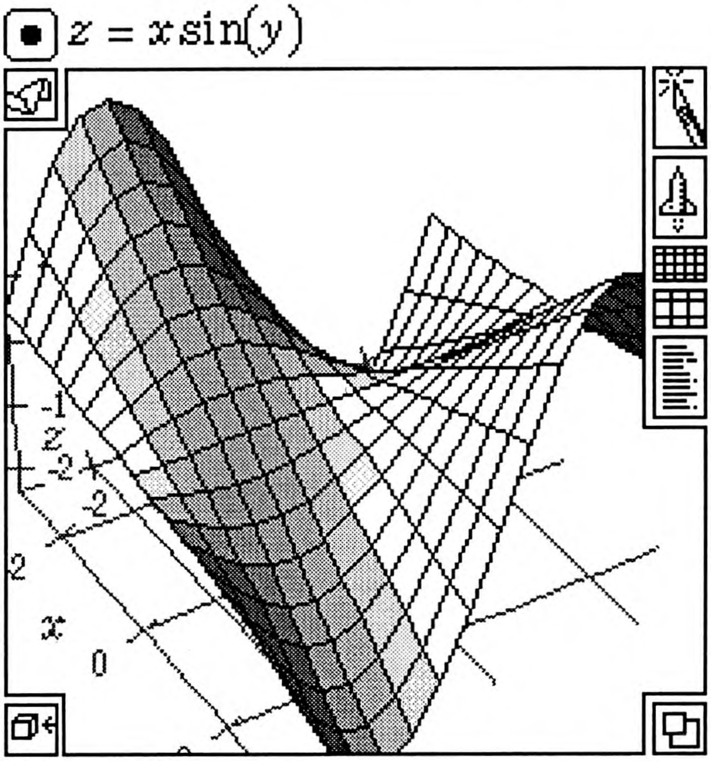
Color 3D graphs create Surface plots with Opaque optical characteristics, and use the Custom color interpretation.

Custom coloring is described in detail in the following Surface Plot Propositions section.

For this graph, the resolution was increased from the default of eight to 16, so there are seventeen vertices from west to east and from south to north.

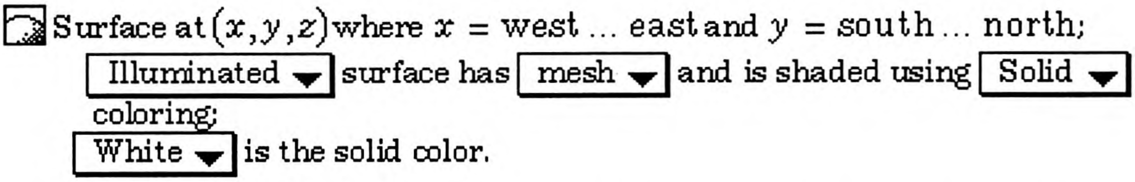
Illuminated 3D

To create an Illuminated 3D graph, select an equation of the form $z = f(x, y)$ and choose **Graph▸z = f(x, y)▸Illum. 3D**:



● $z = x\sin(y)$

The default description of this plot is shown in the graph details:



Surface at (x,y,z) where x = west ... east and y = south ... north;
Illuminated ▾ surface has mesh ▾ and is shaded using Solid ▾ coloring;
White ▾ is the solid color.

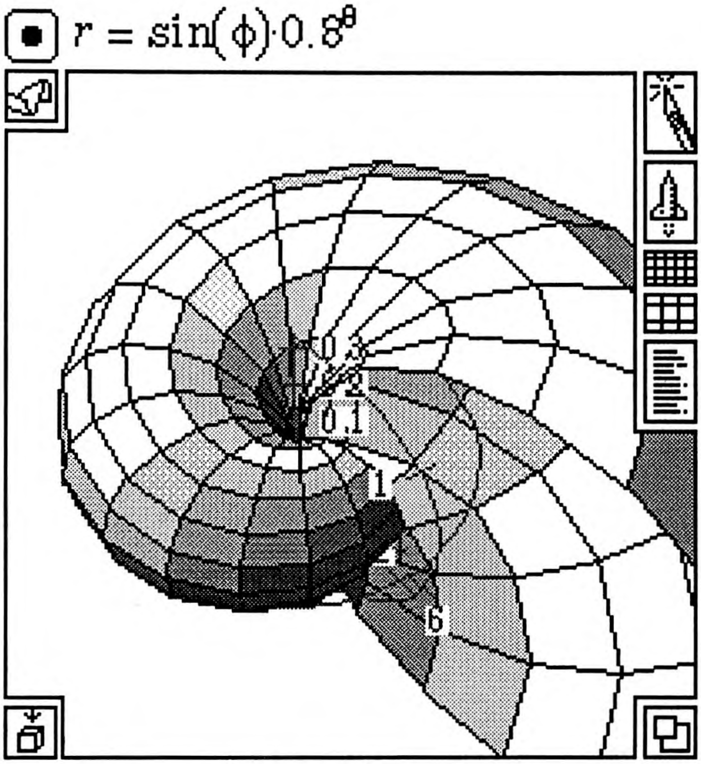
Illuminated 3D graphs generate a Surface plot with Illuminated optical characteristics and uses the Solid color scheme (a single color is used for all panels). White is the default color. The surface is displayed with a mesh. The illumination for an Illuminated Surface plot comes from over your left shoulder.

Solid coloring is described in detail in the following Surface Plot Propositions section.

This graph's resolution was increased from eight to sixteen.

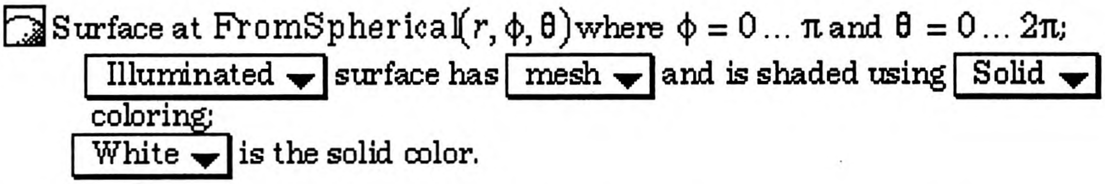
Spherical 3D

To create a Spherical 3D graph, select an equation of the form $z = f(x, y)$ and choose **Graph▸z = f(x, y)▸Spherical 3D**:



● $r = \sin(\phi)\cdot 0.8^{\theta}$

The default description of this plot is shown in the graph details:



Surface at FromSpherical(r, ϕ, θ) where $\phi = 0 \ldots \pi$ and $\theta = 0 \ldots 2\pi$;
Illuminated ▾ surface has mesh ▾ and is shaded using Solid ▾ coloring;
White ▾ is the solid color.

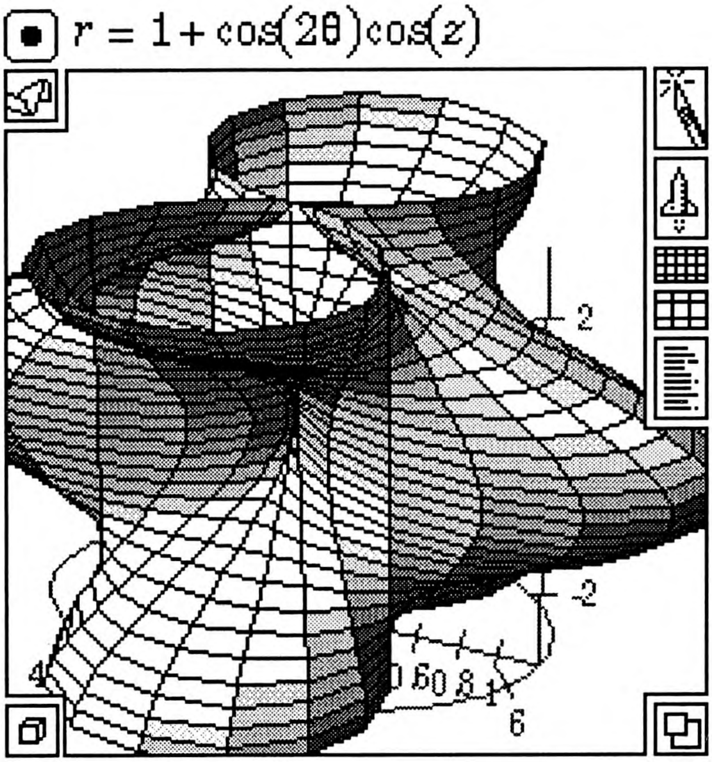
Spherical graphs generate a Surface plot with Illuminated optical characteristics similar to an Illuminated 3D graph except that the data points generated from the location vector are mapped through the function FromSpherical to create a spherical coordinate system. All characteristics can adjusted as in the Surface plot created for an Illuminated 3D graph.

For this graph, the domain for the third element in the location vector, θ, is from zero to 2π. The surface stops after completing one rotation "around the equator." To extend the surface deeper into the shell, increase this range (e.g., 0 ... 3π).

This graph's resolution was increased from eight to sixteen.

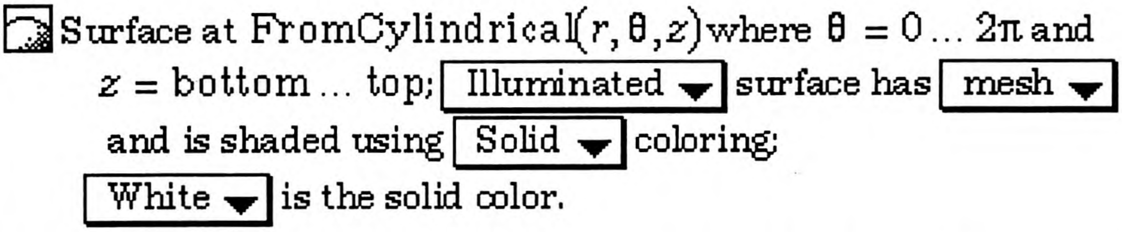
Cylindrical 3D

To create a Cylindrical graph, select an equation of the form $z = f(x, y)$ and choose **Graph▸z = f(x, y)▸Cylindrical 3D**:



● $r = 1 + \cos(2\theta)\cos(z)$

The default description of this plot is shown in the graph details:



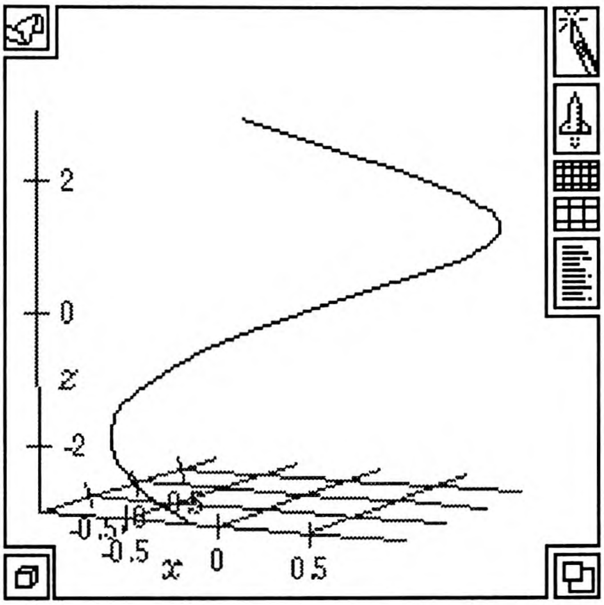
Surface at FromCylindrical(r, θ, z) where $\theta = 0 \ldots 2\pi$ and z = bottom … top; Illuminated ▾ surface has mesh ▾ and is shaded using Solid ▾ coloring; White ▾ is the solid color.

Cylindrical graphs generate a Surface plot with Illuminated optical characteristics similar to an Illuminated 3D graph except that the data points are mapped through the function FromCylindrical to create a cylindrical coordinate system. All characteristics can be adjusted as on an Illuminated 3D plot.

For this graph, the resolution was increased from the default of eight to thirty-two.

Space Curve

To create a Cylindrical graph, select three equations of the forms $x = f(t)$, $y = g(t)$, and $z = h(t)$ then choose **Graph▸Other▸x = f(t), y = g(t), z = h(t) Space Curve**:



The default description of this plot is shown in the graph details:

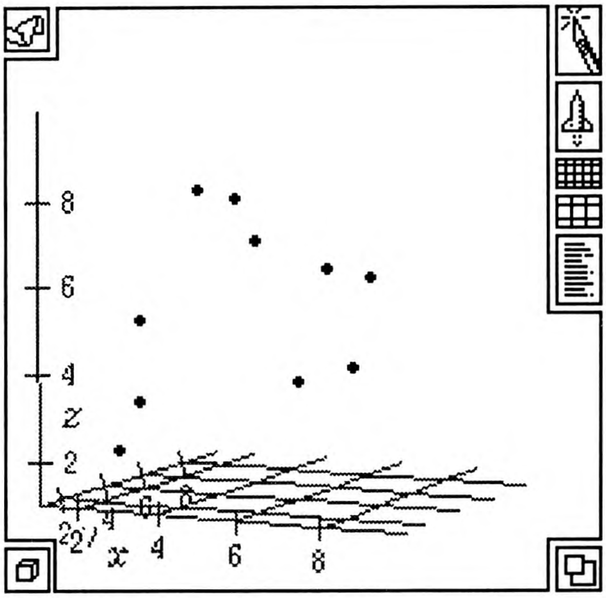
Line at (x, y, z) where $t = -3 \ldots 3$ with a normal▼ line, colored Black▼.

Space Curve graphs simply generate a line plot in three-dimensional space. Unlike 2D graph line plots, the Knife does not work on Space Curve graphs.

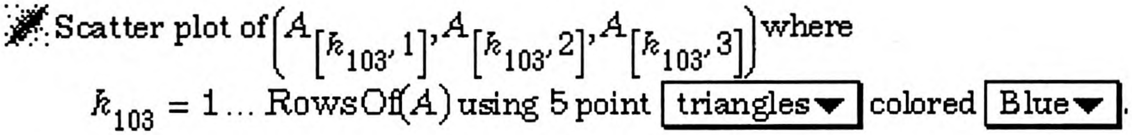
For this graph, the aspect ratio was changed to Stretch to Fit.

Scatter 3D

To create a Scatter 3D graph, select a three-column matrix and choose **Graph▸Scatter▸3D**:



The default description of this plot is shown in the graph details:



Scatter plot of $\left(A_{[k_{103},1]}, A_{[k_{103},2]}, A_{[k_{103},3]}\right)$ where $k_{103} = 1 \ldots \mathrm{RowsOf}(A)$ using 5 point triangles▾ colored Blue▾.

Scatter 3D graphs generate a scatter plot in 3D space. Unlike 2D graph scatter plots, the Knife does not work on Scatter 3D graphs and the accuracy buttons have no useful effect.

For this graph, the marker style was changed to spots.

Surface Plot Propositions

Surface plots are naturally more complex than other kinds of plots. These propositions contain several pop-up menus and values that can be easily adjusted to change many of the surface's characteristics. This section describes these characteristics and color schemes in detail.

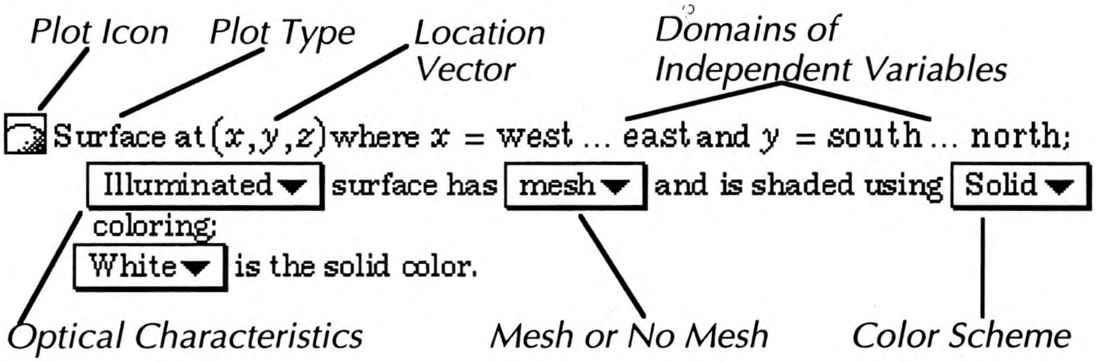
Surface Plot Graphs

Surface plots are created for Density, Complex 3D, Color 3D, Illuminated 3D, Spherical 3D, and Cylindrical 3D graphs. Each of these graphs generate a particular type of Surface plot. By default, these graphs generate the following plots.

Graph Type	Optical Characteristics	Mesh	Color Scheme	Color Setting
Color 3D	Opaque	Mesh	Custom	Oregon
Illuminated 3D	Illuminated	Mesh	Solid	White
Spherical 3D	Illuminated	Mesh	Solid	White
Cylindrical 3D	Illuminated	Mesh	Solid	White
Density	Opaque	No mesh	Gradient	Black to white
Complex 3D	Opaque	Mesh	Complex	*N/A*

Proposition

Surface plot propositions display the following information:



Surface at (x,y,z) where x = west ... east and y = south ... north; Illuminated▾ surface has mesh▾ and is shaded using Solid▾ coloring; White▾ is the solid color.

Location Vector and Domains

A Surface plot proposition contains the location vector and the domains of the independent variables. These are used to generate the vertices of the surface panels. The first value in the location vector determines the west ... east location for each vertex and the second value determines the south ... north location. Value ranges for the domains are set in the first lines of the graph details (the graph bounds). Values for the third element in the location vector (used to establish the bottom ... top location of each vertex) are generated by plugging a series of values for the independent variables into the original equation.

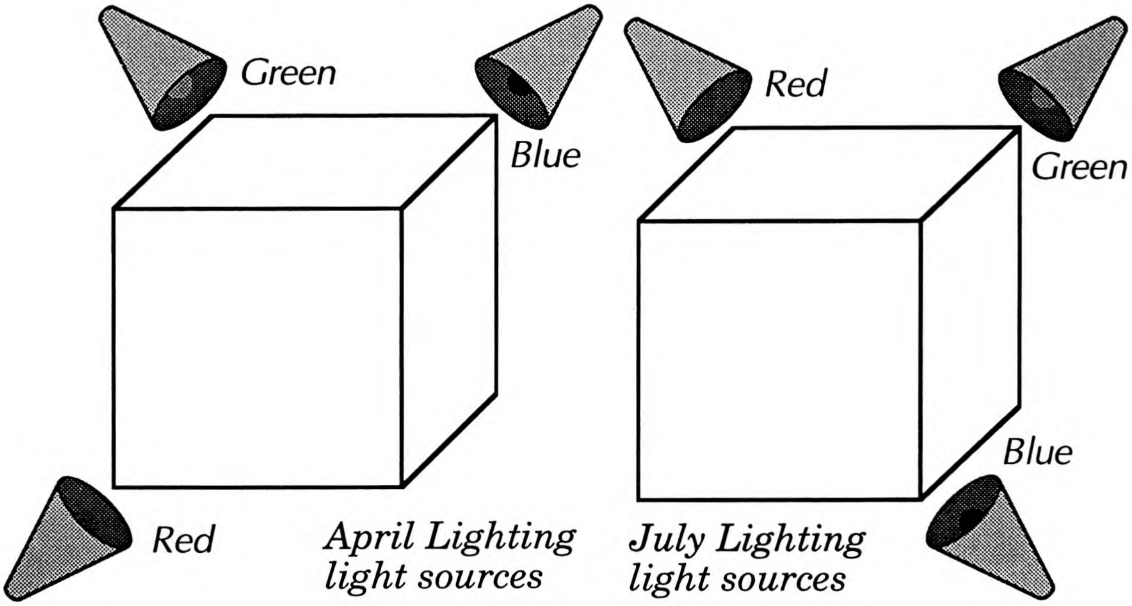
Other types of graphs have other types of vectors. For instance, Spherical and Cylindrical use the functions FromSpherical and FromCylindrical to convert their coordinates. The only requirement is that it calculates to a three vector (in a three-dimensional graph).

Optical Characteristics

The first pop-up menu of a Surface plot proposition sets the fundamental graphic qualities of the plot. The optical characteristics can be set to one of four options:

Optical Characteristics	Appearance
Transparent (Macintosh only)	As viewed through colored cellophane
Translucent (Macintosh only)	As viewed through colored wax paper
Opaque	Made of solid matte colors with uniform diffuse illumination
Illuminated	Made of solid matte colors illuminated by a light source over your left shoulder
April Lighting	More subtle colors due to individually positioned red, green, and blue light sources
July Lighting	Richer colors due to individually positioned red, green, and blue light sources

Transparent and Translucent are only supported on Macintosh screens; they are not supported when printed or with MS Windows computers.



Mesh

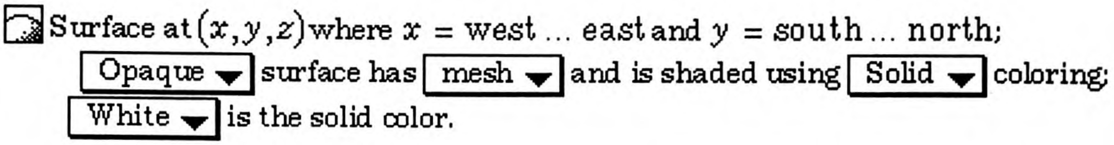
The second pop-up menu of a Surface plot proposition determines whether it is displayed with a mesh (a rectangular grid connecting the vertices generated from the location vector).

Color Schemes

The third pop-up menu of a Surface plot proposition specifies the color scheme and creates unique additional information for each of its five settings: Solid, Gradient, Custom, Complex, and Direct. The color scheme setting determines what appears in the rest of the proposition sentence. Each of these color schemes is discussed below.

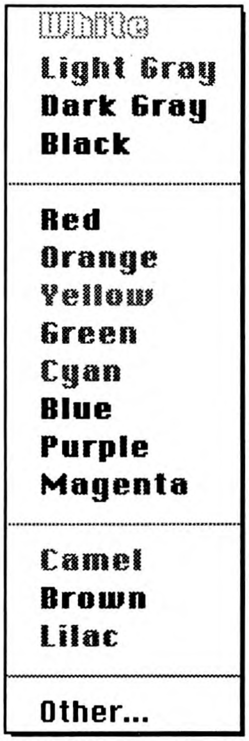
Solid

Solid is the most simple color scheme. The last part of a surface plot proposition using Solid coloring contains a single pop-up menu that can be set to any color.



Surface at (x,y,z) where x = west ... east and y = south ... north;
Opaque ▾ surface has mesh ▾ and is shaded using Solid ▾ coloring;
White ▾ is the solid color.

In all cases, this menu defaults to white, but it can be set to one of fifteen named colors:



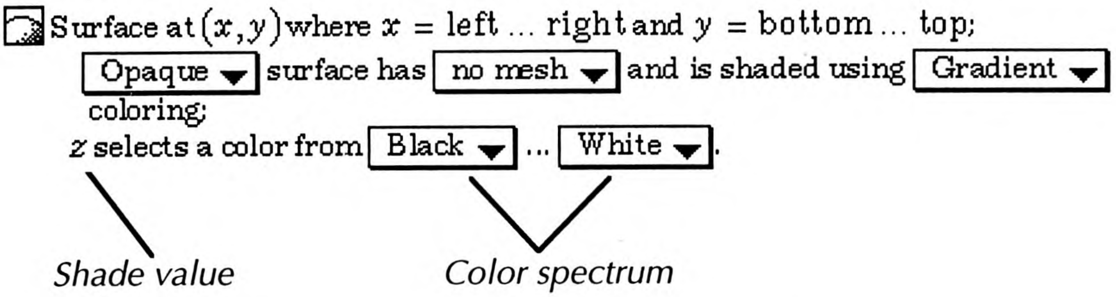
...or, using the last option, to any of millions of colors. The last option, Other..., brings up your computer's standard color selection dialog. With this dialog, you can select from up to 16 million colors, depending on your computer and display device.

If the optical characteristics pop-up menu is set to Transparent, Translucent, or Opaque, all panels in the plot show the same selected color. Illuminated plots combine the lighting effects and the surface color; some panels appear darker than others, and the colors can change for April and July lighting. Solid white is usually best for these.

Illuminated 3D, Spherical 3D, and Cylindrical 3D graphs generate plots with a Solid color scheme.

Gradient

The Gradient scheme uses colors which vary depending upon an expression. The last part of a plot proposition using the Gradient color scheme contains a "shade value," and two pop-up menus used to define a limited color spectrum.



Surface at (x,y) where x = left ... right and y = bottom ... top;
Opaque ▾ surface has no mesh ▾ and is shaded using Gradient ▾ coloring;
z selects a color from Black ▾ ... White ▾.

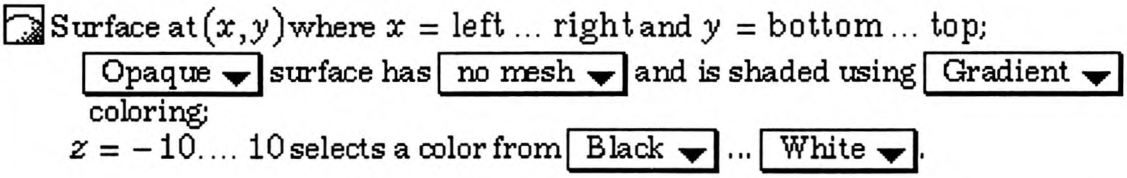
For the Gradient color scheme, a value is derived for each panel and that value specifies the color for that panel; the color of each panel is calculated independently.

After the semicolon, the shade value expression (in this case, z) specifies a color selected from the continuum ranging between the two colors. The default shade value for a Density 2D graph (the only graph that creates a gradient plot by default) is the dependent variable.

The default range for the shade value is 0...1. Where the value of that variable is less than or equal to zero, the first of the two colors is shown. Where the value of that variable is greater than or equal to one, the second color is shown. For surface panels that have values between zero and one, some mix of the two selected colors is shown.

You can replace the shade value with any expression. But only expressions that evaluate to real numbers between zero and one produce a shade between the two colors (unless you specify another range of values as explained below). If the expression evaluates to a number outside the range 0...1, one of the two defining colors is used. If the expression evaluates to an unknown value (the question mark), or a complex number, the panel is shown as white transparent. If the expression evaluates to a matrix (or vector) an error is reported.

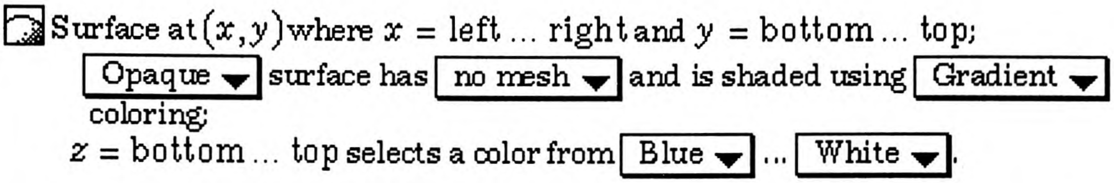
To specify a range of shade values other than 0...1, edit the proposition and insert a different shade value range.



Surface at (x,y) where x = left ... right and y = bottom ... top;
Opaque ▾ surface has no mesh ▾ and is shaded using Gradient ▾ coloring;
$z = -10 .... 10$ selects a color from Black ▾ ... White ▾.

In this case, –10 corresponds to the first color in the color spectrum and 10 corresponds to the second. All values within the given range are displayed in a shade between the two colors. In this example, any value greater than 10 is white and any value less than –10 is blue.

You can also set the range values to named variables or constants, such as bottom and top.



Surface at (x,y) where x = left … right and y = bottom … top;
Opaque ▾ surface has no mesh ▾ and is shaded using Gradient ▾ coloring;
z = bottom … top selects a color from Blue ▾ … White ▾.

Values for the constants bottom and top are given in the graph bounds at the top of the graph details. With this set up, the panels at the top of a plot are white. Those at the bottom are blue, and those in between range from solid blue, through light blue, to white.

If the graph bounds are changed (e.g., with the Knife) the colors automatically track the top and bottom.

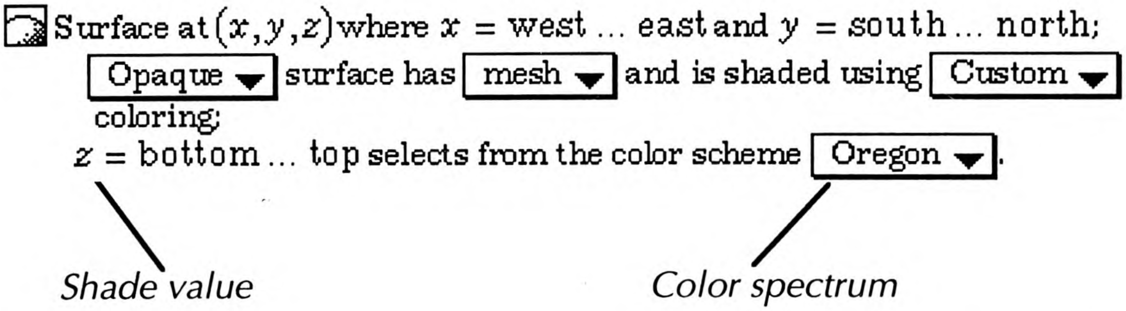
The following table presents a set of possible shade values for the default range (0...1) and a description of the resulting colors. This table assumes that the first pop-up menu is set to red and the second to white.

Shade value	Resulting color
–2.00	Solid red
0.00	Solid red
0.30	Dark pink
0.80	Light pink
1.00	White
7.00	White
3+4*i*	Hollow (clear panel)
?	Hollow (clear panel)
(0.1, 0.2)	Error (a vector)

Density 2D graphs are the only graphs that generate plots with Gradient color schemes by default. However, you can easily change some other scheme to Gradient.

Custom

The last part of a plot proposition that uses the Custom color scheme contains a “shade value,” and a single pop-up menu that can be set to one of seven United States names: Arizona, Kansas, Louisiana, New York, Oregon, Wyoming, and Virginia. Each state name designates a particular group of colors. Kansas uses five colors, all other state names use three colors and black and white.



Surface at (x,y,z) where x = west ... east and y = south ... north;
Opaque ▾ surface has mesh ▾ and is shaded using Custom ▾ coloring;
z = bottom ... top selects from the color scheme Oregon ▾.

The Custom color scheme use a principle similar to Gradient color scheme. A value is derived from an expression for each panel and that value specifies the color for that panel. The color of each panel is calculated independently.

If the shade value expression is between the specified minimum and maximum (in this case, bottom ... top), colors from the named group are used to color a particular panel. If the expression evaluates to a number outside that range, one of the extreme colors is used. (Except for the Kansas color group, the extreme colors are white and black.) If the expression evaluates to an unknown value (the question mark), or a complex number, the panel is shown as white transparent. If the expression evaluates to a matrix (or vector) an error is reported.

By default the range expression bottom ... top is the range of the shade value. But, as with the Gradient color scheme, this can be edited freely. The following table assumes that the range expression is set to 0 ... 1.

Value	Arizona	Kansas	Louisiana
≤ 0	Black	Violet	Black
.25	Brown	Blue	Blue
.50	Red	Green	Violet
.75	Yellow	Yellow	Red
≥ 1	White	Red	White

Value	New York	Oregon	Wyoming	Virginia
≤ 0	Black	Black	Black	Black
.25	Green	Blue	Violet	Brown
.50	Yellow	Green	Red	Gray
.75	Red	Yellow	Orange	Blue
≥ 1	White	White	White	White

As with the Gradient color scheme, the shade value range, by default, is given as bottom ... top. You can specify another range of values, to compress or expand the colors available for the panels of a particular Surface plot.

For example, to use a fine-grained set of colors from purple through wine red, set the color group to Louisiana, and change the shade value range to: z = bottom – 9 ... top + 9.

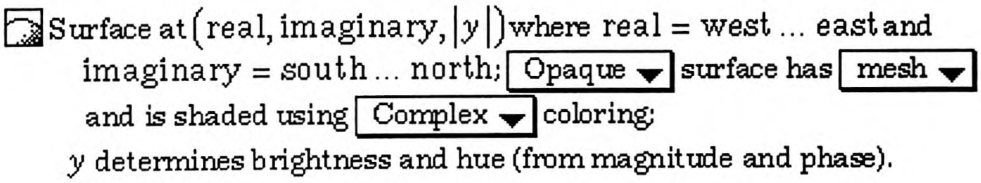
The following table presents a set of possible shade values and a description of the resulting colors. This table assumes that the pop-up menu is set to the Oregon color group (the default), and that the value range is set to 0 ... 1.

Shade value	Resulting color
–2.00	Black
0.00	Black
0.10	Very dark blue
0.20	Dark blue
0.25	Bright blue
0.30	Blue with a bit of green
0.40	Sea green
0.50	Bright green
0.60	Chartreuse
0.70	Yellow with a bit of green
0.75	Bright yellow
0.80	Light yellow
0.90	Very light yellow
1.00	White
7.00	White
3+4i	Hollow (clear panel)
?	Hollow (clear panel)
(0.1, 0.2)	Error (a vector)

This scheme is used for surface plots generated with Color 3D graphs.

Complex

Complex Surface plots use the magnitude and phase of the shade expression (in this case, y) to determine the brightness and hue of the surface panels. These colors are not adjustable. However, you can shift the magnitude or phase of the shade expression by multiplying it by a real or imaginary value (or both). If you replace the shade expression with explicit values (real or imaginary) all panels take on the same color.



Surface at (real, imaginary, $|y|$) where real = west ... east and imaginary = south ... north; Opaque ▾ surface has mesh ▾ and is shaded using Complex ▾ coloring;
y determines brightness and hue (from magnitude and phase).

The Complex surface coloring scheme is similar to that used for complex tables.

The following table presents a set of possible shade values and a description of the resulting colors for the Complex color scheme. (Note: precise colors depend upon your screen or printer. Do not take this table literally.)

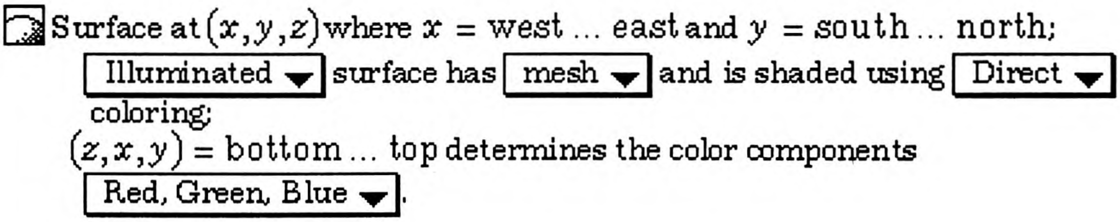
Shade value	Resulting color
–100.00	Near white
–10.00	Light blue
–0.30	Dark blue
0.10	Dark brown
0.30	Mustard yellow
10.00	Light yellow
100.00	Near white
–10.00i	Very light green
–0.30i	Army green
0.30i	Violet
10.00i	Very light lavender
–0.2 –0.2i	Navy blue
0.2 – 0.2i	Pea green
0.2 + 0.2i	Brown
–0.2 + 0.2i	Purple
?	Hollow (clear panel)

This scheme is used for surface plots generated with Complex 3D graphs.

Direct

The Direct coloring scheme lets you numerically specify the precise color you want. Although it is the most powerful coloring scheme, it is also the hardest to use.

You can only create Surface plots that use the Direct color scheme by modifying an existing Surface plot proposition; none of the options on the Graph menu produce this kind of plot. Choose Direct from the third pop-up menu in the proposition. Then change the single value after the semicolon to a three vector (e.g., z, x, y). The elements of this vector determine the value of the three color components named in the final pop-up menu.



Surface at (x,y,z) where x = west … east and y = south … north;
Illuminated ▾ surface has mesh ▾ and is shaded using Direct ▾ coloring;
(z,x,y) = bottom … top determines the color components Red, Green, Blue ▾.

There are five possible interpretations of this vector:

Red, Green, Blue
Cyan, Magenta, Yellow
Hue, Lightness, Saturation
Hue, Saturation, Value
Y, I, Q

Red, Green, Blue

For this setting, each value in the three vector must be between 0 (black) and 1 (full color). White is (1, 1, 1), black is (0, 0, 0), and mixtures make all other colors. This model is closest to the hardware; video screens usually have three color guns responsible for the red, green, and blue parts of the image. The RGB model is used for any hardware device that starts with black and adds light to it to get colors and ultimately white; so-called "additive" colors.

Cyan, Magenta, Yellow

Each value in the three vector must be between 0 (white) and 1 (full color). Black is (1, 1, 1) and white is (0, 0, 0). The CMY model is often used for printing processes that start with white paper and subtract light with pigments to make colors; so-called "subtractive" colors.

Hue, Lightness, Saturation

In this model, the hue is an angular quantity measured in radians. Zero is red, 2.1 is green, 4.2 is blue, and 6.3 (actually 2π) is red again. Lightness ranges from 0 (black) to 1 (white). The saturation also ranges from 0 (grays) to 1 (rich colors). If the lightness is in the middle of its range, then the saturation and hue become important. Extreme values for lightness reduce the effect of the saturation value.

Hue, Saturation, Value

In this model, hue and saturation function the as in the HLS model. Value ranges from 0 (black) to 1 (rich colors and white). The color selection dialog (used to set the color(s) for Solid or Gradient plots if you choose Other), uses this model.

Y,I,Q

This model is a matrix rotation in color space of the RGB model. It is used for broadcast television. The Y value specifies the brightness, the I value varies from cyan to red-orange, Q ranges from green to violet. I and Q assume values in the approximate range ±0.6.

The following tables displays example values the different models use for nine different colors. Values represented by an x can take on any value.

Color	RGB	CMY
Black	(0, 0, 0)	(1, 1, 1)
Gray	(.5, .5, .5)	(.5, .5, .5)
White	(1, 1, 1)	(0, 0, 0)
Red	(1, 0, 0)	(0, 1, 1)
Yellow	(1, 1, 0)	(0, 0, 1)
Green	(0, 1, 0)	(1, 0, 1)
Cyan	(0, 1, 1)	(1, 0, 0)
Blue	(0, 0, 1)	(1, 1, 0)
Magenta	(1, 0, 1)	(0, 1, 0)

Color	HLS	HSV	YIQ
Black	$(x, 0, x)$	$(x, x, 0)$	(0, 0, 0)
Gray	$(x, .5, 0)$	$(x, 0, .5)$	(.5, 0, 0)
White	$(x, 1, x)$	$(x, 0, 1)$	(1, 0, 0)
Red	(0, .5, 1)	(0, 1, 1)	(.3, .6, .21)
Yellow	$(\pi/3, .5, 1)$	$(\pi/3, 1, 1)$	(.89, .32, -.31)
Green	$(2\pi/3, .5, 1)$	$(2\pi/3, 1, 1)$	(.59, -.28, -.52)
Cyan	$(\pi, .5, 1)$	$(\pi, 1, 1)$	(.7, -.6, -.21)
Blue	$(4\pi/3, .5, 1)$	$(4\pi/3, 1, 1)$	(.11, -.32, .31)
Magenta	$(5\pi/3, .5, 1)$	$(5\pi/3, 1, 1)$	(.41, .28, .52)

For examples of graphs using the direct color scheme, see the "Direct Color Models" notebook (Macintosh) or "DirectC.The" notebook (Windows) on your distribution disks in the Graphics directory.

Graph Proposition Elements

Using the commands on the Graph menu, you can create a wide range of two- and three-dimensional graphs with several different types of Line, Contour, Scatter, and Surface plots. You can also add all types of plots to existing graph theories with the commands on the Additional submenu.

After a graph is created, you can use the plot propositions in the graph details to create plots of your own design. The commands on the Graph menu create and augment graphs using a number of defaults. If you edit plot propositions directly, a number of error messages are displayed at intermediate steps, often indicating that certain variables do not currently have values.

To add a new plot to a graph, copy a plot from a neighboring graph, or use the commands on the Graph menu's Additional submenu. If an equation is selected when you choose one of these options, the equation is used to generate the new plot. If no equation is selected, an empty proposition of the selected type is added to the graph details, or a duplicate of an existing plot, or a dialog box may come up asking for the independent and dependent variables.

You should never have to declare any of the names for the graph bounds (left, right, north, west, top, etc.). If such a dialog box does comes up, you are probably trying to copy a plot between a two-dimensional graph theory and a three-dimensional graph theory. This is rarely the best way to add a plot to a graph theory. Copying between graph theories of the same type works fine, as do the Additional commands.

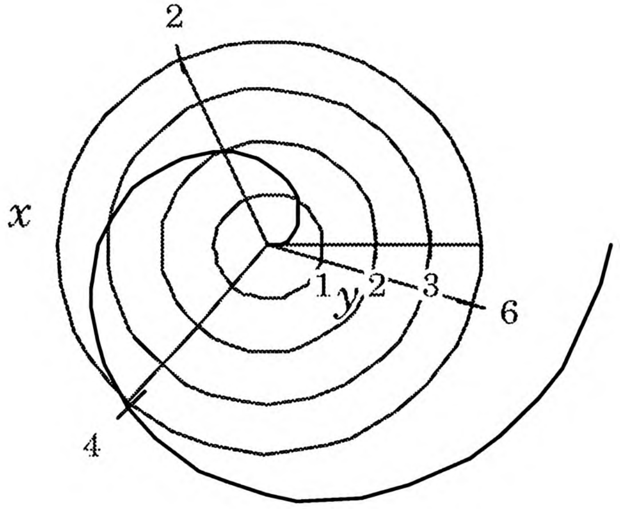
Most two- and three-dimensional graphs are created with all of the necessary name declarations in the Declarations comment near the top of the details box. However, if you are trying to make a polar, cylindrical or spherical plot in a rectangular graph theory, you will need to declare the name "radius." Once declared, put the name declaration in the graph details, not in the Declarations comment for the main theory.

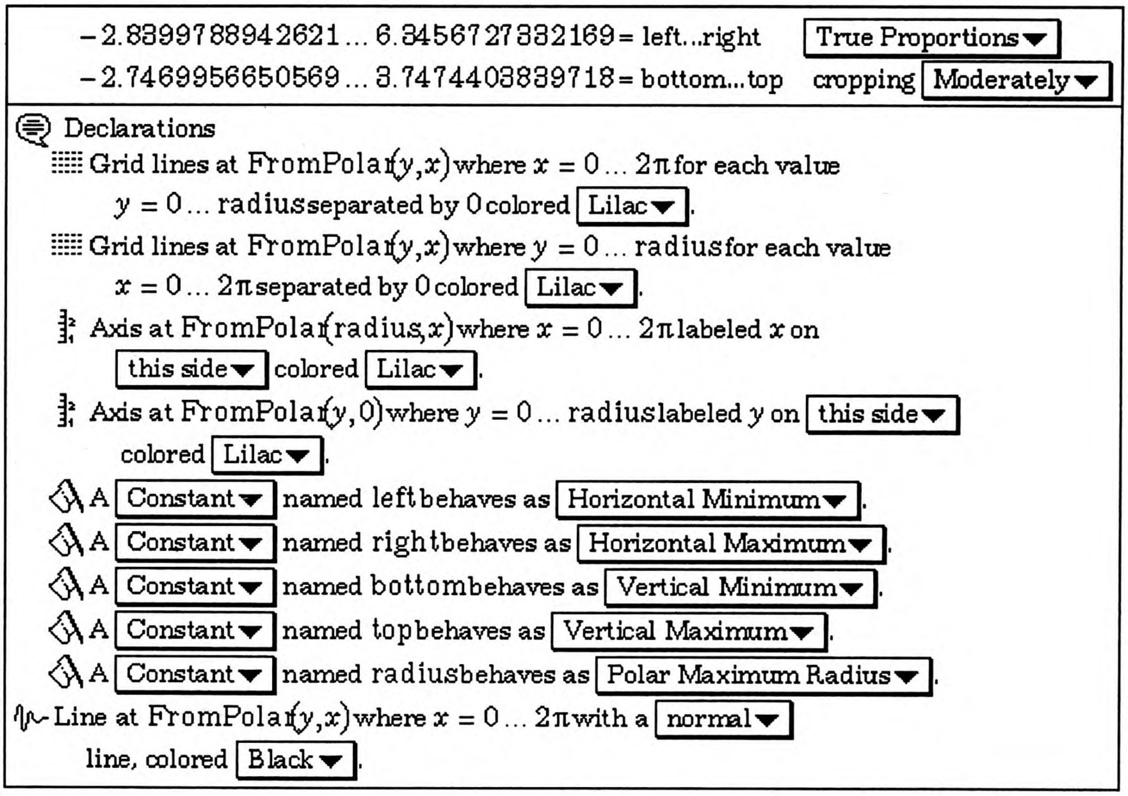
When you cut, copy, and paste name declarations, you may get error messages declaring inconsistencies or unavailable values.

To gain a better understanding of the graph details, consider two graphs, one polar, one linear, created from the simple equation $y = x$. Each graph's details (with exposed declarations) are shown below the graph.

Polar

y is the radius (*r*) and *x* is used as the angular value (often θ)





−2.8399788942621 … 6.3456727332169 = left…right True Proportions ▾

−2.7469956650569 … 3.7474403839718 = bottom…top cropping Moderately ▾

Declarations

Grid lines at FromPolar(*y*,*x*) where *x* = 0 … 2π for each value *y* = 0 … radius separated by 0 colored Lilac ▾.

Grid lines at FromPolar(*y*,*x*) where *y* = 0 … radius for each value *x* = 0 … 2π separated by 0 colored Lilac ▾.

Axis at FromPolar(radius,*x*) where *x* = 0 … 2π labeled *x* on this side ▾ colored Lilac ▾.

Axis at FromPolar(*y*,0) where *y* = 0 … radius labeled *y* on this side ▾ colored Lilac ▾.

A Constant ▾ named left behaves as Horizontal Minimum ▾.

A Constant ▾ named right behaves as Horizontal Maximum ▾.

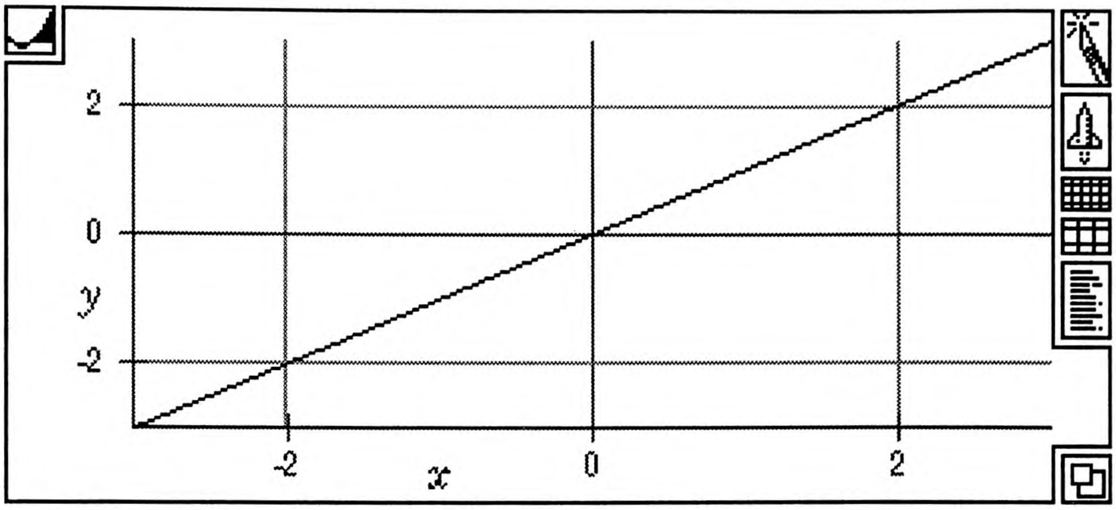
A Constant ▾ named bottom behaves as Vertical Minimum ▾.

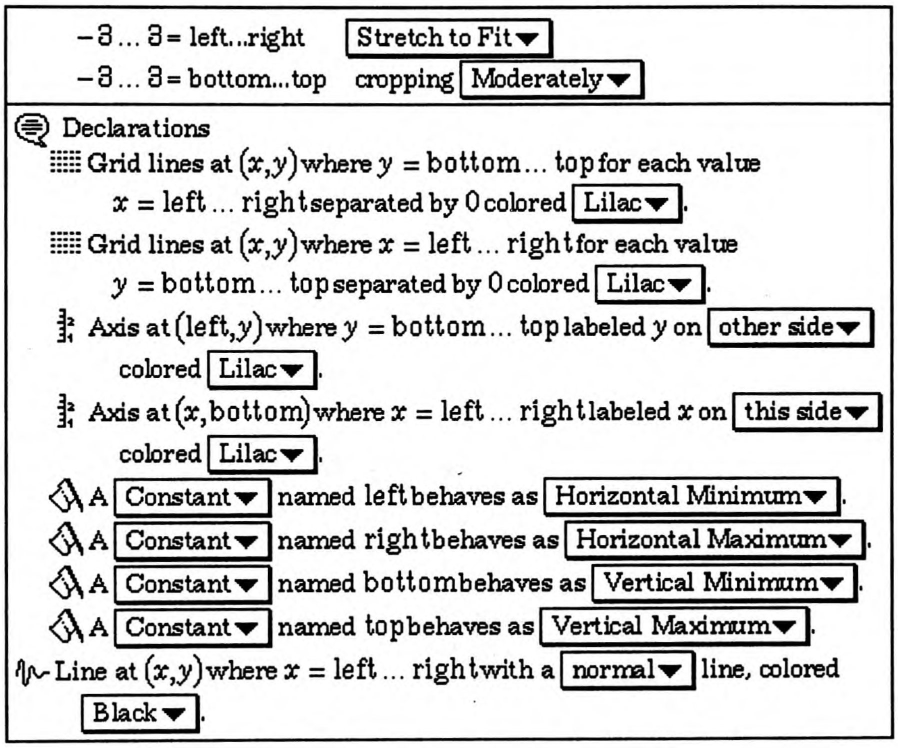
A Constant ▾ named top behaves as Vertical Maximum ▾.

A Constant ▾ named radius behaves as Polar Maximum Radius ▾.

Line at FromPolar(*y*,*x*) where *x* = 0 … 2π with a normal ▾ line, colored Black ▾.

Linear





−3 … 3 = left…right [Stretch to Fit ▼]
−3 … 3 = bottom…top cropping [Moderately ▼]

Declarations
Grid lines at (x,y) where y = bottom … top for each value x = left … right separated by 0 colored [Lilac ▼].
Grid lines at (x,y) where x = left … right for each value y = bottom … top separated by 0 colored [Lilac ▼].
Axis at (left,y) where y = bottom … top labeled y on [other side ▼] colored [Lilac ▼].
Axis at (x,bottom) where x = left … right labeled x on [this side ▼] colored [Lilac ▼].
A [Constant ▼] named left behaves as [Horizontal Minimum ▼].
A [Constant ▼] named right behaves as [Horizontal Maximum ▼].
A [Constant ▼] named bottom behaves as [Vertical Minimum ▼].
A [Constant ▼] named top behaves as [Vertical Maximum ▼].
Line at (x,y) where x = left … right with a [normal ▼] line, colored [Black ▼].

Location Vectors

In the polar graph, the location vectors for all plots (including the Axes and Grid Lines) are mapped through the FromPolar function. The native coordinate system of a two-dimensional graph theory is rectangular; FromPolar is used to convert the coordinates. This function is available as a predefined function. If you create a polar graph by selecting the menu option, a dialog box may come up asking if you want to declare this function as a predefined function. The name declaration is included under the declarations comment at the top of the current notebook.

You can use working statements to create other mapping functions of your own design (e.g., FromHyperbolic) and use them in the same fashion.

Graph Bounds

Graph bounds are always in the native rectangular coordinates, even if the displayed graph uses another coordinate system. Whenever a graph is created, a set of default graph bounds is generated based on the function being graphed.

You can change the graph bounds with the Knife and the open hand pointer, or by editing the values in the graph details. If you want to look at the upper right-hand quadrant of a polar graph, set the bounds to 0 ... 1 = left ... right and 0 ... 1 = bottom ... top.

Ranges and Domains

In polar coordinate systems, it is customary to put the dependent variable (y or r) before the independent variable (x or θ). In rectangular coordinate systems, the order of the variables is reversed. The independent variable (x) comes first.

The domain of the independent variable (x) in a linear graph is, by default, dependent on the graph bounds. The Line plot in the above graph derives x values from the range left ... right, whereas the Line plot in the polar graph uses a "hard-wired" domain of 0 ... 2π. Scrolling either plot with the open hand pointer changes the graph bounds, but the domain of a polar plot is not affected by this change.

In a polar plot, the domain of the independent variable "wraps around" on itself. By default, the domain is one revolution from zero to 2π, measured in radians. The Axes and Grid Lines plots also use this domain. To change the domain in polar plots you have to edit these values directly; they are not dependent on the graph bounds as in linear plots. You must select the values and enter new numbers (or names or expressions that evaluate to numerical values).

The range of the dependent variable in a both graphs is dependent on the equation used to generate the graph. In a linear graph, these values are displayed along the vertical (bottom ... top) axis. In a polar graph the dependent variable is displayed along the radius.

Other Differences

The name "radius" is a defined constant created and declared for all polar (and spherical and cylindrical) graphs. Its name declaration is placed in the Declarations comment at the top of the graph details. The value assigned to this constant is dependent on the logical size of the graphing area. If you zoom out the radius increases and a new set of Axes and Grid Lines are displayed using the new value.

The default aspect ratio for polar graphs is True Proportions. For linear graphs, the default is Stretch to Fit.

ANIMATION

Theorist's animation facility lets you view a family of related graphs in quick succession. The animation creates the illusion of movement by creating several versions of a graph as one variable changes, saving these images, and displaying them in rapid succession.

Animation is a technique for adding another dimension to a graph: time. When you animate a graph that displays the relationship between two variables, you introduce a third variable that changes through time. Similarly, a graph that displays the relationship between three variables, such as an Illuminated 3D graph, when animated, is actually displaying the relationship between four variables.

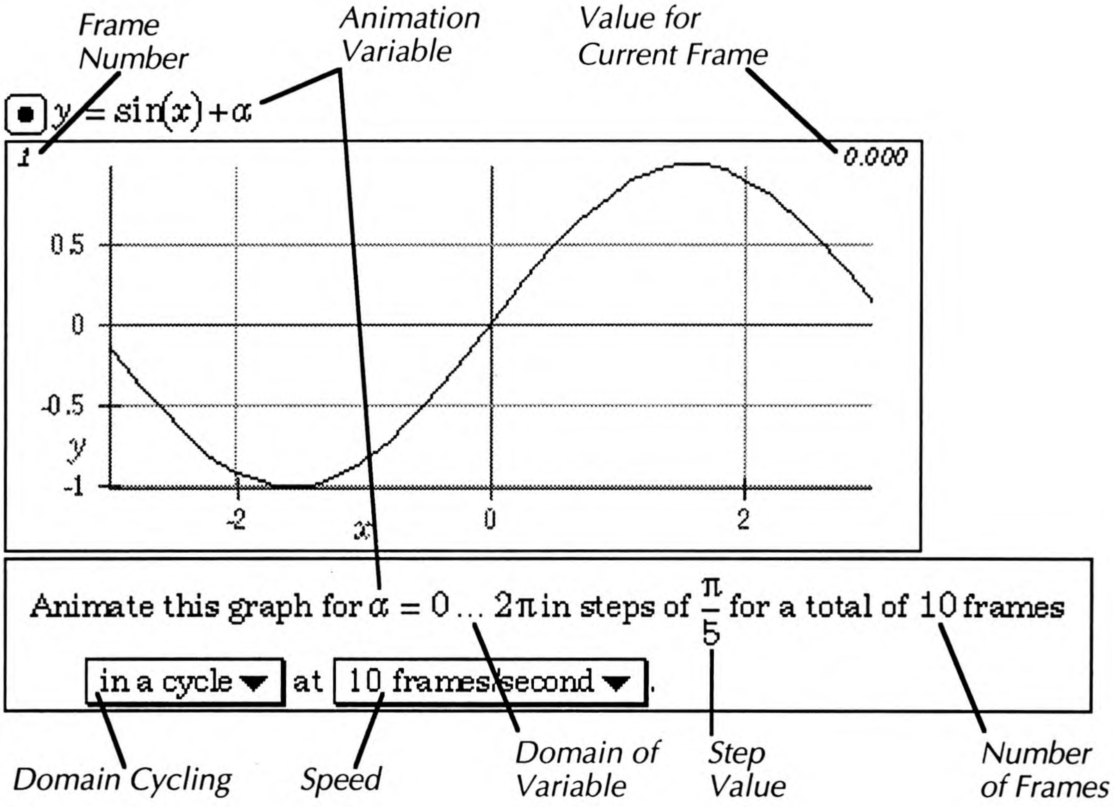
To create an animation, start with a two- or three-dimensional graph. The easiest way to animate a graph is to add a new variable, the animation variable (a, or any name of class Variable) to one of the original equations. This variable can be incorporated in the equation in any fashion. You can add it to any existing variable or expression, or multiply it onto any variable or expression, or use it alone (e.g., $\sin(a)$). The animation variable can also appear in any expression that affects the graph, including the graph bounds and all plot propositions. The variable can appear once, or several times in several expressions.

If you incorporate an undefined variable in an expression used to generate a graph, an error message appears stating that a value can not be found for the new variable. This is perfectly normal. If you want to avoid this message, you can add an equation to temporarily define the variable (e.g., $a = 1$), but this is not necessary.

Select the animation variable (or any expression containing the animation variable—but no other variable), and choose **Graph▶Animation▶Start**. The animation appears in place of the graph proposition.

The graph is rendered (note the calculator icon), displayed briefly, and saved in memory once for each frame. When all the images are rendered and saved, the animation starts in earnest. Frames flash on the screen, creating the illusion of motion.

To halt the animation click the mouse anywhere in the notebook. To resume, click within the graph viewport. To stop the animation and return the viewport to the inanimate graph state, choose **Graph▶Animation▶Stop**. This stops animation, throws away the rendered frames currently in memory, and gets rid of the animation controls box below the graph.



Animate this graph for $\alpha = 0 \ldots 2\pi$ in steps of $\frac{\pi}{5}$ for a total of 10 frames

in a cycle ▼ at 10 frames/second ▼.

The size of the animation depends upon the size of your graph. A copy of the viewport is displayed, but without the editing icons. You can not edit a graph while animating it.

Below the animation are the animation details, including a statement of the animation variable, its domain, and other numbers and settings.

The statement says "Animate this graph for" followed by the animation variable and the domain over which the animation variable will vary. When the graph is created it is initialized to run from zero up to 2π (6.28). You can change either or both of these values.

The step value is the amount the animation variable increases from one frame to the next.

Functions that are naturally periodic should use the "in a cycle" setting. Images for the starting and ending values appear the same; you only want one frame for this value, not two. If your function is non-periodic, use the "one-way" setting which pauses briefly between the last and first frame.

More frames also equals more time to render them.

Following the domain in the statement is the total number of frames to use in the animation. It's simple: more frames equals more quality, but it uses more memory. More frames produce smoother animations. Available memory limits the actual number of frames that you can create. When you make the graph it starts at ten (regardless of the available memory). If more memory is available, you can increase the number of frames. If not enough memory is available, decrease the number of frames. (Otherwise frames are generated during the animation, slowing it significantly.)

Increasing the number of frames decreases the step value and improves the visual resolution of the animation. The step value is determined by dividing the range of values by the number of frames. Changing the starting or ending domain values (or both) adjusts the step size but does not change the number of frames. Changing the step size directly does affect the number of frames.

The domain cycling pop-up menu determines whether the last frame is included. If you are animating one way the last frame is at the last value in the domain. If in a cycle, the last frame is one step in from the last value; it is assumed the first value in the domain corresponds to the last.

For comparison, most cinematic movies are displayed using thirty frames per second.

Use the speed pop-up menu to set the speed of the animation playback. Speeds range from one frame every two seconds (1/2 fps), to thirty frames per second. One additional setting, "Maximum," displays the images as fast as your machine can move them from memory to the screen. These settings are approximate. If the images are very large, or you are running Theorist on a slower CPU, the display might skip steps in order to maintain chronological correctness.

Jumpy Animations

There are several possible reasons why an animation may appear jumpy (as opposed to smooth and flowing).

Too Few Frames

The most common reason for jumpy animations is not enough frames, which you can easily change. More frames produce smoother animations. However, the total possible number of frames is limited by the amount of memory available to Theorist. It's simple: using more frames makes for better quality but uses more memory.

Frame Rate Too Fast

If the animation seems only slightly jumpy, it could be that the frame rate is too high. For instance, if you set it to ten frames per second but your computer can keep up at only nine frames per second, it will skip about every tenth frame in order to keep up. To fix the problem, make the graph smaller, set your computer to use black and white instead of color or gray-scales, or choose a slower frame rate.

Not Enough Memory

If the animation is very jumpy, and you see the calculator icon flashing in the lower left corner of the graph, Theorist may have run out of memory. When this is the case, Theorist starts throwing away images, requiring an image to be recalculated and redrawn the next time it becomes the frame to display.

Virtual Memory

Virtual memory may slow down the animation and make it look jumpy. Accessing frames from the hard disk instead of from memory is a sure path to jumpy animations. If the hard disk is running constantly as it animates, you can increase the speed by reducing the amount of memory you are using for the image data. (See below.) Once it is below a certain threshold disk access drops dramatically and animation becomes much faster. It may take some experimentation to find this threshold.

Saving Animations (Macintosh only)

To save a series of frames, choose **Graph▸Animation▸Save...** . A file selection dialog appears, asking for a file name and location. If QuickTime is installed, saved animations use the QuickTime format and can be displayed with a wide variety of programs that can incorporate QuickTime movies. If QuickTime is not installed, saved animations use the PICS format

No PostScript information is stored with an animation file.

Images saved in the QuickTime or PICS file are bitmapped images with the same "depth" (number of bits per pixel) as your computer screen. (If you are using more than one monitor, the screen with the deepest pixel depth determines the depth of the saved images.) The amount of memory required for each frame is determined by the screen area it uses (the number of pixels), times the depth of your screen.

Memory Usage

While animating, your computer has a certain limited amount of memory to hold the frames. A simplified formula works like this:

Memory Used = Number of Frames × Height × Width × Screen Depth

The size of the graph on the screen can have a dramatic impact on memory usage. For instance, if you make the graph half as big, you have room for *four times* as many frames.

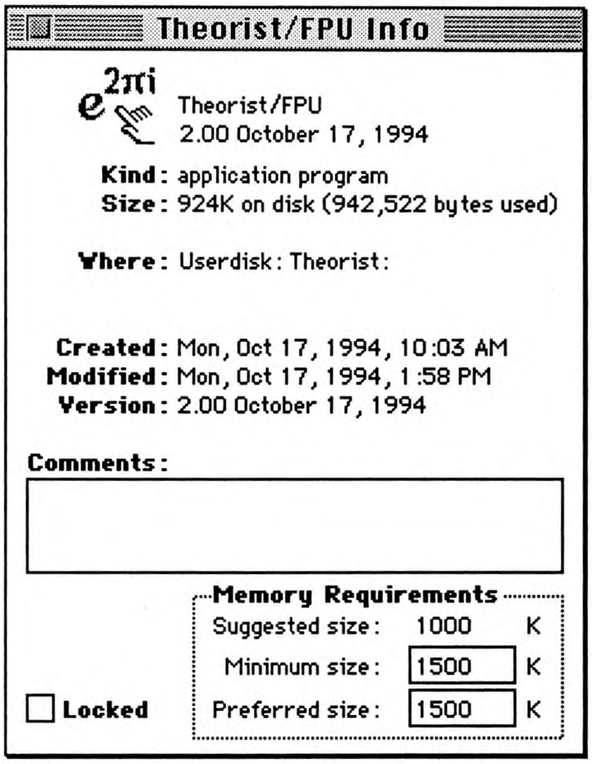
Cutting down your computer's screen depth can also make a big difference. If you normally have eight bits per pixel (typical for a color computer), you can switch to black and white (e.g., for line or contour plots) and have *eight times* as many frames if you are willing to do without colors and shades. Changing from color to gray scale does not change the number of bits per pixel. Usually, the depth is approximately the base 2 logarithm of the number of colors a monitor can display.

Screen Ability	Depth (bits per pixel)
Black & White	1
2-4 shades of gray	2
16 shades of gray	4
16 colors	4
256 shades of gray	8
256 colors	8
16 million colors	24 or 32

Increasing Application Memory Size (Macintosh only)

Even though your Macintosh may have many megabytes of available memory, Theorist may not be able to use it. Theorist uses only the amount of memory it is explicitly allocated in the Finder. Increasing the Theorist application memory size gives Theorist more memory for animations. To do this, select the application in the Finder and choose **Get Info** from the **File** menu to bring up the Info dialog.

This dialog looks slightly different under different System versions.



Theorist/FPU Info

Theorist/FPU
2.00 October 17, 1994

Kind: application program
Size: 924K on disk (942,522 bytes used)

Where: Userdisk: Theorist:

Created: Mon, Oct 17, 1994, 10:03 AM
Modified: Mon, Oct 17, 1994, 1:58 PM
Version: 2.00 October 17, 1994

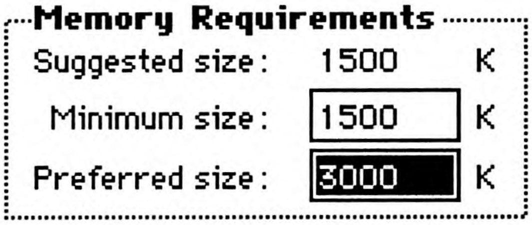
Comments:

□ **Locked**

Memory Requirements
Suggested size: 1000 K
Minimum size: 1500 K
Preferred size: 1500 K

The box in the lower right hand corner specifies the memory to be used by the application. Enter the desired memory where it says "Current size" or "Preferred size".

Shown here is 3000K (three thousand kilobytes), which is 3Mb (three megabytes).



Memory Requirements
Suggested size: 1500 K
Minimum size: 1500 K
Preferred size: 3000 K

Quit Other Applications

A simple way to free up your computer's memory for Theorist is to exit other running application programs. If you exited a spreadsheet application so Theorist could use the memory the program was using, Theorist would have so much more memory you could create a much more complex animation.

Virtual Memory
(Macintosh System 7)

If your Macintosh does not have enough memory to spare for Theorist, you can use its virtual memory capabilities. To do this in the Finder, open the Memory control panel. Click the On button and choose a hard disk to use. With virtual memory up and running, there is much more memory available, some of which you can dedicate to Theorist as described above.

Virtual Memory
(MS Windows)

If your computer does not have enough memory to spare for Theorist, you can use the control panel to access its virtual memory capabilities. With virtual memory up and running, there is much more memory available for Theorist.

Install More Memory

A sure-fire way to make more memory available to Theorist is to install more memory in the computer. This involves buying RAM chips or modules and physically plugging them inside the computer.

Image Calculation and Rendering Speed

For more complex graphs, you may notice that it takes Theorist some time to calculate and draw the graphs as it works at the initial animation stage of calculating, rendering, and storing the images. The speed at which Theorist calculates and renders each frame depends mainly upon:

❶ Complexity of the graph. (How many plots? How complicated are the necessary calculations? What is the graph's resolution?)
❷ CPU speed and existence of math co-processor

The image size and the number of bits per pixel have negligible effects.

Image Animation Speed

With the speed pop-up menu in the animation controls box, you can set the speed at which Theorist flips through and displays the frame images in the actual animating phase (*after* calculating and rendering). There are no calculations involved in this, the actual animation; the computer spends most of its time simply throwing bitmap images up on the screen in rapid succession. The fastest setting is limited by these main factors:

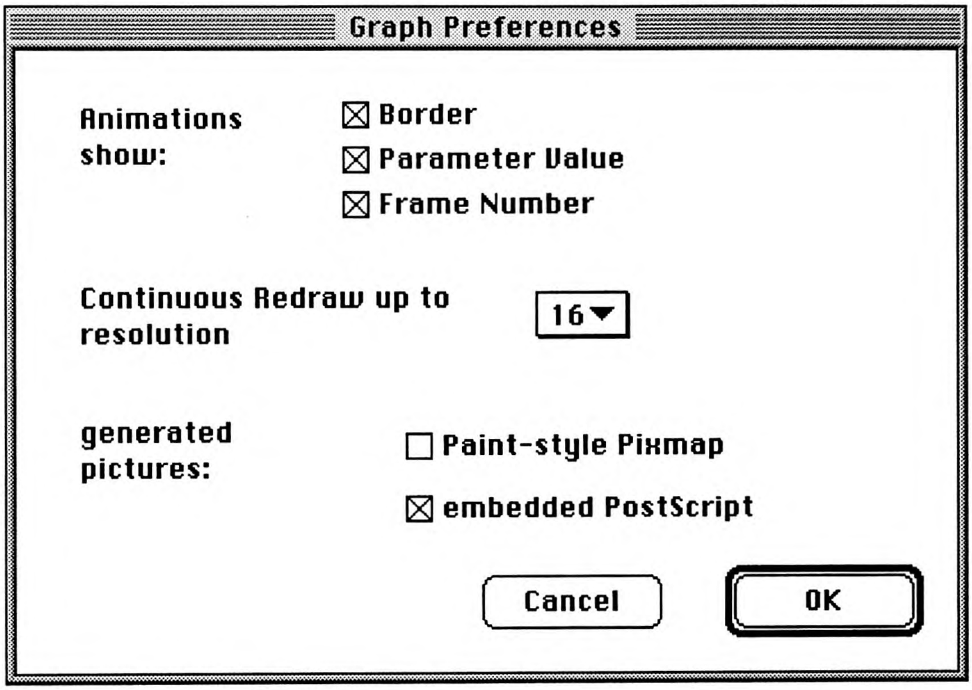
❶ Size of the graph, in pixels
❷ Number of bits per pixel
❸ CPU speed
❹ Hard disk speed (if using virtual memory)

If virtual memory is in use and the hard disk is running constantly as it animates, you can increase the speed by reducing the amount of memory you are using for the image data (see above). Once it is below a certain threshold disk access drops dramatically and animation becomes much faster. It may take some experimentation to find this threshold.

The existence of a math co-processor is irrelevant in this animating phase, as is the complexity and resolution of the graphs. The images have already been calculated; all that is happening now is the machine is moving bytes around.

Preferences

Choose **Graph▸Preferences...** to bring up the graph prefs dialog.



Graph Preferences

Animations show:
☒ Border
☒ Parameter Value
☒ Frame Number

Continuous Redraw up to resolution 16▾

generated pictures:
□ Paint-style Pixmap
☒ embedded PostScript

This dialog lets you change three options related to animations. They normally are all on. The Border option causes a black line to surround each frame. With this off, there are no borders on the images. The Parameter Value option shows the parameter value in the of the viewport's upper right. The Frame Number option shows the frame number in the viewport's upper left.

Printing and Exporting Graphs

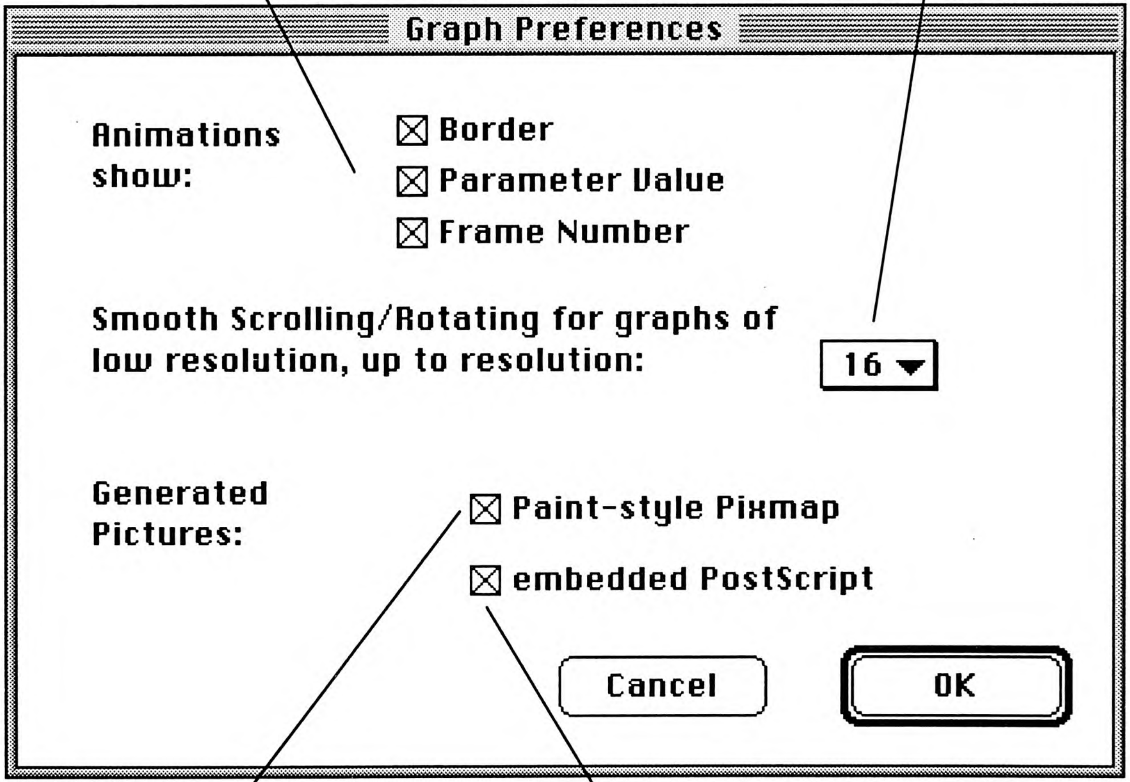
There are a variety of ways to print the graphs you create in Theorist. Notebooks can be printed directly from the program and graphs can also be copied and saved in several formats for exporting to other programs.

Graph Preferences

The **Preferences...** item on the **Graph** menu brings up a dialog in which you can specify various options that affect the appearance of graphs as they are displayed and printed, and the format used to copy graphs to other software packages. Animation preferences are also available, but are discussed elsewhere in this manual.

show/hide border around viewport
show/hide parameter value in viewport
include frame number in viewport

redraw scrolling or rotating graphs up to this resolution



Graph Preferences

Animations show:
☒ Border
☒ Parameter Value
☒ Frame Number

Smooth Scrolling/Rotating for graphs of low resolution, up to resolution: 16 ▼

Generated Pictures:
☒ Paint-style Pixmap
☒ embedded PostScript

make bitmapped (on) or object-oriented (off) image

embed PostScript (Macintosh only)

Smooth Scrolling/ Rotating

This setting determines whether graphs scroll (2-D) or rotate (3-D) smoothly. Set it to Always if you have a high performance computer and you want graphs to redraw as you drag with the hand cursor. Set it to Never if you have a slow computer or you would rather have the graph redraw only after the mouse is released after dragging. Choose one of the other number settings if you want Theorist to decide based upon the particular resolution of a particular graph. This setting does not affect graph printing or exporting and is not related to the Always Redraw item in the Graph menu.

Bitmapped vs. Objects

The Paint-style Pixmap checkbox in the left-hand column determines whether a graph is exported (copied or saved) as a bitmapped image or as an object-oriented description of the panels, vertex locations, and color settings. This option affects exported graphs but not graphs as they appear on the screen (i.e., that with the highest number of bits per pixel).

Bitmapped images have the same pixel depth and color map as the monitor used to generate the images. If you are using a Macintosh with multiple monitors of different types, generated graph images depend on the setting of the deepest screen.

With black and white images, simple bit-by-bit storage takes the least memory. On the other hand, color images that are eight-bits deep (or more) are often more efficiently stored as objects. The amount of memory used for an object-oriented image depends on how many line segments (for Line plots) or how many individual panels (for Surface and Contour plots) are used to make up a graph. This in turn depends on the resolution set for that graph.

The amount of memory used for a bitmapped image depends on the absolute size of the image (amount of screen real estate, in square pixels) and the depth of each pixel (bits per pixel). Memory usage also depends on the size of the graph.

Embedded PostScript
(Macintosh only)

This check box determines whether PostScript descriptions are embedded in exported or printed graphs. If PostScript is not embedded, the Macintosh's native QuickDraw is the display technology used. This option affects exported graphs but not graphs as they appear on the screen.

If you are printing on a StyleWriter, ImageWriter, DeskWriter, or other QuickDraw printer, turn this option off to save processing time and memory. A PostScript description of an image is usually quite large, and unnecessary for these printers. However, this option must be on to print to a PostScript printer.

The PostScript description of a graph always contains full color information and overrides the Paint-style Pixmap setting in this dialog when the picture is printed. The pixmap setting still affects the amount of memory used and the way it displays on-screen and prints on non-PostScript printers. The resolution of the image depends on the resolution of the destination PostScript printer.

Note: Surface plots with Translucent or Transparent color schemes are displayed with hollow panels on PostScript devices.

Including a PostScript description can double or triple the size of an image.

Printing Graphs

To print a graph as part of a notebook choose **Print...** from the **File** menu. (If you are using a Macintosh, the Embedded PostScript option in the Graph Prefs dialog should be checked if you are printing on a PostScript printer.) Printing notebooks is described in the Introduction section of this manual and in the Notebooks section of the Theorist Learning Guide.

Exporting Graph Images

Graphs can be exported as graphic images in two of three different formats (MetaFile for Windows, PICT for Macintosh, and EPS for either system).

Copy as Picture

Choose **Copy as Picture** from the **Edit** menu to copy an image of the selected graph (or proposition or expression) to the clipboard. (The standard Copy command copies graph data, not a graph image.)

The type of image created is determined by the settings in the Graph Preferences dialog box.

An image copied to the clipboard can be pasted into any Macintosh program that accepts PICT format pictures or into Windows program which accepts MetaFile format graphics.

Generate MetaFile

For Windows only; use Generate PICT File on Macintosh.

Select a graph and choose **Generate MetaFile...** from the **Graph** menu to create a file that contains exactly the same information as Copy as Picture puts into the clipboard. The command brings up a dialog that prompts you for a file name and location. The resulting file can be imported into programs which support MetaFiles.

Generate PICT File

For Macintosh only; use Generate MetaFile on Windows.

Select a graph and choose **Generate PICT File...** from the **Graph** menu to create a file that contains exactly the same information as Copy as PICT puts into the clipboard. The command brings up a dialog that prompts you for a file name and location. The resulting file can be imported into programs which support PICT files.

Generate EPS File

Select a graph and choose **Generate EPS File...** from the **Graph** menu to create a file that contains an Encapsulated PostScript description of the image. The command brings up a dialog that prompts you for a file name and location. The resulting file can be imported into programs which support EPS files.

On the Macintosh, holding down the Option key while choosing the **Generate EPS File...** menu item generates a file of type TEXT, rather than the default EPSF type. This file can be transferred to any other computer system and can be opened with a text editor.

An EPS file is a machine and resolution independent description of a graphic image. EPS files can be moved from one machine to another, or from one architecture to another, without loss of meaning. The textual contents of the file follow Adobe Systems standards for Encapsulated PostScript. All surface colors are included, even on monochrome systems.

Do not use EPS files with printers which do not support PostScript.

Editing

Editing mathematical expressions is a distinct art. Theorist incorporates a number of powerful conventions to simplify the modification of expressions.

This section provides a description of:

- The structure of mathematical expressions
- Using the palette to enter expressions
- An example sequence of deletions
- Techniques for selecting and editing:
  - Propositions
  - Expressions and sub-expressions
  - Multiple expressions
  - Names
  - Matrices and vectors
  - Comments
- Examples of creating and modifying:
  - Simple expressions
  - Functions
  - Linear operators
  - Calculus expressions
  - Iteration expressions
  - Matrices and vectors
  - Other expressions

Expression Structure

Creating and editing mathematical expressions is no ordinary task. When publishers typeset mathematics they often refer to it as penalty work; it takes twice as long as typesetting straight text. The computer scientist Donald Knuth went so far as to develop his own typesetting software ($\mathrm{T_EX}$) to create his textbooks just the way he wanted.

The difficulty is inherent in the logical and visual structure of mathematical expressions. Unlike the linear structure of textual information, expressions are hierarchical structures. As an example, consider the following expression:

$$3x\sqrt{\frac{xy}{2}}$$

Most computer languages (Algol and its descendants, FORTRAN, Basic, Pascal, C, etc.) manage to reduce the two-dimensional complexity of such an expression to a linear string of characters. But in a graphic environment, it is possible to see and work with the expression more naturally as a set of nested structures:

$$\boxed{\boxed{3}\bullet\boxed{x}\bullet\boxed{\sqrt{\boxed{\frac{\boxed{x}\bullet\boxed{y}}{\boxed{2}}}}}}$$

Theorist allows you to select and edit each nested object in an expression.

This section discusses:

- Editing basics
- The Precedence Hierarchy and “Fortranish”
- Escape levels

Editing Basics

To edit an existing statement, the most important concept to remember is that each op *encloses* its operands. In fact, each statement, no matter how complex, is actually one or more expressions contained by a single op. This op is the highest-level structure in that expression. In an equation this op is the equals sign. To select an equation, click on the equals sign. You can use the highest-level op of any expression to select the entire expression.

To edit an existing expression, you will get the most consistent results if you select the portion of the expression you want to edit, rather than trying to type into an insertion point.

To create a new statement, press return. An empty statement appears (with a highlighted question mark) ready to be typed into:

□?

If the selection point is in a proposition, the new statement appears immediately after that proposition. If there is no selection, the new statement appears at the bottom of the current notebook.

To create a comment proposition, press enter on the keyboard's numeric keypad. The proposition appears at the bottom of the notebook or after a selected proposition. Text typed into a comment proposition wraps to new lines based on the size of the comment box. To create a hard carriage return (i.e., to force text to start on a new line) in a comment, press shift return. Pressing return alone creates a new statement (as it always does). If you intend to create a comment proposition but mistakenly create a new statement (by pressing return), you can change the new proposition into a comment by pressing enter immediately.

Two commands on the Edit menu, Select Out and Select In, are powerful tools for selecting expressions and subexpressions.

Select Out expands the current selection to the next largest structural level. If two or more selections collide, they undergo pairwise annihilation. That is, if an even number of selections collide, both selections disappear; if an odd number of selections collide, one survives and selects the larger expression.

For example, in the expression $x + y + z$, if x and z are selected, and you invoke Select Out, both selections reach out to enclose the whole sum, but they collide and no selections remain. However, if x, y, and z are each individually selected, Select Out will select the entire sum, as if only one term was selected.

Select In selects all subexpressions of the currently selected expression. If you hold down Shift, Select In selects only the first subexpression.

Select In, Select Out, and additional selection tools are available on the palette.

Typing a closing parenthesis moves the cursor from anywhere inside an expression to the end of that expression.

If you select part (or parts) of an expression, the effect of the characters you type depends on whether the selections (and the entered characters) are ops or literal names (or numbers).

Most ops are created with punctuation symbols (e.g., +, *, /, \). Names are created with letters (Greek or Roman), and numbers are created with Arabic numerals (0, 1, 2, ...). Entering an op encloses any current selection(s) and *applies* the op to that selection (or selections). Typing in a name or literal number *replaces* the selection(s) with that name or number. Several examples are included in this section.

The Tab key moves the selection to the next question mark in the same proposition. If there are no question marks, the selection moves to the next subexpression of the same op. The arrow keys move the cursor in the designated direction. Due to the structure of propositions this may not always be your intended direction.

Precedence Hierarchy and Fortranish

Theorist was designed to allow you to enter and edit equations as naturally as possible, interacting with them on screen as you would on paper. The mouse (and commands such as Select In and Select Out) let you work with expressions in a non-linear fashion. However, keyboard entry is quite linear and perhaps the fastest method for entering expressions.

Theorist offers two different methods of keyboard entry: one technique particularly designed for the graphic editing environment, and another that follows the established conventions of computer programming languages (called "Fortranish").

In general, Theorist follows the traditional rules of operator hierarchy established by the Algol programming language (and used by its heirs, FORTRAN, C, BASIC, Pascal, etc.). The hierarchy that Theorist uses is shown in the following table. Each group of ops is at the same level.

Op Name	Op Character(s)	Notation Form
Square Root	\ *(back slash)*	Prefix
Absolute Value	| *(vertical bar)*	Prefix
Subscript	_ *(underscore)*	Infix
Factorial	!	Postfix
Power (exponent)	^	Infix
Adjoint	Adjoint(	Postfix
Partial Derivative	Derivative(	Prefix
Dot Product	*@	Infix
Cross Product	*#	Infix
Function Call	(*(after function name)*	Infix
Division	/	Infix
Summation	@	Infix (multiple)

Multiplication	*	Infix
Negation/Subtraction	–	Prefix/Infix
Addition	+	Infix
Matrix Row	;	Infix
Matrix Column	,	Infix
Evaluate At	EvaluateAt(	Infix (multiple)
Range	.. *or* ...	Infix
Integral	Integral(*or* $)	Prefix/Infix
Pi Product	# *or* PiProduct(	Infix (multiple)
Conditional	Conditional(	Prefix
Relational Ops	= ≠ < ≤ > ≥	Infix

An op can either be entered before, in between, or after its arguments. That is, ops are applied using prefix, infix, or postfix notation:

- Prefix ops are inserted before an argument. For example, to enter a square root, type a backslash and then the number.

- Infix ops are inserted between their arguments. For example, to enter a sum, type the first expression, a plus sign, and the second expression.

- Infix (multiple) ops have more than two arguments and are inserted between the first and second arguments. Use the tab key to move to the other arguments. For example, the summation op is inserted between the increment variable and the first value for that variable. Press tab to move to the next argument.

- Postfix ops are inserted after their arguments. For example, to enter five factorial, type the number five followed by an exclamation point.

Precedence rules are applied when an expression is potentially ambiguous. For example, the precedence rules dictate the expression $5 + 7 \times 3$ be interpreted as five added to the product of seven and three.

If there is any conflict between ops of the same level, the ops bind from right to left. This is the reverse of the order often used. This means that $a/b/c$ indicates:

$$\frac{a}{b/c}$$

and not:

$$\frac{a/b}{c}$$

By default, Theorist breaks a few of the rules listed in the precedence table. Certain machine responses seem more natural in a graphic environment. For example, if you create an exponent (using the circumflex, shift 6), the cursor stays at the exponent level until you explicitly return it to the baseline (with the esc key).

To create the expression x^{2+n}, using the default settings, type **x^2+n**. Although this makes sense, it breaks with the standard expression hierarchy because the power op should bind more tightly than the addition op.

The Fortranish option (on the **Notebook** menu's **Preferences** submenu) allows control over such bindings while you are typing expressions.

The following table shows how you would create the two expressions x^{2+n} and $x^2 + n$ with Fortranish on and off.

	(default)	
To Create:	**With Fortranish off**	**With Fortranish on**
x^{2+n}	**x^2+n**	**x^(2+n)**
$x^2 + n$	**x^2 esc +n**	**x^2+n**

If you are familiar with entering equations in a programming language, you may want to turn Fortranish on. With this switch on, the hierarchy rules are followed explicitly. If these rules are not second nature for you, leave Fortranish off.

Expressions are copied to the clipboard in the Fortranish syntax. If you Copy an expression selection, the result is a text string representation of the expression that you can Paste into another program that accepts expressions (e.g., programming languages, spreadsheet programs). Use **Copy as Picture** (on the **Edit** menu) to create a graphic copy of an expression.

Fortranish was designed to be as compatible as possible with the equation syntax of other computer systems. Most simple equations should move between systems with a minimum of editing.

If you are familiar with entering equations in Expressionist (the equation editor from Prescience/WMS), leave Fortranish off and enter equations as you would with that software. (Many of Expressionist's keyboard com mands create the same structures in Theorist.)

All examples in this section assume that Fortranish is off, unless stated otherwise.

Escape Levels

Because Theorist does not always follow the precedence rules of equation entry, you need a technique for moving between the logical levels of an expression. Moving down from a superscript, or up from a subscript, or out of a square root, or out of a fraction's numerator or denominator, are all movements from one level of an expression to another escape level. The following ops form one distinct escape level:

- Power (exponents)
- Index (subscripts)
- Division (numerators and denominators)
- Square root
- Absolute value
- Integral
- Summation
- Pi Product
- Partial derivative
- Matrix (each element)
- Evaluate At

Each op that graphically encloses its subexpression makes an escape level.

The easiest way to "escape" from a level is to press the esc key.

USING THE PALETTE

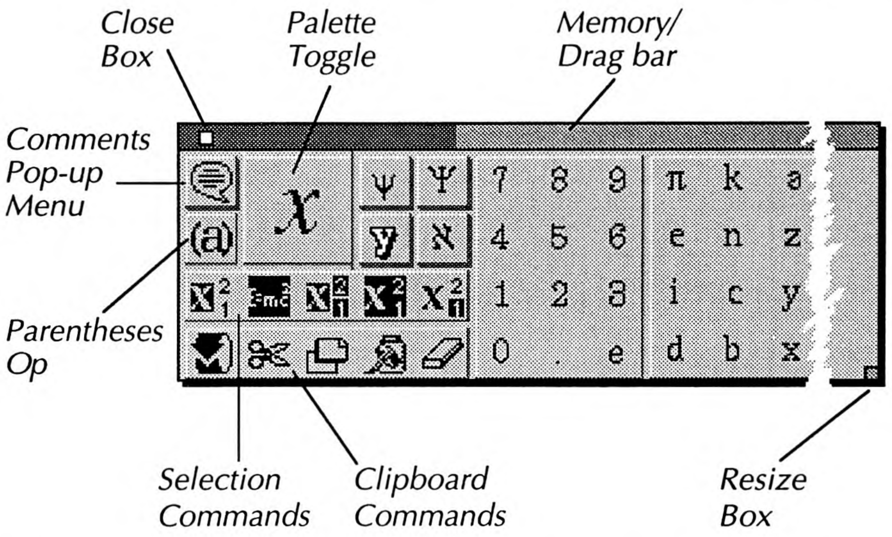
The "palette" is a rectangular window that appears above the current notebook. With the palette you can point and click to select the items you want to use to make up an expression without using the keyboard (except for numbers and new names). For some people, using the palette is easier than typing equations, and is probably faster when you first use Theorist. With practice, keyboard entry is usually faster.

To hide (or show) the palette, choose the **Palette** item on the **Windows** submenu of the **Notebook** menu. If this item is checked, the palette is displayed. To return the palette to its home position, hide it and show it again.

The palette has two sides. The first is called the "variables palette," and the second is called the "functions palette." You can switch between the two sides by clicking on the two big icons on the left side labeled *x* and *f*(*x*).

Shared Palette Abilities

There are some parts of the palette which are common to both the functions palette and the variables palette. These are the drag bar, close box, resize box, comment icon, parentheses op icon, selection icons, clipboard icons, and the toggle button to switch to the other palette.



On the Macintosh, the drag bar indicates the amount of available memory. As mem ory is used, the bar extends to the right. As memory becomes available, the bar moves back to the left. If the bar moves very close to the right-hand edge of the palette, the menu bar flashes on and off indicating that the program is running very low on memory and may run out. This bar is also used to move the palette; just drag it to a new location.

On a computer running MS Windows, the drag bar is titled with the current notebook's name and is used to move the palette; just drag it to a new location. The drag bar is *not* a memory bar as on the Macintosh.

The close box and resize box work like standard window controls.

The comment bubble icon is the pop-up menu for comments, which includes icons to change the font specifications of comments, change comment text colors, and show or hide comment rulers.

The parentheses icon enters the parentheses op, surrounding the selection in parentheses.

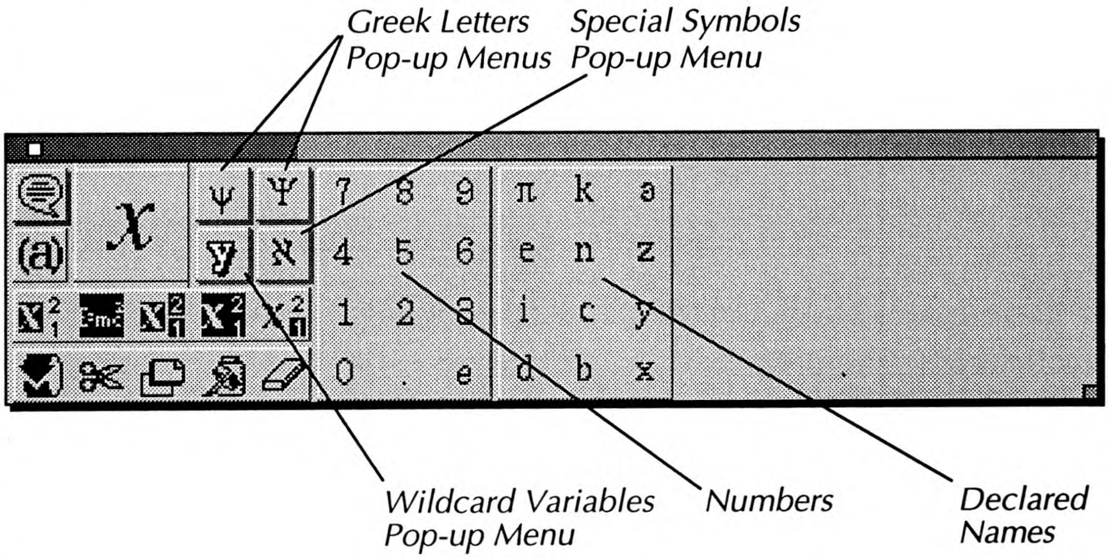
The selection tools make various expression selections as shortcuts to dragging.

The clipboard commands are the same as those on the Edit menu.

These and other shared palette abilities are explained in detail in the Palette section of this manual.

Variables Palette

The variables palette shows variables, constants, and operators that you can insert into an expression. Variables and the other items on this side of the palette always replace the current selection.



Greek Letters Pop-up Menus
Special Symbols Pop-up Menu
(a)
x
ψ Ψ
y ℵ
7 8 9
4 5 6
1 2 3
0 . e
π k a
e n z
i c y
d b x
Wildcard Variables Pop-up Menu
Numbers
Declared Names

Use the ψ pop-up menu to insert lower case Greek letters and the Ψ pop-up for upper case Greeks.

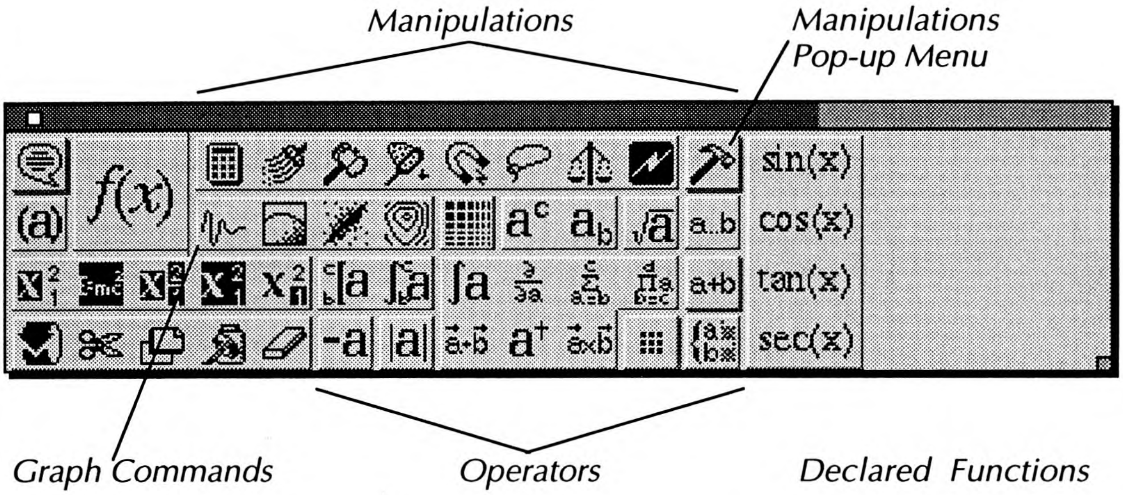
Use the 𝓎 pop-up menu to insert wildcard variables.

The large area on the right of the variable palette displays the declared names of all variables, constants, and operators available in the current notebook.

Variables palette abilities are explained in detail in the Palette section of this manual.

Functions Palette

The functions palette displays ops and functions. All items on this side of the palette enclose an argument. If you click on an op or a function while something is selected, the selection is enclosed by that op or function.



Use the pop-up menu on the bottom row to enter matrices of up to four-by-four elements. This pop-up menu also provides choices for inserting a new row or column in a selected matrix.

The area on the right of the function palette displays declared functions. Click on a function name to enclose the current selection(s) with that function.

Functions palette abilities are explained in detail in the Palette section of this manual.

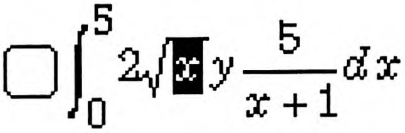
Example Sequence of Deletions

If you select an item in an expression and delete it, the item either disappears completely (if the selection is a string, a text string, or an N-ary selection), leaving an insertion point, or the item turns into a question mark. Deleting a question mark deletes its enclosing op, if any.

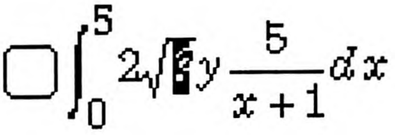
If an entire proposition is selected (including its icon), pressing delete removes the proposition and selects the proposition just above.

Pressing delete when there is no selection creates a new proposition at the bottom of the current notebook. Pressing delete again removes that proposition and selects the proposition just above. If the next proposition (above) is collapsed, it is not selected. To delete a collapsed proposition, you must select it explicitly.

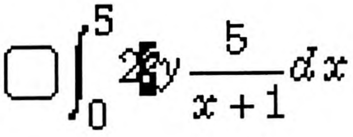
By repeatedly pressing delete, you can unravel and delete an arbitrarily large expression from the inside out. In the following figure, x is selected in the initial expression, then delete is pressed eight times.



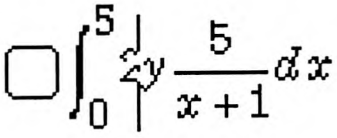
Selected x



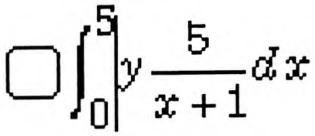
Deleted x



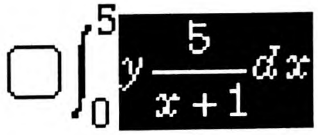
Deleted square root



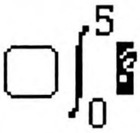
Blinking insertion point



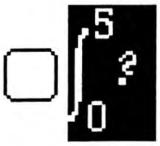
Deleted 2; blinking insertion point



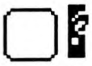
Integrand selected



Integrand deleted



Integral selected



Integral deleted

SELECTING AND EDITING OBJECTS

This section discusses how to select and edit the following objects:

- Propositions
- Expressions and sub-expressions
- N-ary expressions
- Names
- Matrices and vectors
- Comments

All editing is based on selections. Whatever you type is applied to or inserted into the current selection. In addition, all manipulations work on the current selection. This section discusses the different types of selections and the kinds of editing you can do with them.

Selections work differently in different programs. In word processors, you select a sequence of characters, or possibly a sequence of paragraphs. In spreadsheets, you can select a set of cells which are a subset of a large matrix. In drawing programs, you select a set of objects. In outliners, you select hierarchical trees of topics.

In Theorist, you have a mixture of all of these types of selections. Not only that, but there can be multiple simultaneous selections. Each selection can be one of several types. Editing and manipulations are done on all selections simultaneously (to the limits of plausibility).

There are four ways to make a selection with the mouse:

- Click
- Double-click
- Click-and-drag
- Double-click-and-drag

To create multiple selections, hold down shift as you click and drag or double-click. To un-select an expression in a multiple selection, shift-click within that expression.

You can not copy from a multiple selection, but you can paste into a multiple selection. Whatever you paste is replicated in all selections.

Selecting and Editing Propositions

Click on a proposition's icon to select that proposition and all of its outlining sub-topics (or daughters). Once selected, you can operate on all of them as a unit. These selections are useful for rearranging your propo sitions.

For example, if you press delete, the selected proposition and all of its daughters are removed. If you copy a proposition, all of its daughters are copied with it.

To duplicate a proposition, select it (by clicking on its icon), click ⧉ from the palette (or choose **Copy** from the **Edit** menu), then select another proposition and click 📋 (or choose **Paste**, also on the **Edit** menu). The copied proposition appears after the selected proposition. Using cut instead of copy works the same way, except the first proposition is deleted from its original position.

Selecting and Editing

To make an expression selection, click and drag within an expression. Only a well-formed subexpression can be selected. For example, you can not select a square root without also selecting its argument.

There are many types of "insertion point" selections. These are analogous to insertion point selections in a word processor; they mark a place for insertion. For all insertion points, the selection is indicated by a blinking vertical bar. All non-blinking selections enclose an expression.

Typing keystrokes and clicking on the palette inserts what you enter into the current selection(s). Op characters (mostly punctuation marks) always enclose the current selection (if it is not an insertion point). Name and number characters always replace the current selection.

Expressions can have the following types of selections:

- The entire expression
- A blinking cursor at the beginning
- A blinking cursor at the end
- Any type of selection in one or more subexpressions

You can use the mouse to click-and-drag to select any expression. A single click selects a single character name; a double click selects a multi-character name. You can also click on the distinctive part of an op to select the op and all its enclosed expressions:

- To select an equation, click on the equals sign.
- To select an integral, click on the integral sign.
- To select a fraction, click on the horizontal bar.
- To select an expression raised to a power, click just under the superscript.

There are some restraints on multiple selections. You can not select something within an existing selection, nor can you select something that encloses an existing selection.

For example, if you select 456 in the expression 123 + 456 + 789, and then try to select the whole sum of three numbers by shift clicking and dragging, instead of making two selections, the two selections cancel each other out. The same thing happens if you try these two selections in the opposite order (selecting the sum and then attempting to select the number). Nor can you click-drag select two terms (e.g., 456 + 789) and then shift click to add the single term 123, or click-drag to add just the 2.

However, you can select each of the three numbers 123, 456 and 789 individually without conflict. Similarly, you can click-and-drag to select part of each term (e.g., 2, 5, and 89). But individual numbers (e.g., 123) can not have two distinct selections.

Selecting N-ary Expressions

N-ary selections are selections that include one or more elements of a sum, product, or row vector. These expressions behave as if they were strings with each term acting as a single character. For example, in a sum of four terms, you can click and drag to select one, two, three, or all four elements:

$$1 + 2 + 3 + 4$$

You can also insert a blinking cursor between any of the elements (on top of one of the plus signs) or at the beginning or end of the sum. Use the arrow keys to insert the cursor at the beginning or end. If you select with a double-click, the selection is the individual term, not the sum. To select the entire sum, double click on one of the ops (the plus sign in this case), or double-click on one of the terms and start to drag across the terms.

Selecting Names and Numbers

Names and literal numbers in expressions behave as strings of characters like a typical word processor. For example, the name “right” has six different possible insertion points: one at the beginning, one at the end, and four between the characters. You can also click-drag to select one or more characters. To select the name as a whole, double click anywhere within it.

Selecting Matrices and Vectors

Matrices are composed of two or more elements. Each of these elements can contain whatever selections are appropriate for that expression. Matrices as a whole have three special types of selections:

- Sub-matrix of the matrix
- Vertical insertion points between columns
- Horizontal insertion points between rows

To select an entire matrix, double-click on one element and drag, after the second click, to any neighboring element. To select a sub-matrix, click on one element, and drag to another element. To make a vertical insertion point, click between two columns. To make a horizontal insertion point, click between two rows.

To delete one or more rows from a matrix, select the entire row(s) you want from left to right and press delete. Similarly, to delete one or more columns, select the entire column(s) from top to bottom and press delete. If you select a sub-matrix, each element is replaced with a question mark. If you press delete with a row (or column) insertion point selection, you delete the row (or column) to the left (or above) the insertion point, if any. If there is no row or column, delete selects the whole matrix. Pressing delete again removes the matrix.

To edit several matrix elements at once, click and drag over several elements to select them. Every keystroke, or palette entry, is applied to each of the selected elements. If you select the entire matrix (including its enclosing parentheses), whatever you enter replaces the matrix. To select all elements of a matrix individually, select the entire matrix then choose **Select In** from the **Edit** menu.

If you Paste an expression into a sub-matrix selection it is replicated in each element, unless the clipboard contains a matrix. If you paste a matrix (cut or copied from another matrix) into a sub-matrix selection, the matrix is pasted into the sub-matrix copying the elements one by one. If the copied matrix selection and the area you are pasting into have the same number of rows and columns, the elements are pasted in exact correspondence. If not, the pasted matrix repeats itself (if it is too short to fill up the selection) or terminates (if its too long).

For example, consider the following matrix:

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 \\ 16 & 17 & 18 & 19 & 20 \\ 21 & 22 & 23 & 24 & 25 \end{pmatrix}$$

If you copy the element 5 and then select the sub-matrix from 8 through 25 and paste, you produce this matrix:

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 5 & 5 & 5 \\ 11 & 12 & 5 & 5 & 5 \\ 16 & 17 & 5 & 5 & 5 \\ 21 & 22 & 5 & 5 & 5 \end{pmatrix}$$

On the other hand, if you copy the sub-matrix $\begin{pmatrix} 4 & 5 \\ 9 & 10 \end{pmatrix}$ and paste it into the 8 through 25 sub-matrix, you produce the following:

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 4 & 5 & 4 \\ 11 & 12 & 9 & 10 & 9 \\ 16 & 17 & 4 & 5 & 4 \\ 21 & 22 & 9 & 10 & 9 \end{pmatrix}$$

If you enter characters or paste a selection into a vertical or horizontal insertion point, one or more new rows or columns are created. Whatever you entered or pasted is inserted into the new elements as though they had previously existed and were selected before the paste.

If an entire matrix is selected, entering characters or pasting a selection replaces the matrix.

Matrices work easily with spreadsheet programs. Copy a selection of cells out of a spreadsheet, then select a matrix or sub-matrix to paste the cells into. If you select a whole matrix, the matrix adjusts to accept the cells from the spreadsheet as individual elements. If you select a sub-matrix, pasting works as with any other copy and paste operation.

Selecting Comments

A selection of text inside a comment is called a comment selection. You can have only a single selection in a comment. To edit comment propositions use cut, copy, and paste, and the delete key as you would in a word processor.

Selection Tools

The palette has five buttons for selecting expressions. They are especially useful for selecting individual elements, making multiple selections, and changing the selection.

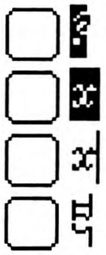
Palette selection tools are explained in detail in the Palette section of this manual. Their keystroke equivalents are given in Appendix A.

Examples of Expression Creation and Modification

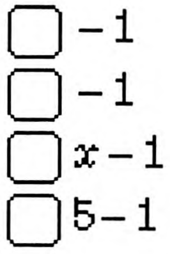
Create expressions by typing or selecting items from the palette. The expression is entered wherever there is a selection. The interaction of the type of the selection(s) and what you enter determines the way the expressions appear. This interaction is rather complex, because of Theorist's power and flexibility. This section provides a number of examples of creating and editing:

- Simple expressions
  - Numbers
  - Names
  - Ops
- Functions
- Linear operators
- Differential and Integral expressions
- Iteration expressions
- Matrices
- Other expressions

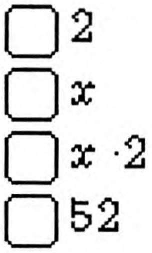
As an example of the interaction between types of selections and entered expressions, consider the following selection set:



With the above selections, typing **-1** produces:


−1
−1
x − 1
5 − 1

...whereas typing **2** produces:



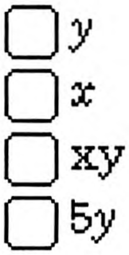
□ 2

□ x

□ $x \cdot 2$

□ 52

...and typing **y** produces:



□ y

□ x

□ xy

□ 5y

xy is new name, not x times y.

Typed characters are interpreted as letters, numerals, or ops. Letters (Greek and Roman) form names of functions, variables, and constants. Numerals form numbers. Ops connect and enclose names and numbers to make expressions. Numerals can not be used as part of a name except as subscripts.

Examples in this section assume that all necessary predefined names have been declared and their associated predefined behaviors have been accepted. For more information about predefined names and behaviors, see the Expressions section.

Unless otherwise noted, the Fortranish option is off.

Simple Expressions

The following examples show how to create simple expressions, including:

- Numbers
- Names
- Ops

Numbers

Numbers are created by typing in numerals from the keyboard, with one exception. To create a number in scientific notation, enter the base value followed by the character “e”. This produces a multiplication sign (×) followed by the number ten. Entered numbers become the exponent; entered letters are multiplied on as names.

To Create:	Type:	Comment
55	**55**	
–55	**-55**	The negation op encloses the number. Select a question mark before entering a negative value. Any other selection retains the selected expression and subtracts the number.
543.21	**543.21**	
.01	**0.01**	Enter the zero before the decimal to avoid confusion with the range op.
1.2×10^{6}	**1.2e6**	Use scientific notation.
1.2×10^{-4000}	**1.2e-4000**	Exponent can range between –4900 and +4900
1.23456789	**1.23456789**	Choose the display precision on Prefs menu. Nineteen digits are stored internally.
∞	**'N**	Infinity (a predefined name)
3+4i	**3+4i**	Complex numbers use the predefined name i
?	**?**	The question mark represents an empty or undefined value; select it and enter an expression.

Names

Names consist of one or more characters (Greek or Roman) typed in sequence. All capital and lower case Roman letters can be used. Use the apostrophe (') for the prime mark. Use two prime marks in a row for double prime (").

Numerals can *not* be used as part of a name, except as subscripts.

Enter Greek letters and special characters by typing apostrophe (') followed by the appropriate key, or select the character from the variables palette.

To Create:	Type:	Comment
x	**x**	Single character names are displayed in italics
sin	**sin**	Multi-character names are displayed in roman font
Re	**Re**	Names are case-sensitive; Re ≠ re
x'	**x'**	Use an apostrophe to create prime marks
ψ	**'y**	Use apostrophe and a letter to enter Greek characters
Ψ	**'Y**	Use apostrophe and Shift with a letter to enter upper-case Greek characters
$\aleph$	**'A**	Use apostrophe (and sometimes Shift) to create other symbols (see Appendix A)
g_h	**g_h**	Use the underscore character to create subscripts
g_{sun}	**g_sun**	Use any name to create a subscript
g_2	**g_2**	Use any integer (±32000) for subscripts
$\aleph_0$	**'A_0**	
k	**?k**	Enter a question mark and any lower-case character (a - z) to create wildcard variables.

Ops

Ops are usually entered as punctuation marks. Any selected expression is enclosed by the entered op.

To Create:	Type:	Comment
$x + 10$	**x+10**	Spaces entered in sums are ignored
$5x$	**5*x**	

$5x$	**5x**	Concatenation acts as multiplication between numbers and names
$5(3x + 1)$	**5*(3*x+1)**	Enter multiplication symbols (*) explicitly, if you want to
$5(3x + 1)$	**5[3x+1]**	Use parentheses or brackets to enclose an expression
$5(3x + 1)$	**5{3x+1}**	Or braces
$5(3x + 1)$	**5(3x+1]**	Pairs of parenthetic marks do not have to match
xy	**x*y**	
xy	**x** space **y**	Indicate multiplication between names with an asterisk (*) or a space; concatenating names creates a new name (xy not xy)
$x + \frac{1}{x} - 2$	**x+1/x** esc **-2**	
$x + \frac{1}{x} - 2$	**x+1/x-2**	Fortranish on
$\frac{x+1}{x-2}$	**(x+1)/x-2**	
$\frac{x+1}{x-2}$	**(x+1)/(x-2)**	Fortranish on
x^2	**x^2**	Basic syntax
$x^2 + 1$	**x^2** esc **+1**	
$x^2 + 1$	**x^2+1**	Fortranish on
$(1 + x)^2$	**(1+x)^2**	
x^{n+1}	**x^n+1**	
x^{n+1}	**x^(n+1)**	Fortranish on
x^{2n}	**x^2*n**	
x^{2n}	**x^(2*n)**	Fortranish on

$\sqrt{2}$	**sqrt(2)**	FORTRAN language syntax
$\sqrt{2}$	**\2**	Use the back slash character to enter square roots quickly
$\sqrt{x^2-1}$	**\(x^2-1)**	Fortranish on. Use parentheses to enclose the argument. If not, the root encloses only the x^2
$\sqrt{\frac{x+1}{x-1}}$	**sqrt((x+1)/(x-1))**	Use parentheses to specify the entire argument and the numerator and denominator
$n!$	**n!**	
$(2n)!$	**(2n)!**	Use parentheses to indicate a factorial expression with more than one element
$\frac{1}{n!}$	**1/n!**	Parentheses are not needed in this fraction; the factorial op binds tightly to the n
$\lvert x\rvert$	**|x**	Use a single vertical bar (|) to create an absolute value; don't type a closing vertical bar
$\lvert g_2-g_1\rvert$	**|(g_2 esc -g_1**	
$\lvert g_2-g_1\rvert$	**abs(g_2 esc -g_1**	FORTRAN language syntax
$x \bullet y$	**x*@y**	Use * @ to create a dot product
$x \times y$	**x*#y**	Use * # to create a cross product
$A^{\dagger}$	**Adjoint(A)**	

The following examples show how to edit simple expressions.

To Change This:	To This:	Do This:
$5x$	$5xy$	Select x, type ***y**

$5x$	$5\frac{x}{y}$	Select x, type **/y**
$5x$	$\frac{5x}{y}$	Select $5x$, type **/y**
$5x$	$5(x + y)$	Select x, type **+y**
$5x$	$5\sqrt{x}$	Select x, type ****
$5x$	$\frac{5}{x}$	Select x, Cut, Delete, type /, and Paste the x back in as the denominator

The minus sign does double duty as an unary minus op (–5) and as a sign indicating subtraction.

To Change This:	To This:	Do This:
x	$x - y$	Select x, type **-y**
$x + y$	$x - y$	Select y, type **-**
$x - y$	$x + y$	Select y or $-y$, type **-**
$x + y$	$x + y + z$	Select y or $x+y$, type **+z**
$x - y$	$x - y + z$	Select $-y$, type **+z**
$x - y$	$x - (y + z)$	Select y, type **+z**
$5x$	$-5x$	Select $5x$, type **-**
$5x$	$-5x$	Click before 5, type **-**
$5x$	$(-5)x$	Select 5, type **-**
$5x$	$5(-x)$	Select x, type **-**

Functions

The following examples show how to create various functions.

To create:	Type:	Comment
$\sin(x)$	**sin(x)**	The first parenthesis creates the function call op; the second escapes the function call op.
$\sin\left(\sqrt{\frac{x}{b_0}}\right)$	**sin(sqrt(x/b_0))**	The two closing parentheses escape the square root op and the sin function, respectively.

$\frac{\sin[\mathfrak{x}]}{\cos[\mathfrak{x}]}$	**sin(?x)/cos(?x)**	
$\sin^2 x$	**(sin(x))^2**	The form "$\sin^2 x$" is not recommended. This form changes to $(\sin(x))^2$ the next time you Simplify.
cosh(2x)	**cosh(2x)**	
log(x)	**log(x)**	log base 10 of x
ln(x)	**ln(x)**	natural log
$\log_2(x)$	**log_2** esc **(x)**	log base 2 of x
$\log_x(y)$	**log_x** esc **(y)**	log base x of y
e^{ikx}	**e^i*k*x**	

Linear Operators

To multiply non-commutative operators onto each other, use the standard technique for multiplication: type an asterisk (*) or a space. Matrices are M-Linear operators and derivatives are D-Linear operators, as are any operators you create and designate to be M- or D-Linear operators. D-Linear operators may not be commutative or associative: if A, B, and C are D-Linear operators, the expression $A*B*C$ is interpreted as $A(BC)$, which is *not* the same as $(AB)C$.

The following examples show how to create various Linear operators.

To Create:	Type:	Comment
$A\,x - \lambda\,x$	**A*x-'L*x**	Apostrophe plus letter for Greek
$A\,x - \lambda\,x$	**A** space **x-** option shift L space **x**	Optional Macintosh way
$A^{-1}BA$	**(A^-1)B*A**	
$A^{\dagger}BA$	**Adjoint(A)*B*A**	

Matrices

A matrix with a single row is displayed as a comma-separated list (a, b, c). Matrices with more than one row are displayed without commas.

The following examples show how to create various matrices and vectors.

To Create:	Type:	Comment
$(1, 2, 3)$	**(1,2,3)**	Use commas to separate elements
$\begin{pmatrix} a \\ b \\ c \end{pmatrix}$	**(a;b;c)**	Use semicolons to separate rows
$\begin{pmatrix} 1\,2\,3 \\ 4\,5\,6 \\ 7\,8\,9 \end{pmatrix}$	**(1,2,3;4,5,6;7,8,9)**	Use commas to separate elements and semicolons to separate rows
$\begin{pmatrix} 0 & i \\ -i & 0 \end{pmatrix}$	**(0,i;-i,0)**	
$\left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}\right)\begin{pmatrix} x \\ y \\ z \end{pmatrix}$	**Derivative(x)),Derivative(y)),Derivative(z))*(x;y;z)**	Type Derivative(to create a partial derivative
A_n	**A_n**	Single subscripts indicate a row of a matrix or an element of a vector
$A_{(n,m)}$	**A_(n,m)**	Two subscripts indicate a single element of a matrix
$(a, b, c)_2$	**(a,b,c)_2**	This expression simplifies to *b*

Differential & Integral Expressions

The following examples show how to create various calculus expressions. To create the partial differential op (∂) type **Derivative(**. To create an integral without limits (∫), type a dollar sign, **$**.

To Create:	Type:	Comment
$\frac{\partial}{\partial x} y$	**Derivative(x))*y**	Enter an asterisk (*) to explicitly indicate multiplication
$\frac{\partial}{\partial x} y$	**Derivative(x)** esc space **y**	A space can also indicate multiplication
$\frac{\partial}{\partial x} x^2$	**Derivative(x))^2**	A second closing) does the Escape

$\frac{\partial}{\partial x}\left(x^2+2x\right)$	**Derivative(x))(x^2+2*x)**	
dx	**d*x**	Multiplication with an asterisk (*)
$d\,(xy)$	**d(x*y)**	
$\int xdx$	**$x** space **d** space **x**	Enter an integral without limits before its argument
$\int_0^\infty xdx$	**Integral(x** space **d** space **x)** tab **0** tab **'N**	Enter an integral with limits after its argument
$5\int_0^\infty xdx$	**5*Integral(x*d*x):0:'N**	Use parentheses to indicate integral's argument
$\int_0^5 \sqrt{\frac{x-1}{x+1}}\,dx$	**Integral(sqrt((x-1)/(x+1))*d*x):0:'N**	

The following examples show how to create derivatives from various expressions.

To Change This:	To This:	Do This:
y	$\frac{\partial}{\partial x}y$	Click at left of *y;* type **Derivative(x)**
$\frac{\sqrt{y}}{z}$	$\frac{\partial}{\partial x}\frac{\sqrt{y}}{z}$	Click at left edge of the fraction; type **Derivative(x)**
$\frac{\sqrt{y}}{z}$	$\frac{\sqrt{\frac{\partial}{\partial x}y}}{z}$	Click at left edge of the *y,* type **Derivative(x)**
$\frac{\partial}{\partial x}$	$\frac{\partial}{\partial x}y$	Select the partial derivative; type ***y**
anything	$\frac{\partial}{\partial x}$ anything	Enclose anything in parentheses; click on the left parenthesis; type **Derivative(x)**

Iterations

The following examples show how to create various iteration expressions.

To Create:	Type:	Comment
$\sum_{n=0}^{5} a_n$	**Summation(n):0:5)a_n**	Summations are multiplied onto their arguments because they are D-Linear operators
$\sum_{n=0}^{5} \frac{x^n}{n!}$	**Summation(n):0:5)(x^n)/n!**	
$\prod_{n=0}^{5} a_n$	**PiProduct(a_n):n:0:5**	Pi Products enclose arguments rather than multiply onto them

Other Expressions

The following examples show how to create various expressions, some of which are useful for working with graph theories.

To Create:	Type:	Comment
$a ... b$	**a...b** *or* **a.b**	A single period between names is sufficient to indicate a range op
$-2 ... 2$	**-2...2** *or* **-2..2**	Use at least two dots to create a range op between numbers; the first dot is interpreted as a decimal point
$x = a - 2 \ldots a + 2$	**x=a–2..a+2**	
$a = b$	**a=b**	
$x = a - 2$	**x=a-2**	
$a < b$	**a<b**	
$a > b$	**a>b**	
$a \neq b$	**a** option = **b**	Optional Macintosh syntax
$a \neq b$	**a<>b**	Pascal syntax

$a \le b$	**a** option < **b**	Optional Macintosh syntax
$a \ge b$	**a>=b**	Pascal syntax
${}_{x=0}^{x=1}\left[\frac{x^6}{6}\right.$	**EvaluateAt((x^6)/6):(x=0):x=1**	
$\nabla = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}\right)$	**'Z=(Partial(x)),Derivative(y)**	

Tips and Techniques

This section lists a few tips and techniques for using Theorist. This is not the best place to start if you are new to the program, but if you are familiar with Theorist's basic operations, these tips should help increase your speed and efficiency. The topics are grouped into these sections:

- Algebra
- Numeric Integration
- Numeric Integration Preferences
- Symbolic Integration
- Graphs
- Editing
- Sending Notebooks via Electronic Mail
- Recovering Corrupted Notebooks

ALGEBRA

This section describes general algebraic techniques which are useful with Theorist.

Moving Negations

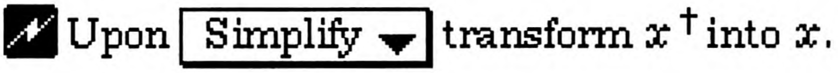
To move a negative sign from one side of an equation to the other, select the expression contained by the negative op (but not the negative op itself) and Isolate the expression to one side or the other.

Moving Factors

If a Move Over manipulation pulls over more than you want (perhaps a numeric coefficient), select the part that should not have moved, and use Move Over to drag it back.

Making a Real Variable

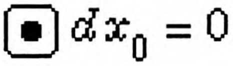
There is no direct way to declare that a variable is real (as opposed to complex). However, you can create transformation rules that will do the trick.



Upon Simplify ▾ transform $x^{\dagger}$ into x.

Making a Variable a Constant

If x is of class Variable, any subscripted name that uses x as a base is also a variable. However, you can make a subscripted variable name act as a constant with a working statement.



$d x_0 = 0$

Enclosing Expressions

To make a part of a sum or product act as a single unit, select the terms that should make up the subexpression and type an opening parenthesis, or simplify the expression in place.

Using Substitutions

Polynomials with non-simple terms (e.g., sin(x)) can not be factored with the Factor manipulation. For example, the following expression does not factor.

$$(\sin[x])^3 + 3(\sin[x])^2 + 3\sin(x) + 1$$

However, you can create a another expression, Substitute it into this expression and Factor the new expression.

$$\sin(x) = z$$

$$\square\ (\sin[x])^3+3(\sin[x])^2+3\sin(x)+1$$

$$\triangle\ (\sin[x])^3+3(\sin[x])^2+3\sin(x)+1 = z^3+3z^2+3z+1$$

$$\triangle\ (\sin[x])^3+3(\sin[x])^2+3\sin(x)+1 = (z+1)^3$$

Clever substitutions are also needed to factor an expression with pairs of terms. For example, the following expression does not factor in its current form.

$$(a-b)^3-6(-a+b)^2+12(a-b)-8$$

You could try to substitute an expression such as:

$$y = a-b$$

But this only substitutes into the two instances of $a - b$ (it will not substitute into $-a + b$). Instead, Isolate either a or b and use that.

$$a = b+y$$

Drag this equation over the first expression. This substitutes throughout, to give you an expression in y.

$$y^3-6y^2+12y-8$$

And this expression responds favorably to Factor.

$$\triangle\ y^3-6y^2+12y-8 = (y-2)^3$$

Factoring Multivariable Polynomials

Using the Factor manipulation on polynomials with many variables, or polynomials with symbolic coefficients, may not produce a satisfactory result. Sometimes the manipulation churns for a long time and accomplishes nothing.

In these situations, you can use Collect, Commute, and Simplify to step the polynomial to the form you want. For example, using Factor with the following polynomial produces a large and ugly result.

$$bx^2+x^2-ax-bx-abx-x+a+ab$$

The trick is to find the variable with the lowest, or one of the lowest, powers, and Collect on that variable. Commute the expression until *b* is the last factor of the first term.

$$x^2b+x^2-ax-bx-abx-x+a+ab$$

Then select the whole expression and Collect. This groups together terms that have *b* as a factor.

$$a+\left(a+x^2-ax-x\right)b+x^2-ax-x$$

The remaining terms are the same as the ones in the parentheses. Use Commute to rearrange them, then select these four terms and Simplify. This unifies them as a sum.

$$\left(a+x^2-ax-x\right)b+\left(a+x^2-ax-x\right)$$

Collect the whole expression to combine the two sums.

$$\left(a+x^2-ax-x\right)(b+1)$$

The polynomial in *a* can also be reduced. Use Collect to pull out the *a*, and then use Collect on the last two terms to pull out the *x*.

$$(-x+1)a+x^2-x$$

$$a(-x+1)+(x-1)x$$

Select both of these terms and use Collect again.

$$(a-x)(-x+1)$$

The complete “factored” result is:

$$([a-x][-x+1])(b+1)$$

This result can be cleaned up with Simplify.

$$(a-x)(b+1)(-x+1)$$

Note: These tricks do not work with all polynomials. Some just do not factor. Frequently, though, in algebraic derivations, you run into polynomials that can be factored, but it may take a little work with Commute, Collect, and Simplify.

NUMERIC INTEGRATION

This section describes why symbolic integration does not always supply an answer to a particular problem· of integration and introduces the concept of exact numeric integration. This section presents a description of:

- The pitfalls of symbolic integration
- Solving integrals numerically
- Some potential problems caused by round-off errors
- Calculating numeric integrals over the complex plane
- Working with numeric integrals of expressions that contain singularities
- Working with multiple numeric integrations

Symbolic Integration is frequently much more difficult than the simple examples presented in this manual. In fact, most expressions have no integral that we can write in terms of functions that we know. This has to do with the nature of differentiation.

When you take the derivative of a function, usually it gets larger (i.e., more complicated). For instance,

$$\square \frac{\partial}{\partial x} x^2 \sin(x) \cdot 2^x$$

$$\triangle \frac{\partial}{\partial x} x^2 \sin(x) \cdot 2^x = 2^x x^2 \cos(x) + \ln(2) \cdot 2^x x^2 \sin(x) + 2 \cdot 2^x x \sin(x) \quad \text{Simplify}$$

Imagine the space of all expressions. If every expression gets more complicated when you take its derivative, and if the number of complicated expressions is much larger than the number of simple expressions, it follows that there are plenty of left-over complicated expressions that are not the derivatives of some simple expression. Although a full treatment is beyond the scope of this manual, you can see that not all integration problems have an easy answer.

This is not to say that these expressions can not be integrated. For instance, the following integral can not be solved in terms of trig, hyperbolic, and exponential functions and anti-functions.

$$\int e^{-x^2} dx$$

If, however, you include in your repertoire of functions the Error function (one of the so-called "special functions"), you can write the answer simply:

$$\int e^{-x^2} dx = \frac{\sqrt{\pi}}{2} \operatorname{erf}(x)$$

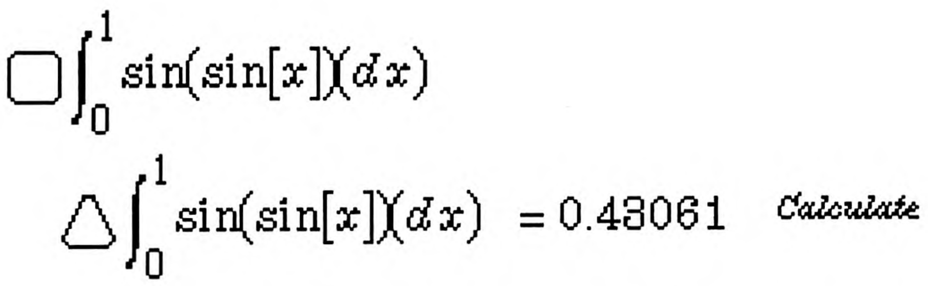
So really we are talking about whether the integral of the function can be written in terms of functions that we "know." The only reason we can integrate this function,

$$\int \frac{dx}{\sqrt{1+x^2}}$$

is because we know about the hyperbolic functions and their antifunctions.

Many years ago it was fashionable to invent new special functions for all the useful integrals that had no closed form solution, and to spend much effort trying to express one such function in terms of others (Abramowitz and Stegun—see Bibliography). With today's high speed computers and sophisticated quadrature algorithms, it is frequently easier to just do it all numerically and not worry about what somebody else called this function.

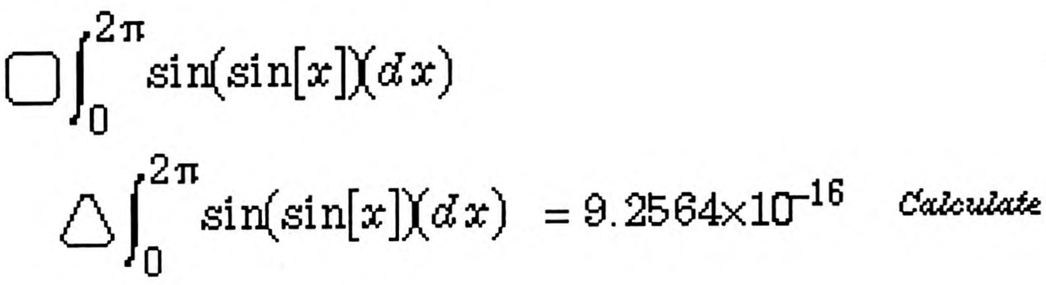
To integrate numerically, you must have a definite integral with two limits.



□ $\int_0^1 \sin(\sin[x])(dx)$

△ $\int_0^1 \sin(\sin[x])(dx) = 0.43061$ Calculate

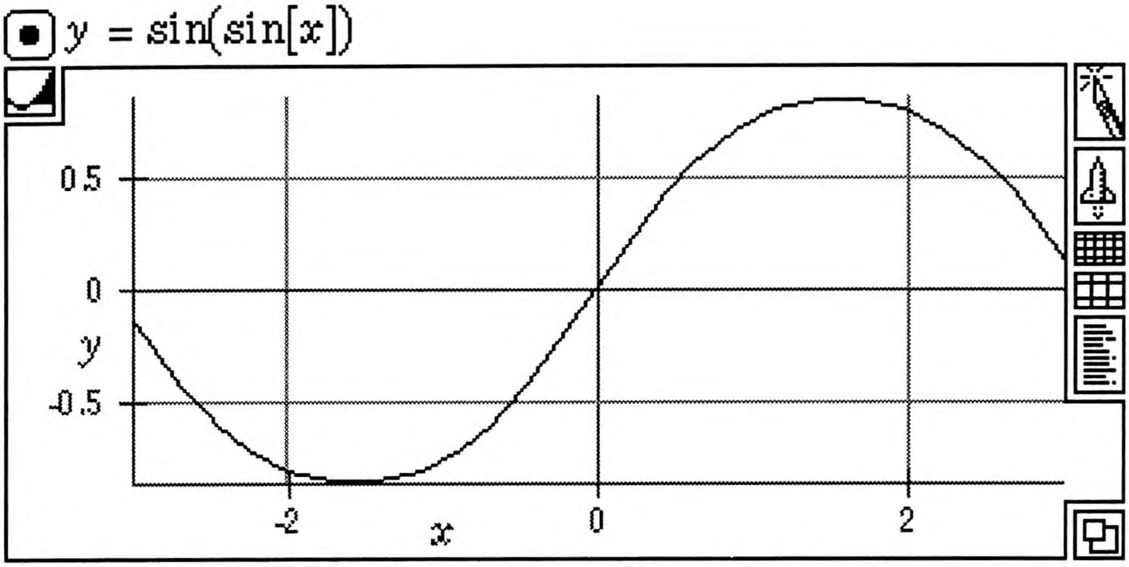
Answers are usually good to about 9 decimal places. (See the Numeric Integration Preferences section to learn how to adjust this.) There are exceptions, though. For example, the numeric solution for the following consists entirely of round off error.



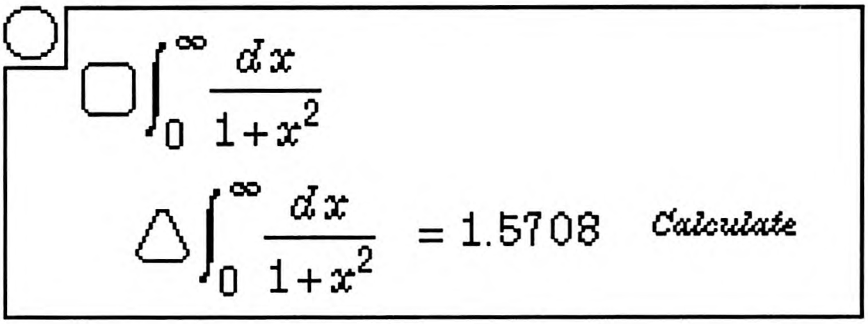
□ $\int_0^{2\pi} \sin(\sin[x])(dx)$

△ $\int_0^{2\pi} \sin(\sin[x])(dx) = 9.2564\times10^{-16}$ Calculate

This integral should evaluate to exactly zero because sin(sin(x)) retraces itself in the negative direction in the second half of each cycle, just like sin(x):



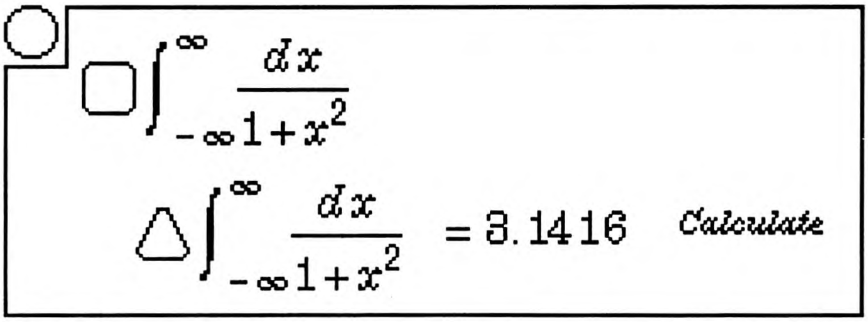
When you work with numeric integration, you can set the limits to infinity. For example:



○ □ $\int_0^\infty \frac{dx}{1+x^2}$

△ $\int_0^\infty \frac{dx}{1+x^2} = 1.5708$ Calculate

Or, setting both to infinity:

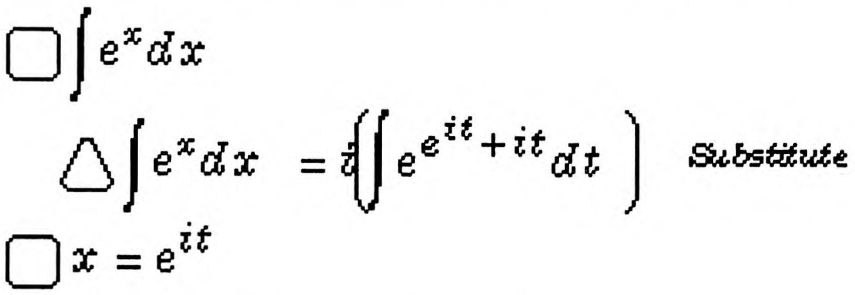


○ □ $\int_{-\infty}^\infty \frac{dx}{1+x^2}$

△ $\int_{-\infty}^\infty \frac{dx}{1+x^2} = 3.1416$ Calculate

All numeric integrals must have one real-valued variable to integrate over. If you want to have a path integral over the complex plane, you must somehow break up the path into sub-paths, integrate each individually over a single real variable parameter, and then add the partial results together.

For instance, the integral of e^x over the circle $|x| = 1$, can be calculated with one sub-path. The path is traced out by t as it runs from 0 to 2π.

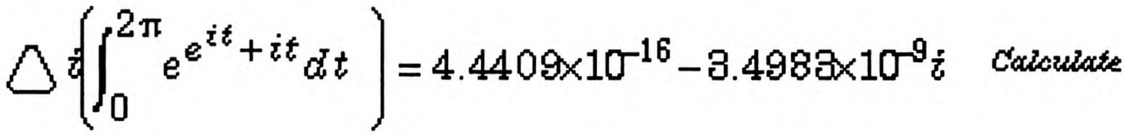


$$\square \int e^x dx$$

$$\triangle \int e^x dx = i\left(\int e^{e^{it}+it} dt\right) \quad \textit{Substitute}$$

$$\square\ x = e^{it}$$

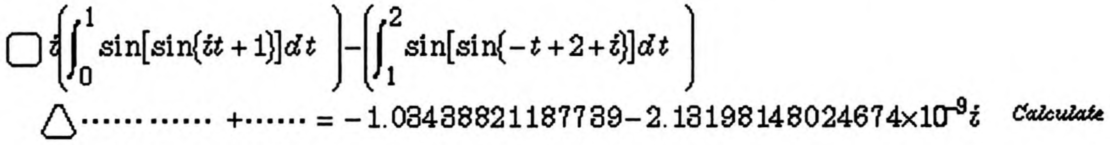
This becomes (with the limits filled in by hand):



$$\triangle\ i\left(\int_0^{2\pi} e^{e^{it}+it} dt\right) = 4.4409\times10^{-16} - 3.4983\times10^{-9} i \quad \textit{Calculate}$$

The solution is an approximate representation of zero. (By Cauchy's integral theorem this should come out to zero because it starts and ends at the same place and there are no poles.)

But if you want to integrate sin(sin[x]) over a path from 1 to 1 + i, and then from there to i, you could use the following technique.

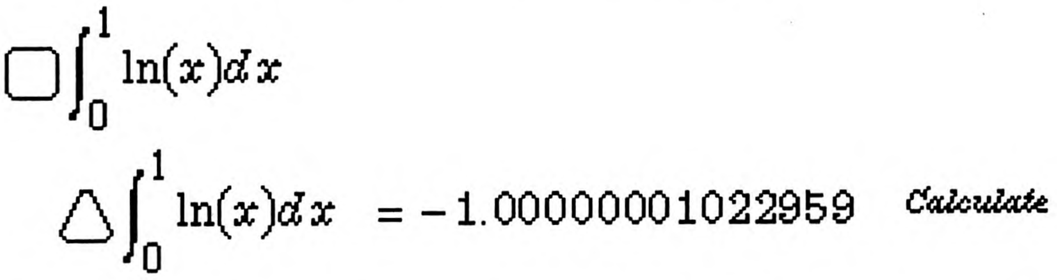


$$\square\ i\left(\int_0^1 \sin[\sin\{it+1\}] dt\right) - \left(\int_1^2 \sin[\sin\{-t+2+i\}] dt\right)$$

$$\triangle \cdots\cdots\cdots + \cdots\cdots = -1.03438821187739 - 2.13198148024674\times10^{-9} i \quad \textit{Calculate}$$

(The 10^{-9} part is numeric integration error.)

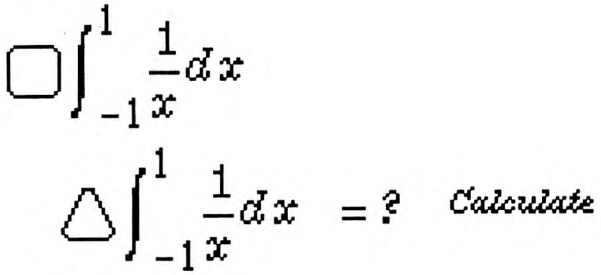
When you do numeric integrals, be careful if your integral has any singularities in it. In general, Theorist "feels out" sharp fluctuations in your function and uses smaller increments, but at a certain point it simply gives up. For instance, if you do this integral,



$$\square \int_0^1 \ln(x) dx$$

$$\triangle \int_0^1 \ln(x) dx = -1.0000000001022959 \quad \textit{Calculate}$$

...with maximum display precision, you get a very good answer. (The answer should be -1 exactly.) In that case the area is bounded. (It is the same area as e^{-x} from 0 to ∞.) On the other hand, the following integrals cause problems.



$$\square\int_{-1}^{1}\frac{1}{x}dx$$

$$\triangle\int_{-1}^{1}\frac{1}{x}dx \quad = ? \quad \textit{Calculate}$$

In both of these cases, the area under 1/x near zero is infinite. The integral tries to get this infinite area to cancel out with its rival; the numeric integration algorithm figured out that it was not continuous. (The real answer is undefined. Doing it algebraically depends upon which branch of the logarithm you take and is ambiguous.) If you are afraid of running into such a singularity, draw a graph of the function and search for singularities. Once you have found them, break up the integration path at each singularity.

Multiple Integrals

To do multiple numeric integrals with Theorist, you must remember to declare your variables to be independent. By default, Theorist assumes all variables are dependent upon each other in an unspecified way. By including an independence declaration (Notebook ▸Insert submenu), you can make these integrals simplify as they should.

When working with multiple integrals numerically, Theorist reduces some of its accuracy requirements to achieve reasonable execution times. Calculating integrals is essentially a process of repeated evaluation of the function. Although the points of evaluation and increment sizes are strategically chosen, there are still many points. For double integrals, the number of points is squared, and for triple integrals it is cubed. This can lead to unacceptable performance, so Theorist adjusts its algorithm. (See the Numeric Integration Preferences section for more details.)

Here is some advice for dealing with a difficult multiple integral:

❶ Evaluate it symbolically. Even if the whole thing can not be evaluated symbolically, sometimes integration in one or another variable can be done symbolically. You may want to reshuffle the integral, making the inside one the outside one and vice versa. Be careful with the limits; they probably will have to be changed also.

❷ Separate the integrand into parts that are independent of one or the other variable. If the two parts are added together, they are separate integrals. If they are multiplied, you can separate them into two integrals that are multiplied together, which calculates much faster.

❸ Change the coordinate system. For instance this integral,

$$\int_{-\infty}^{\infty}\left(\int_{-\infty}^{\infty} e^{-x^2-y^2}dx\right)dy$$

...can be evaluated by changing to polar coordinates. (Be careful with such a conversion; you must convert the differential in addition to the limits, and Theorist does not do it for you because the conversion of the limits is very subtle.) At that point, it can even be evaluated symbolically.

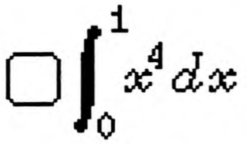
❹ Use some sort of approximation for the integrand.

Numeric Integration Preferences

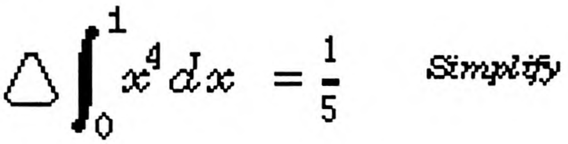
Theorist's algorithm used for calculating integrals is an adaptive procedure that automatically subdivides the interval in question for maximum precision, subdividing more in rough places than in smooth places.

The algorithm is controlled by five parameters that you can change in the numeric integration preferences dialog. This section describes what they do.

Simplifying the integral of x^4 from 0 to 1 produces the answer $\frac{1}{5}$.



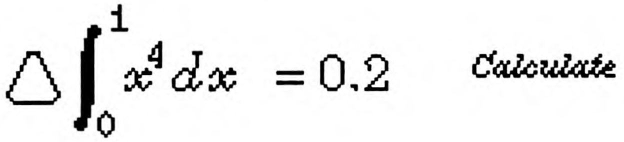
$$□ \int_0^1 x^4 dx$$



$$△ \int_0^1 x^4 dx = \frac{1}{5}$$

With the display precision set to the full 15 you can see it is an exact 0.2.

Calculating it produces the same answer, 0.2.



$$△ \int_0^1 x^4 dx = 0.2$$

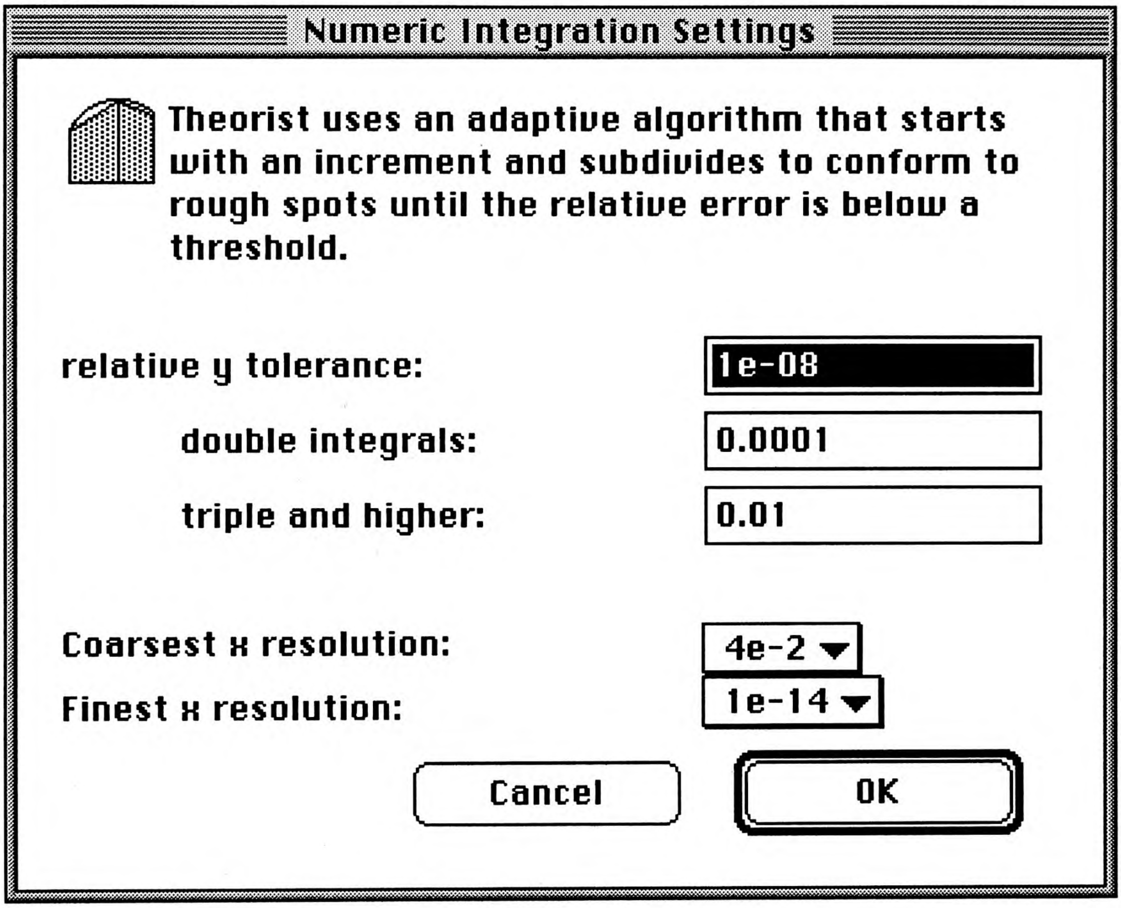
This is because the algorithm models each interval as a fifth order polynomial. Effectively, it finds the coefficients and then calculates the integral for that interval from the coefficients, which is a simple formula. It then tries subdivision to see if it can increase the accuracy.

If an integral is a polynomial with an order of five or less, not only is the algorithm exact, but it also is fast, because it quickly realizes that further subdivision is pointless. Not all the numeric integration settings affect the result. The preference controls are only useful for difficult integrals.

Calculating the integral of x^8 from 0 to 1 (an eighth order polynomial) produces the answer $\frac{1}{9}$ (which is 0.1111111...) but you will get some residual inaccuracy. (The exact answer you get depends upon the machine and the version of Theorist in use.)

There should be about nine 1's before the inaccuracy starts. By default, the relative error tolerance is 10^{-8}. For an answer that is about 10^{-1} the error should be less than 10^{-9}. In this case, it will probably be about a third of that. It is not unusual for the error to be much smaller than the tolerance, but you can not count on it being that small; you can only count on it being smaller than the tolerance. (But keep reading for some pathological cases.)

You can change this relative tolerance by choosing **Numeric Integration...** from the **Manipulate** menu's **Preferences** submenu.



Numeric Integration Settings

Theorist uses an adaptive algorithm that starts with an increment and subdivides to conform to rough spots until the relative error is below a threshold.

relative y tolerance: 1e-08

double integrals: 0.0001

triple and higher: 0.01

Coarsest x resolution: 4e-2

Finest x resolution: 1e-14

Change the default setting, $1e^{-8}$, to $1e^{-4}$ and press return. Then, Calculate the integral again and note that you get fewer digits of precision. It is hard, however, to predict what the error will be; sometimes the answer is much better than what the tolerance would suggest.

If you change the tolerance to 0.1, a very coarse setting, your answer is still good to about four digits. In fact, any tolerance from 0.5 (the maximum reasonable) down to 0.0006 gives you the same result. This is because the algorithm always does a minimal amount of sampling. By default, it does at least about two dozen samples even if it feels that the function is perfectly smooth.

This is controlled by the first pop-up menu, **coarsest x resolution**. If you set it to $1e^{-1}$ (from its default of $4e^{-2}$), and set the tolerance to 0.5, it instead does about eight samples and gives you an answer with just a few digits of resolution. The numbers $1e^{-1}$ (=0.1) and $4e^{-2}$ (=0.04) and the others are all approximately the fraction of the interval that is used as the starting increment. The actual locations evaluated are unevenly spaced and the separation varies from that shown on the pop-up, but this number should give you an approximate idea of what it is doing. You should adjust this setting upward if your integral is tricky and Theorist is fooled by its smooth appearance. You should adjust it downward, along with the epsilon, to speed integral calculation (e.g., if a graph uses it).

Delete all of your calculations and type **Integral(y*d*x)** tab **0** tab **1.**

$$\Box \int_0^1 y\, dx$$

On the next line type **y=x^2** esc **+Conditional(1,(x>0.7)(x<0.71); 0,1.**

$$\Box\, y = x^2 + \left(\begin{cases} 1 & [x > 0.7][x < 0.71] \\ 0 & 1 \end{cases}\right)$$

Choose **Graph▸y = f(x)▸Linear** to make a graph of it so you can see it looks just like a parabola. The graphing software was fooled by the conditional, which is a sudden pulse right at 0.50. Use the Knife to zoom in on the section near x = 0.70 through 0.71. It has height 1 and width 0.01 so the additional contribution to the integral is 0.01.

Set the numeric integration preferences to default $1e^{-8}$ tolerance and coarsest $1e^{-1}$. The value of this integral should be 1/3 plus 0.01 if done exactly. If you numerically integrate, you get the answer 0.333333333..., which shows that the extra pulse was ignored. Since there are only about eight samples, the algorithm completely stepped over the pulse and missed it. It would have got the idea to subdivide more, except the points it sampled all seemed like they came from x^2, which is perfectly smooth as far as it is concerned.

In the numeric integration preferences dialog, set the Coarsest setting to smaller values. Each time, select a character in one of the assumptions and re-type it to trigger recalculation. For instance, select the 2 from x^2 and type **2** so the equation remains the same (though Theorist thinks it is different). You will notice that you have to go to $1e^{-3}$. That means it starts with one thousand evaluations before it even decides to go deeper. Once you choose that setting, you will notice the time it takes to do the integral is much longer. There are two reasons for this:

❶ It takes longer to do 700 evaluations than 25 or 200.

❷ Once it "feels" that the pulse is in there, it will "feel around" and subdivide intervals in the neighborhood of the discontinuities in an effort to improve the resulting integral. It probably does another six hundred evaluations near the pulse.

And the answer, you will see, is .343333333..., meaning it correctly integrated the pulse.

Since the algorithm is designed for more continuous, analytic functions, it has a hard time with the step functions we put in. If the pulse were a Gaussian bell curve, for instance, the algorithm would be able to feel it from a distance even with a coarse setting. Once it feels something funny, it subdivides to get more resolution, and usually correctly discovers and carves its way around just about anything that is there.

At this point, you have adjusted the "coarsest" setting so Theorist notices the rough spot. To increase accuracy from here, change the "relative *y* tolerance." This makes the adaptive algorithm work harder. Tightening the Coarsest setting causes it to put more work into the smooth parts, which is pointless.

Get the *y* equation and change it to $y = 1/x$. Change the Numeric Integration Preferences back to their default settings (relative *y* tolerance 1e^{-8} and Coarsest = 4e^{-2}). If you try to integrate it, it will take a long time and return the value **?**. It returns **?** because it is a problem integral—it does not converge. The place where it does not converge is right at $x = 0$. The algorithm zeroes in on the problem spot and subdivides repeatedly in a vain attempt at reaching some conclusion. It only quits when it has reached the threshold indicated at the bottom of the numeric integration preferences dialog. By default it is set to 10^{-14}, the finest setting. This means that, if warranted, it subdivides the interval into pieces that are one 10^{14}th of the whole interval in order to achieve more accuracy, but finer than that, it gives up and returns **?**. (Note that the whole interval is not covered with increments that small; it uses such increments only in the immediate neighborhood of the problem. Just outside of that neighborhood, however, are increments just two or four times larger, and outside of those, larger ones, until you get out to the normal part again.) With $1/x$, it continued to subdivide because there was no solution, and it stopped at the Finest limit, after more than 2000 function evaluations.

To save time, you can restrict this to a more coarse setting. Set this to 10^{-5} and see how it takes much less time. At this setting this integral only evaluates about 600 times.

SYMBOLIC INTEGRATION

This section describes:

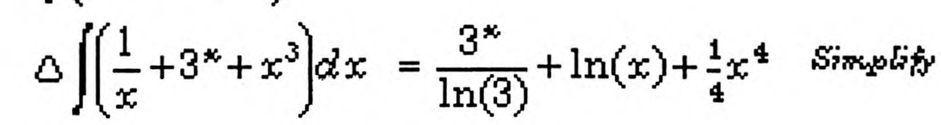
- Using the Simplify manipulation to solve elementary integrations
- Integrating linear and rational functions
- Integrating expressions that contain constants
- Using Independence Declarations in order to solve integrations
- Using the transformations in the New Notebook

All of the transformations in this tutorial use rules in the New Notebook. As long as you have the unmodified New Notebook file, as it came with Theorist, in the same directory as the Theorist program, choosing New Notebook from the File menu produces a notebook which includes these rules. It contains a useful subset of trig, log and hyperbolic rules, and other rules that are handy for integration. If you are going to be doing any serious integration, it is highly recommended that you start with that notebook. A more complete set of rules for transcendental functions can be found in the Transcendental Functions directory, in the Mathematics directory. A more complete set of rules for integration can be found in the Integration notebook in the Integral Tables directory. (Refer to Appendix C for a complete list of notebook directories and filenames.)

If you only need a few general rules of integration, the notebook named “Integration” may provide all the tools you need. It contains a few irrational integrals, a few exponential and logarithmic integrals, but only simple forms. All of the other notebooks cover specific categories of integrals in more detail.

One strategy that works well is to leave all of the integration rules in their separate notebooks and, when you need to do an integral, copy the integral out of your notebook, open one of the specific integral notebooks, paste your integral into it, solve it, then copy it back to your original notebook.

Many integrals can be solved automatically by Theorist when it simplifies an expression. For example, the following integral solution appears when you use the Simplify manipulation.



$$\square \int\left(\frac{1}{x}+3^{x}+x^{3}\right)dx$$

$$\triangle \int\left(\frac{1}{x}+3^{x}+x^{3}\right)dx = \frac{3^{x}}{\ln(3)}+\ln(x)+\frac{1}{4}x^{4} \quad \textit{Simplify}$$

Theorist can integrate all of the trig, arctrig, hyperbolic and antihyperbolic functions automatically. For example:

$$\square \int \sin(x)\,dx$$

$$\triangle \int \sin(x)\,dx \;=\; -\cos(x) \quad \textit{Simplify}$$

And:

$$\square \int \operatorname{arcsech}(x)\,dx$$

$$\triangle \int \operatorname{arcsech}(x)\,dx \;=\; x\operatorname{arcsech}(x)+\arcsin(x) \quad \textit{Simplify}$$

Theorist can also integrate functions of linear functions. For example:

$$\square \int \left(\sin[5x+7]+[ax-\pi]^{22}\right)dx$$

$$\triangle \int \left(\sin[5x+7]+[ax-\pi]^{22}\right)dx \;=\; \frac{1}{23}\frac{(ax-\pi)^{23}}{a}-\frac{1}{5}\cos(5x+7)$$

Theorist can also integrate any expression of the form $^{dx}\!/_{x}$ into a logarithm.

$$\square \int \frac{e^{x}}{e^{x}+5}\,dx$$

$$\triangle \int \frac{e^{x}}{e^{x}+5}\,dx \;=\; \ln\left(e^{x}+5\right)$$

You can integrate rational functions (a polynomial divided by another polynomial) by clicking on the integral sign and choosing Expand. Expand will do the extra manipulations needed to integrate these integrals. However, these manipulations may take some time. For example, the following integral requires three expands.

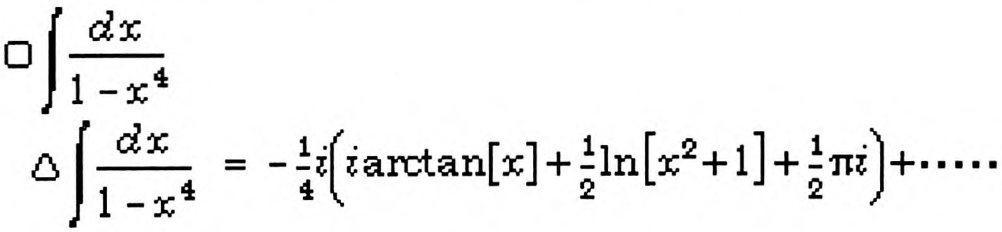
$$\square \int \frac{x^4+4x^3-7x^2+3x+14}{x^2-1}\,dx$$

$$\triangle \int \frac{x^4+4x^3-7x^2+3x+14}{x^2-1}\,dx \;=\; \frac{1}{3}x^3+2x^2-6x+\left(\int \frac{7x+8}{x^2-1}\,dx\right)$$

$$\triangle \int \frac{x^4+4x^3-7x^2+3x+14}{x^2-1}\,dx \;=\; \frac{7}{2}\ln\left(x^2-1\right)+\frac{1}{3}x^3+2x^2-6x+8\left(\int \frac{1}{x^2-1}\,dx\right)$$

$$\triangle \int \frac{x^4+4x^3-7x^2+3x+14}{x^2-1}\,dx \;=\; 8\operatorname{arctanh}(-x)+\frac{7}{2}\ln\left(x^2-1\right)+\frac{1}{3}x^3+2x^2-6x$$

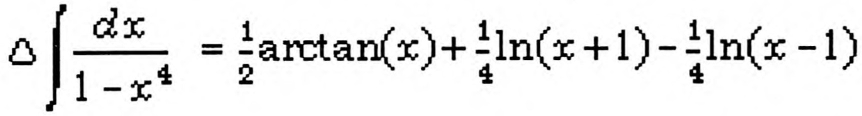
Some of these integrals (especially with larger denominators) may yield complex logarithms. The New Notebook has rules to digest these and to turn them into the arctangents they should be. For example, consider the following integral. Upon the first Expand, you get:



$$\square \int \frac{dx}{1-x^4}$$

$$\triangle \int \frac{dx}{1-x^4} = -\frac{1}{4}i\left(i\arctan[x]+\frac{1}{2}\ln\left[x^2+1\right]+\frac{1}{2}\pi i\right)+\cdots\cdots$$

And upon another Expand you get:



$$\triangle \int \frac{dx}{1-x^4} = \frac{1}{2}\arctan(x)+\frac{1}{4}\ln(x+1)-\frac{1}{4}\ln(x-1)$$

When you do integrals like this, you will often need to use the Expand manipulation one or more times to get the i's to multiply out. (To make the two logarithms combine, Collect, select the inner part, and Collect again.)

When you expand an integral like this, Theorist goes through these steps to integrate it:

❶ Expands the numerator and denominator to make polynomials
❷ If the numerator is higher order than the denominator, Expand to divide until this is no longer true
❸ Take remainder, Factor denominator
❹ Take fraction, Expand to do a partial fraction decomposition
❺ Integrate each piece

Because Theorist does this somewhat blindly, it can easily get lost and spend a lot of time factoring and expanding polynomials, until it runs out of memory, especially for higher order polynomials. Therefore, you may want to follow the above steps yourself to avoid unnecessary delays.

You can integrate many trig-heavy expressions with the substitution $y = \tan\left(\frac{x}{2}\right)$. Here's how to do it:

❶ Enter integral.
❷ Take any trig function (other than sin and cos) and transform into sin and cos.
❸ Enter $y = \tan\left(\frac{x}{2}\right)$.
❹ Solve for $x = 2\arctan(y)$. Substitute into integral.
❺ Carefully use Collect, Expand, and Simplify to make most or all of the factors of (y^2+1) go away, or at least reduce to a single factor. *Do not* just Expand or it takes a long time.

❻ When the integral gets more simple, it should be a rational function. Expand to solve it.

❼ Back-Substitute to get rid of *y*.

When you integrate functions, be sure all the other names used are either constant or are independent of the variable of integration.

The name *a* is declared as a constant.

$$\square \int \frac{a}{x} dx$$

$$\triangle \int \frac{a}{x} dx = a\ln(x)$$

Whereas *y* is *not* a constant, and *x* and *y* are not declared as independent.

$$\square \int \frac{y\,dx}{x}$$

$$\triangle \int \frac{y\,dx}{x} = \int \frac{y}{x} dx$$

Sometimes if you just change the form of an expression, it will integrate. For example, consider the following integral.

$$\square \int \frac{dx}{\sin(x)}$$

Transform the denominator to $1/_{\csc(x)}$. Then it is an easy Simplify.

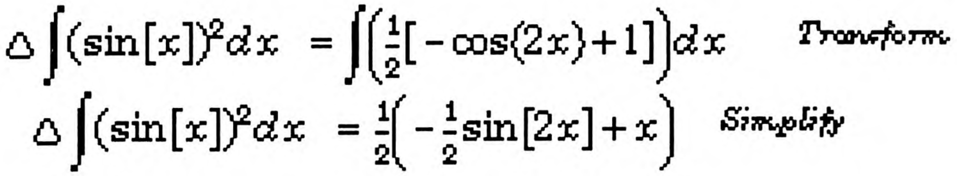
$$\triangle \int \frac{dx}{\sin(x)} = \int \frac{dx}{\frac{1}{\csc(x)}} \qquad \text{Transform}$$

$$\triangle \int \frac{dx}{\sin(x)} = \ln(-\cot[x] + \csc[x]) \qquad \text{Simplify}$$

Similarly, consider this integral:

$$\square \int (\sin[x])^2 dx$$

After transforming $(\sin[x])^2$, it simplifies easily.



$$\triangle \int (\sin[x])^2 dx = \int \left(\frac{1}{2}[-\cos(2x)+1]\right) dx \quad \textit{Transform}$$

$$\triangle \int (\sin[x])^2 dx = \frac{1}{2}\left(-\frac{1}{2}\sin[2x]+x\right) \quad \textit{Simplify}$$

Graphs

Subscripted variables

If a subscripted variable does not appear in the dialog box for selecting the dependent and independent variables to use for a graph, enter a minimum or maximum expression using the subscripted name and make the expression a working statement. For example, if z_a does not appear in the dialog box, enter an equation, $z_a > 0$.

Subscripted names often refer to matrix elements and, if a subscripted name is interpreted that way by the program, matrix elements are not shown in this dialog box as possible variables.

Balancing Bounds

To create a range of values for the one of the graph bounds centered on zero, select the range (click on the three dots), Select In twice and enter a value. This only works if the first range value is already negative and the second is not. You can also shift-click to select more than one range, and change them all at once.

Theories

A theory is a part of a notebook fenced off from the rest. All propositions outside it are in effect inside, but propositions inside the theory have no effect outside it. You can use theories to make multiple plots from the same equation. For an example of this, see the Spherical Family notebook from the Notebooks Disk supplied with the Theorist package.

Editing

Negating Expressions

To negate an existing expression, select it and type -. Just entering a minus sign op into an insertion point usually enters a minus sign and a new (empty) expression.

Editing on the Left

Many of the rules for keyboard entry work in reverse if you click on the left side of an expression. For example, if you click to the left of a variable and type +, a new question mark is added to the left side rather than the right.

You can enclose a name or number in a square root by clicking on its left side and typing a backslash. This works for all prefix ops (square root, absolute value, partial derivative, negate, integral).

Editing Matrices

To select all the elements of a matrix, double-click-and-drag on any element. That is, click twice on one element, and hold and drag the mouse after the second click to any other element. To select a row (or column), click and drag across all of the elements in that row (or column). To select more than one row (or column) click on an element in the top-right corner and drag to the element in the bottom-left corner, or vice versa. With these selections you can cut, copy, or delete the elements.

Removing a row (or column) leaves a blinking insertion point the length of the row (or height of the column). Pressing delete removes the next row above (or the next column to the left). You can also create a blinking insertion point by clicking between matrix elements. Rows (or columns) can be pasted into this insertion point. Single characters, or expressions copied from somewhere else, are pasted in as a set of elements, filling up the width of the row (or height of the column).

Creating Identity Matrices

An identity matrix contains a diagonal of ones, and all other elements are zero, such as:

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

To create an identity matrix:

❶ make a square matrix
❷ select all elements individually (select the matrix as a whole then use Select In)
❸ choose **Row/Column Number** from the **Notebook** menu's **Insert** submenu
❹ in the Insert Index dialog, choose Row Index starting at 1
❺ enter an equals sign
❻ choose **Row/Column Number** from the **Notebook** menu's **Insert** submenu
❼ in the Insert Index dialog, choose Column Index starting at 1
❽ select the entire matrix and use the Calculate manipulation

For smaller identity matrices, it may be easier to simply use multiple selections or type directly.

SENDING NOTEBOOKS VIA ELECTRONIC MAIL

You may want to use electronic mail to share your Theorist notebooks (or just individual equations or expressions) with friends and colleagues. Unfortunately, most contemporary electronic mail systems assume you are sending human-readable text, and therefore feel free to make small changes that a human could supposedly tolerate, such as adding introductions and trailers, stripping off high bits from characters, adding or removing spaces here and there, etc. For most kinds of non-textual files, you can not just paste the data or transmit a file into your e-mail message directly. In many cases, there are ways built into the e-mail systems to move such files, but this is not necessary with Theorist.

Theorist notebook files are actually textual and the data is the same as the data that you get when you copy and paste. You can send this data over e-mail and it will come out intact on the other end for a friend to copy into Theorist or to open as a notebook.

To send just an equation, select the equation, choose **Copy** from the **Edit** menu, and go to your terminal emulator program and Paste. If you get an equation in the mail like this, simply take the text, copy it out of your terminal emulator, and paste it into Theorist.

You can send equations, or small subexpressions, or whole propositions, or a set of indented propositions that you copy out of your notebook. Anything that you can copy out, using the normal Copy command, will work. (Do *not* use Copy as Picture—this will yield a binary graphic that will not go over e-mail easily.)

You can even send a whole graph, but you should be careful because most graphs depend on the equation(s) they were built upon, and whoever gets your graph will want that information, too.

To send a whole notebook, first save it in a file. (On a Macintosh, you should hold down the Option key as you choose Save, so the file will be saved as type TEXT so you can open it with a word processor or terminal emulator.) Then, just send the text file with your terminal emulator program.

When you receive the data, you can save the whole e-mail message in a file, or just the notebook part, if you want. The notebook data starts with a ~v and ends with a ~e. When Theorist reads a notebook file, it skips all text until it finds a ~v, and then only reads until the ~e, and ignores the rest. As long as the extra text before the notebook data has no tilde characters in it (or if it does, make sure there is no v immediately following it), you can open the whole file directly.

Then, open the file by choosing **Open** from the **File** menu. (On a Macintosh, to open a TEXT-type notebook, hold down the Option key as you choose Open.) Theorist should read the notebook in without problems. If there are any problems, see the next section's tips.

You should be careful copying and pasting text through a terminal emulator if you are using a high speed modem. Sometimes, some part of the system will not be able to keep up and text will be lost, sometimes several lines worth hidden somewhere in the middle of your data. It is best to experiment first with some text document that is about 10K in size, sending it up, and retrieving it back, and comparing with the original to see if any data was corrupted. A better way is frequently to use xmodem or its variants ymodem or zmodem (do a **man -k xmodem** on Unix systems).

Recovering Corrupted Notebooks

Theorist file storage is reasonably robust. It is designed to be able to read in your notebook even if it was generated by an older version of Theorist, or if it has been sent through electronic mail.

Occasionally, you may have a situation where a Theorist notebook has become corrupted and Theorist tells you that it simply can not reconstruct the data. There are a number of measures you can take to try to recover parts of your data. Keep in mind, however, the best way by far is to make frequent backups.

The measures in this section are only for emergency situations. You should not be doing them on a regular basis. Notebooks that have been cut and pasted together as described in this section are not guaranteed to work as documented in the manual.

On a Macintosh, the first thing to try is to open the file with the Option key held down. This will activate parts of the code that are more lenient and will increase your chances of successfully loading the data.

If that does not work, you can try opening the file with a text editor. (On the Macintosh, you will probably have to use an editor that ignores the file type, such as BBEdit, or else you must change the file type to TEXT with ResEdit or a similar tool.)

Once you are looking at the data, your equations will be in front of you in textual form, roughly the same as when you copy and paste them, probably near the bottom. In addition, there will be a lot of other data that you may or may not need.

Open a new notebook in Theorist and start copying and pasting sections out of the text editor into Theorist. Frequently, the error is just in one location, so you might be able to grab large sections of the file and paste them in without problems.

You should grab pieces that are self-contained. Equations are stored textually in a format that is the same as when you copy and paste them: they should be recognizable and you should copy out whole equations or expressions at once.

The rest of Theorist notebook data consists of commands that start with ~ and a letter. ~v signifies the beginning, and ~e signifies the end. Capital letter combinations are more significant. A good strategy is to copy a section of text starting at one tilde-capital and ending before another.

Graphs are quite complex. They start with a ~G or ~Z, and end with a ~t. Individual blocks of comment text start with]|Expr|[and end with]|[. You can copy such a block out and paste it into a comment, or into Expressionist, to recreate only that text. Do *not* paste such data directly into a Theorist assumption.

Once you have recovered your data, save it safely in your new notebook. Theorist should always be able to read in a notebook that's been freshly saved.

Remember that, if you open a file, and it gets an error, the notebook will not open. But, if you paste in data, and it gets an error, the part that was pasted in successfully will remain in.

Menus

This section provides a list of all menu commands in Theorist with a brief description of what each option does. All menu items execute a command immediately, branch to submenus, or bring up a dialog box. Any menu option that ends in an ellipsis (...) brings up a dialog box. Items which end with a small arrow open submenus, and options which begin with a small arrow are submenu items.

The Theorist menus are:

- File (for creating, opening, closing, saving, and printing notebooks and stationery files, and exiting Theorist)
- Edit (for editing equations, expressions, comments, propositions, and graphs)
- Notebook (for arranging and working with propositions and notebooks)
- Manipulate (for evaluating, rearranging, and exploring expressions)
- Graph (for making and modifying graphs, and for extracting data from graphs)

See Appendix A for a handy reference for keystroke equivalents of menu items.

File Menu

New Notebook

Creates a new notebook with the default propositions as saved in the "New Notebook" (Macintosh) or "NewNote.The" file (Windows).

Open Notebook...

Brings up the file selection dialog, asking for a location and name of an existing notebook file.

Close Notebook

Closes the current notebook window, as if you clicked on the close box. If there are any un-saved changes, you are asked whether you want to save the file.

Save Notebook

Saves the current notebook window in its file. If it is a new untitled notebook (i.e., it has not yet been saved), you are prompted for a file name.

Save as...

Brings up the file selection dialog, asking for a location and name for saving the current notebook window in a file. Any file previously associated with the current notebook remains as most recently saved.

Auto Save

When this item is checked, Theorist periodically saves the current notebook window in its file.

Revert to Saved

Disposes of all changes to the current notebook since it was last saved or opened. You are asked to confirm this command.

Page Setup... *or* **Printer Setup...**

(The menu item is named Page Setup on Macintosh or Printer Setup on Windows.) Brings up a dialog for you to change or verify the print settings. The dialog varies depending upon the type of printer and printer driver you are using.

Print Notebook...

Prints the current notebook. A dialog box appears with printing options. The dialog varies depending upon the type of printer you are using.

Quit *or* **Exit**

(The menu item is named Quit on Macintosh or Exit on Windows.) Quits Theorist, first prompting you to save changed files, if any. Any startup notebooks that opened automatically are saved automatically.

EDIT MENU

Undo

Reverts the current notebook to its state before the most recent change. Some changes can not be undone.

Cut

Copies the current selection to the clipboard, then deletes the selection. Expressions are stored as text in the Fortranish format.

Copy

Copies the current selection to the clipboard. Expressions are stored as text in the Fortranish format.

Copy as Picture

Copies the current selection (proposition, expression, or graph) to the clipboard as a picture. Selected expressions are stored as MetaFiles (Windows) or PICTs (Macintosh), with embedded information to reconstruct them in Expressionist. Graphs are copied as graphic images as specified in the graph preferences.

Paste

Pastes the contents of the clipboard, replacing, or just after, the current selection. If there is no selection, the clipboard is pasted at the bottom of the current notebook.

Clear

Deletes the current selection; the same as pressing Delete on the keyboard.

Select In

Changes each current selection to a multiple selection, selecting each subexpression of the current selection. Hold down Shift to select only the first subexpression.

Select Out

Changes each current selection to a selection of the enclosing expression or proposition. If two or more selections collide, they undergo pairwise annihilation; that is, if an even number of selections collide, no selections are left, but if an odd number of selections collide, one selection remains.

Comment Font...

Brings up a dialog to set the font, size, and style of the current comment selection. This command only works on comment selections.

Comment Style▸

This submenu includes items for changing the style of the current comment selection to plain, bold, italic, or underline.

Comment Color▸

This submenu includes items for changing the current comment selection to one of fourteen predefined colors, or from a palette of all available colors.

Notebook Menu

Clarify Scans the current notebook for any syntax errors, undeclared names, or other problems and re-generates all graphs from current expressions.

Windows▸ This submenu contains a list of all open notebooks. Choosing an item from this list makes that notebook the active window.

Expose Un-collapses the selected item(s). It exposes all daughter propositions of the selected proposition or exposes an expression collapsar for view.

Expose All Un-collapses all propositions and all collapsars in the current notebook.

Collapse Collapses the selected item(s). It collapses all daughter propositions beneath a proposition or collapses an expression into a collapsar.

Collapse All Collapses all propositions in the current notebook. Collapse All does *not* collapse any expressions.

Indent Left Moves each selected proposition one level to the left. If a selected proposition is an inline proposition, it becomes a daughter proposition. The selections can be all or part of a proposition or expression.

Indent Right Moves each selected proposition one level to the right. If a selected proposition is already a daughter proposition, it becomes an inline proposition. The selections can be all or part of a proposition or expression.

Insert▸ The **Insert** submenu provides commands for inserting propositions and parts of expressions.

▸Row/Column Number Inserts an integer into the current selection (if the selection is some or all of the elements of a matrix) that is the row or column number of the matrix element. If a selection is inside nested matrices, the outermost matrix is used to determine the row or column number.

▸Page Break Inserts a page break proposition icon, which forces subsequent propositions to appear on a new page when the notebook is printed.

▸Case Theory Creates a new Case Theory that encloses all currently selected propositions (if any) and all daughter or dependent propositions (if any).

▸Transform Rule Inserts one or more new transformation rule proposition(s). If one or more equation is selected, a transformation rule is created for each equation. The left side of each equation becomes a pattern and the right side becomes a replacement.

▸Independence Decl.

Inserts a new Independence Declaration after the current selection or at the bottom of the current notebook if there is no selection.

Make Working Stmt

Turns the currently selected assumption(s) or conclusion(s) into working statements. Any existing working statements that conflict with the new one(s) stop being working statements. The selections can be all or part of a proposition or expression.

Stop Working Stmt

Changes selected working statement(s) to regular statements. If the statement is required to be a working statement (e.g., for a graph), the statement (or one like it) becomes a working statement again immediately. The selections can be all or part of a proposition or expression.

Get Info

For each selected object, this command displays a dialog that describes the object and its relationship to other objects. You can get information about any proposition, name, number, expression, or graph theory.

Display Precision▸

The items on this submenu let you set the total number of decimal digits, counting from the leading digit, shown on the screen for numerical values. This preference affects display only; the full 15 digit precision is maintained internally. (Default: 5 digits)

Comment Rulers...

Brings up a dialog which lets you set the units of measurement for comment rulers, set the default ruler size, and determine whether rulers are shown in the notebook or hidden.

Notebook Font...

Brings up a dialog which allows you to sets the font, size, and style characteristics to be used for displayed expressions and comments.

Preferences▸

The items on this submenu control various notebook preference options.

▸Fortranish

Determines whether keyboard input follows FORTRAN precedence rules explicitly. (Default: off)

▸Palette AutoToggle

If checked on, the palette flips between the functions side and the variables side, anticipating your next click. If unchecked (off), the palette toggles only when you click on its icon. (Default: on)

▸Enter Key...

Brings up a dialog which lets you define the functions of the Return and/or Enter keys.

▸Show Icons

Shows or hides all proposition icons and graph buttons. (Default: on)

▸Show Steps

Shows or hides all manipulation steps to the right of conclusions. (Default: on)

▸Print Header...

Brings up a dialog which lets you specify the information to be printed on each page of the notebook.

MANIPULATE MENU

Calculate Evaluates an expression to a number like a pocket calculator.

Simplify Executes a wide range of operations designed to reduce an expression to a simple canonical form.

Expand Executes a range of operations designed to expand an expression to its most explicit form.

Collect Executes several operations designed to group similar parts of an expression together.

Factor Executes several operations designed to dismantle an expression into its most fundamental parts or terms.

Isolate “Solves” an equation for a selected expression.

Apply Selects each side of a selected equation (or the numerator and denominator of a selected fraction) so that all subsequent manipulations are applied equally to both sides. Can also augment expressions in many other ways without destroying an expression’s logical integrity.

Transform Executes all transformation rules set to execute “Upon Transform.” Transform is designed as a means for you to extend Theorist’s capabilities by identifying expressions of particular forms and turning them into different forms.

Other▶ The items on this submenu invoke special manipulations.

▶UnCalculate Undoes the effects of the Calculate manipulation (if possible), eliminating round-off error and symbolically reconstructing numeric values.

▶MiniExpand Executes the Expand manipulation on the outer-most layer of an expression.

▶Substitute Replaces expressions of a particular form with an equivalent expression of a different form.

▶Taylor Series Generates a Taylor series of a continuous function.

▶Int. by Parts Integrates the selected part of a larger integral.

▶Commute Rearranges terms in a sum, or factors in a product or dot product.

▶Move Over Moves an expression from one side of an equation to the other.

Find Graph Root

Finds the numerical value of the horizontal coordinate that zeros the vertical coordinate of a Line plot in a 2-D graph theory. Also finds the numerical value of the horizontal and vertical coordinates that simultaneously zeros two Contour plots in a 2-D graph theory. In both cases, values for all variables are presented in a separate case theory.

Table ▸

The items on this submenu invoke special table manipulations.

▸Generate...

Brings up a dialog which allows you to create a new table by evaluating the selected expression, varying the value of a variable.

▸Integrate Differential Equation...

Brings up a dialog which allows you to integrate an initial-value ordinary differential equation.

▸Fourier Transform

Creates a new table which is the Fourier transform of the selected table.

▸Inverse Fourier Transform

Creates a new table which is the inverse Fourier transform of the selected table.

ReManipulate Now

Repairs conclusions which need to be evaluated because one or more assumptions have changed.

Always ReManipulate

When this item is on, Theorist always repairs conclusions which need to be evaluated because one or more assumptions have changed.

Divert Cascade

Reattaches a broken cascade of conclusions, replacing the selected problem conclusion with the selected replacement conclusion.

Preferences▸

The items on this submenu control various manipulations preference options.

▸Auto Simplify

When this item is on, Theorist executes the Simplify manipulation (up to ten times or until the expression stops changing) after executing any other manipulation other than Apply and Commute. (Default: on)

▸Auto Casing

When this item is on, Theorist forces manipulations to generate separate case theories or arbitrary constants to display all results of a manipulation that generates multiple values. (Default: off)

▸Auto Collapsar

When this item is on, Theorist displays all expressions longer than the screen width as collapsars. (Default: off)

▸Arbitrary Constants...

Brings up a dialog to change the settings for arbitrary constants and variables.

▸Numerical Integration...

Brings up a dialog to change the parameters which affect the speed and accuracy of numerical integration.

Graph Menu

y = f(x) ▶
Each command on this submenu creates a graph with one dependent and one independent variable.

▶Linear
Creates a rectangular 2-D graph with a linear coordinate system.

▶SemiLog
Creates a rectangular 2-D graph with linear horizontal coordinates and logarithmic vertical coordinates.

▶SemiLog-X
Creates a rectangular 2-D graph with logarithmic horizontal coordinates and linear vertical coordinates.

▶Log-Log
Creates a rectangular 2-D graph with logarithmic horizontal and vertical coordinates.

▶Polar
Creates a rectangular 2-D graph with a polar coordinate system.

▶Complex 3D
Creates a rectangular 3-D graph containing a colored surface plot depicting a complex function on the complex plane.

z = f(x, y) ▶
Each command on this sub-menu creates a graph with one dependent variable and two independent variables.

▶Density 2D
Creates a rectangular 2-D graph displaying the dependent variable as a range of gray values.

▶Contour 2D
Creates a rectangular 2-D graph displaying the dependent variable as a selection of labeled contours.

▶Color 3D
Creates a rectangular 3-D graph containing a colored surface plot.

▶Illum. 3D
Creates a rectangular 3-D graph containing an illuminated surface plot.

▶Spherical 3D
Creates a 3-D graph with a spherical coordinate system containing an illuminated surface plot.

▶Cylindrical 3D
Creates a 3-D graph with a cylindrical coordinate system containing an illuminated surface plot.

Other▶
This sub-menu contains commands to create special graphs.

▶x = f(t), y = g(t) Parametric
Creates a 2-D parametric plot from the selected expression, with two dependent variables being functions of one independent variable, the parameter.

▸x = f(t), y = g(t) , z = h(t) Space Curve

Creates a 3-D parametric space plot, from the three selected equations, with three dependent variables being functions of one independent variable, the parameter.

▸f(x, y) = g(x, y) Implicit

Creates a 2-D implicit plot from one selected equation which has two variables in it. (They can appear on either side or both sides, in any arrangement.)

Scatter ▸

Commands on this submenu create scatter plots of different types.

▸Linear

Creates a scatter plot graph with a linear coordinate system.

▸SemiLog

Creates a scatter plot graph with linear horizontal coordinates and logarithmic vertical coordinates.

▸SemiLog-X

Creates a scatter plot graph with logarithmic horizontal coordinates and linear vertical coordinates.

▸Log-Log

Creates a scatter plot graph with logarithmic horizontal and vertical coordinates.

▸Polar

Creates a scatter plot graph with a polar coordinate system.

▸3D

Creates a 3-D scatter plot graph.

Additional ▸

The items on this submenu add various graph plots, axes, or grids to an existing graph.

▸Add Line Plot

Adds a line plot to an existing graph theory.

▸Add Surface Plot

Adds an illuminated surface plot to an existing graph theory.

▸Add Scatter Plot

Adds a scatter plot to an existing graph theory.

▸Add Contour Plot

Adds a contour plot to an existing graph theory.

▸Add Axes

Adds a set of axes to an existing graph theory.

▸Add Grid Lines

Adds a set of grid lines to an existing graph theory.

Animation ▸

Contains commands to start, stop, or save an animation.

▸Start

Begins animation.

▸Stop

Ends animation.

▸Save...
(Macintosh only)

Brings up the file selection dialog, asking for a location and name for saving the animation in a file. If QuickTime is installed, the file is saved in the QuickTime format, otherwise the file is saved in the PICS format.

Generate EPS File...

Creates an Encapsulated PostScript file from a graph. On the Macintosh, it is a EPSF type file, but if you hold down the option key while choosing this item the file type is TEXT.

Generate PICT File...
(Macintosh only)

Creates a PICT file from a graph. The picture format is identical to that created with **Copy as Picture** on the **Edit** menu, but rather than putting the image in the clipboard, this command brings up the file selection dialog to create a named file.

Generate MetaFile...
(Windows only)

Creates a MetaFile from a graph. The picture format is identical to that created with **Copy as Picture** on the **Edit** menu, but rather than putting the image in the clipboard, this command brings up the file selection dialog to create a named file.

Always Redraw

Ensures each graph is always redrawn whenever some condition affecting it changes. (Default: on.)

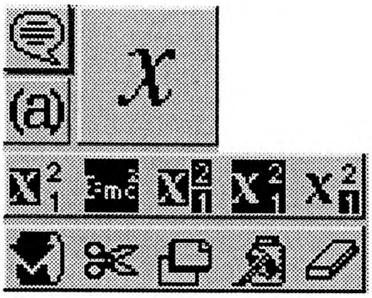
Preferences...

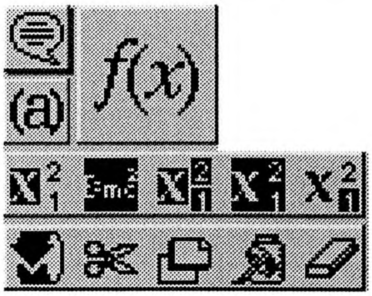
Brings up a dialog with animation options and options for graph display, printing, and exporting.

Palette

Shared Palette Buttons

These buttons are common to both the functions palette and the variables palette. They are in the same locations on both palettes.



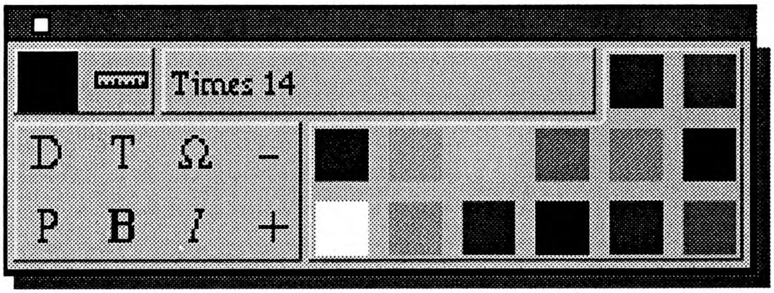


Switch Palette *x* *f*(*x*)

The large button on the left side of the palette, labeled *x* or *f*(*x*), switches between the variables palette and the functions palette.

Comment Controls pop-up

Pops up a sub-palette containing various comment control buttons. Release the mouse button with the pointer on the desired icon to perform its function.



comment controls pop-up palette

Ruler

(On the Comments pop-up sub-palette.) Shows or hides comment rulers.

Colors

(On the Comments pop-up sub-palette.) Each color button changes the color of the comment selection to the desired color.

Font Specs *Times 14 i*

(On the Comments pop-up sub-palette.) Brings up the font/size/style dialog to change the specifications of the selected comment text.

Default Font D

(On the Comments pop-up sub-palette.) Changes the font, size, and style specifications of the selected comment text to the default font (Notebook Font) family and size, in the plain style.

Default Font Family T

(On the Comments pop-up sub-palette.) Changes the font family of the selected comment text to the default font (Notebook Font) family, without affecting size or style characteristics. The character used as this button's icon changes with the Notebook Font; it uses the first letter of the Notebook Font family name (e.g., T for Times and B for Bookman). If the font family name does not start with an upper case English letter it uses the letter F.

Symbol Font Ω

(On the Comments pop-up sub-palette.) Changes the font family of the selected comment text to Symbol.

Plain Style P

(On the Comments pop-up sub-palette.) Changes the font style specifications of the selected comment text to plain.

Bold Style **B**

(On the Comments pop-up sub-palette.) Applies the bold font style to the selected comment text.

Italic Style *I*

(On the Comments pop-up sub-palette.) Applies the italic font style to the selected comment text.

Decrease Size –

(On the Comments pop-up sub-palette.) Decreases the font size of the selected comment text.

Increase Size +

(On the Comments pop-up sub-palette.) Increases the font size of the selected comment text.

Parentheses (a)

Surrounds the selection in parentheses or other bracket styles, depending on the nesting.

Select Proposition

Selects the entire proposition.

Select Out x^2_1

Selects the expression just outside the current selection.

Select First x^2_1

Selects the first element of the current selection.

Select Next x^2_1

Selects the next element of the current structure.

Select In x^2_1

Selects the individual elements within the current selection.

Undo

Reverts the current notebook to its state before the most recent change or command. Some changes can not be undone.

Cut ✂

Copies the current selection to the clipboard, then deletes the selection. Expressions are stored as text in the Fortranish format.

Copy ⧉

Copies the current selection to the clipboard. Expressions are stored as text in the Fortranish format.

Paste

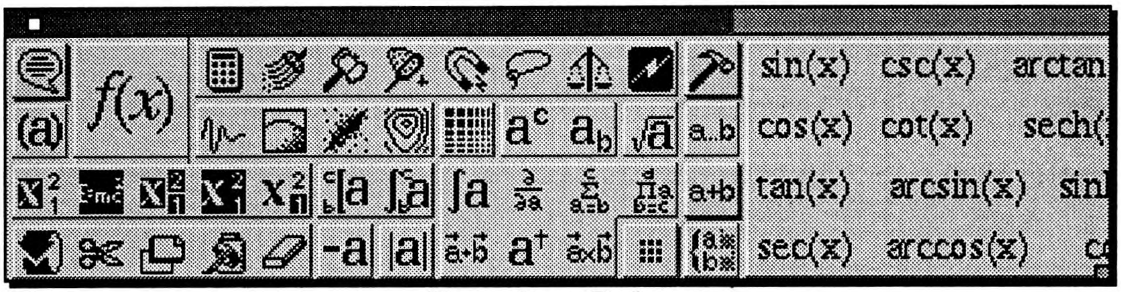
Pastes the contents of the clipboard, replacing, or just after, the current selection. If there is no selection, the clipboard is pasted at the bottom of the current notebook.

Clear

Deletes the current selection; same as pressing delete on the keyboard.

Functions Palette

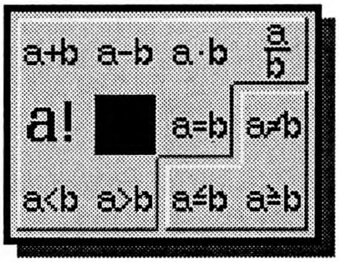
The Functions palette contains buttons which insert operators, make selections, perform clipboard commands, create tables, create graphs, and perform algebraic manipulations. Of course, it also has buttons which insert functions. The right area of the palette lists defined function names, which you can click on to insert in the notebook. Clicking the big *f*(*x*) button switches to the variables palette.



Functions Palette

Operators pop-up a+b

Pops up a sub-palette containing various operator buttons. Release the mouse button with the pointer on the desired operator icon to insert it in the notebook.



operators pop-up palette

Addition a+b

(On the operators pop-up sub-palette.) Creates the addition op.

x	$x + ?$
selection	*result*

Subtraction a-b

(On the operators pop-up sub-palette.) Creates the subtraction op.

x	$x - ?$
selection	*result*

Multiplication a·b

(On the operators pop-up sub-palette.) Creates the multiply op.

x	$x\ ?$
selection	*result*

Fraction $\frac{a}{b}$

(On the operators pop-up sub-palette.) Creates the division op.

x	$\frac{x}{?}$
selection	*result*

Factorial
a!

(On the operators pop-up sub-palette.) Appends the factorial exclamation point to the selection.

x	$x!$
selection	*result*

Equals
a=b

(On the operators pop-up sub-palette.) Creates the equals op.

x	$x = ?$
selection	*result*

Not Equals
a≠b

(On the operators pop-up sub-palette.) Creates the not equals op.

x	$x \neq ?$
selection	*result*

Less than
a<b

(On the operators pop-up sub-palette.) Creates the less than op.

x	$x < ?$
selection	*result*

Greater than
a>b

(On the operators pop-up sub-palette.) Creates the greater than op.

x	$x > ?$
selection	*result*

Less than or Equal to
a≤b

(On the operators pop-up sub-palette.) Creates the less than or equal to op.

x	$x \leq ?$
selection	*result*

Greater than or Equal to
a≥b

(On the operators pop-up sub-palette.) Creates the greater than or equal to op.

x	$x \geq ?$
selection	*result*

Parentheses
(a)

Surrounds the selection in parentheses or other bracket styles, depending on the nesting.

x	(x)
selection	*result*

Superscript
a^{c}

Attaches a superscript to the selection, for an exponent.

x	$x^{?}$
selection	*result*

Subscript
a_{b}

Attaches a subscript to the selection, for an index.

x	$x_{?}$
selection	*result*

Square Root
√a

Surrounds the selection with a radical.

x	$\sqrt{x}$
selection	*result*

Range a..b	Creates the range op.	x *selection*	$x \ldots ?$ *result*
Evaluate At $\Big[_{b}^{c} a$	Creates the Evaluate At structure.	x *selection*	$\Big[_{?=-1}^{?=1} x$ *result*
Integration $\int_{b}^{c} a$	Creates the definite integral op (with limits).	x *selection*	$\int_{?}^{?} x \, d \, ?$ *result*
Integration $\int a$	Creates the indefinite integral op (without limits).	x *selection*	$\int x \, d \, ?$ *result*
Partial Derivative $\frac{\partial}{\partial a}$	Creates the partial derivative op.	x *selection*	$\frac{\partial}{\partial x}$ *result*
Summation $\sum_{a=b}^{c}$	Creates the summation op.	x *selection*	$\sum_{x=?}^{?}$ *result*
Pi Product $\prod_{b=c}^{d} a$	Creates the pi product op.	x *selection*	$\prod_{?=?}^{?} x$ *result*
Negation -a	Negates the selection.	x *selection*	$x - ?$ *result*
Absolute $\|a\|$	Surrounds the selection with absolute value bars.	x *selection*	$\|x\|$ *result*
Dot Product $\vec{a} \cdot \vec{b}$	Creates the dot product op.	x *selection*	$x \bullet ?$ *result*
Adjoint $a^{\dagger}$	Appends the adjoint character † to the selection.	x *selection*	$x^{\dagger}$ *result*
Cross Product $\vec{a} \times \vec{b}$	Creates the cross product op.	x *selection*	$x \times ?$ *result*

Conditional

Creates the conditional op.

x

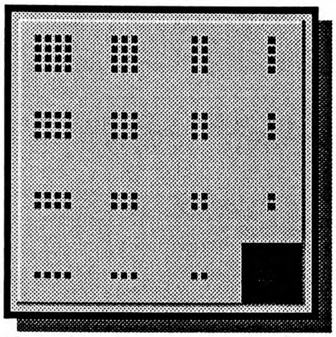
selection

$$\begin{cases} x & (? < 0) \\ ? & (? \geq 0) \end{cases}$$

result

Matrices pop-up

Pops up a sub-palette containing buttons for creating matrices of various sizes. Release the mouse button with the pointer on the desired matrix icon to insert it in the notebook.



Matrices pop-up palette

Calculate

Performs the Calculate manipulation.

4π

selection

12.5663706143592

after manipulation

Simplify

Performs the Simplify manipulation.

$5 + x^2 + x^3 + x \cdot 2 + \frac{x^{10}}{2}$

selection

$\frac{1}{2}x^{10} + x^3 + x^2 + 2x + 5$

after manipulation

Expand

Performs the Expand manipulation.

$\left(\left[x + 1\right]^2 + 1\right)^2$

selection

$x^4 + 4x^3 + 8x^2 + 8x + 4$

after manipulation

Collect

Performs the Collect manipulation.

$\frac{1}{x} + \frac{1}{y} + \frac{1}{z}$

selection

$\frac{xy + xz + yz}{xyz}$

after manipulation

Factor

Performs the Factor manipulation.

$x^2 - 11x - 26$

selection

$(x + 2)(x - 13)$

after manipulation

Isolate

Performs the Isolate manipulation.

$y = 5\,x + 7$

selection

$x = \frac{1}{5}(y - 7)$

after manipulation

Apply

Performs the Apply manipulation.

$0 = x^2 - 4x + 3$	$0 = x^2 - 4x + 3$
selection	*after manipulation*

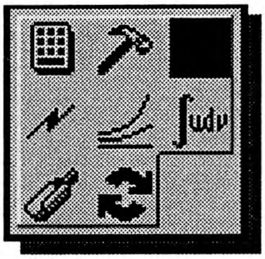
Transform

Performs the Transform manipulation. The results of this manipulation depend upon the declared transformation rules, if any. For example:

$(\cos[x])^2$	$1 - (\sin[x])^2$
selection	*after manipulation*

Manipulations pop-up

Pops up a sub-palette containing various manipulation buttons. Release the mouse button with the pointer on the desired manipulation icon to perform the manipulation.



manipulations pop-up palette

UnCalculate

(On the manipulations pop-up sub-palette.) Performs the UnCalculate manipulation.

12.5663706143592	4π
selection	*after manipulation*

MiniExpand

(On the manipulations pop-up sub-palette.) Performs the MiniExpand manipulation.

$([x+1]^2 + 1)^2$	$(x+1)^4 + 2(x+1)^2 + 1$
selection	*after manipulation*

Substitute

(On the manipulations pop-up sub-palette.) Performs the Substitute manipulation.

$x = \frac{1}{2}$	$z = y + x$	$z = y + \frac{1}{2}$
selected equation	*selected target*	*after manipulation*

Taylor Series

(On the manipulations pop-up sub-palette.) Performs the Taylor Series manipulation, invoking the Taylor Series dialog.

Integrate by Parts

(On the manipulations pop-up sub-palette.) Performs the Integrate by Parts manipulation.

$\int x \sin(x)\, dx$	$\int x \sin(x)\, dx = -x \cos(x) + \sin(x)$
selection	*after manipulation*

Move Over

(On the manipulations pop-up sub-palette.) Performs the Move Over manipulation.

$5x + 3 = y$	$5 = \frac{y-3}{x}$
selection	*after manipulation*

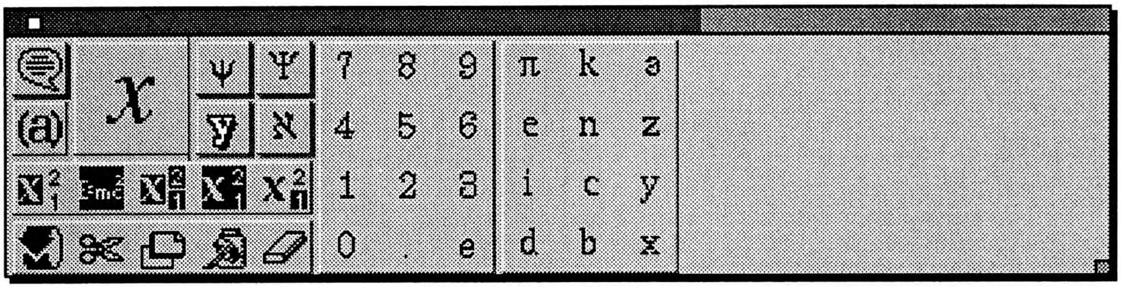
Commute

(On the manipulations pop-up sub-palette.) Performs the Commute manipulation.

$3x^2 + x^2 - 2x + 5$	$3x^2 + (- 2x + x^2) + 5$
selection	*after manipulation*

VARIABLES PALETTE

The Variables palette contains buttons which insert Greek letters and special symbols, make selections, perform clipboard commands, and insert wildcard names. The right area of the palette lists defined variable and constant names, which you can click on to insert in the notebook. Clicking the big *x* button switches to the functions palette.



Variables Palette

Numbers
1 2 3
0 . e

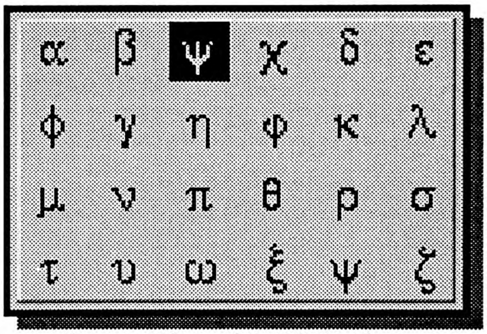
Click on a number to insert it in the notebook. There is also a decimal point and e for scientific notation.

Names

The right area of the variables palette is reserved for declared constants and variables. As you declare names, Theorist places them here. Click on a name to insert it in the notebook.

Lower Greek pop-up
ψ

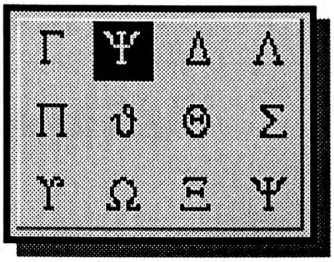
Pops up a sub-palette containing the lower case Greek letters. Release the mouse button with the pointer on the desired character to insert it in the notebook.



lower Greek sub-palette

Upper Greek pop-up
Ψ

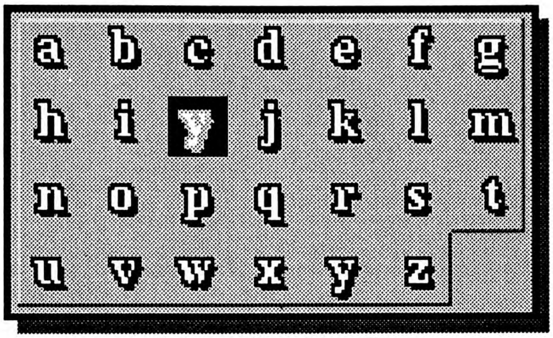
Pops up a sub-palette containing the upper case Greek letters. Release the mouse button with the pointer on the desired character to insert it in the notebook.



upper Greek sub-palette

Wildcard pop-up

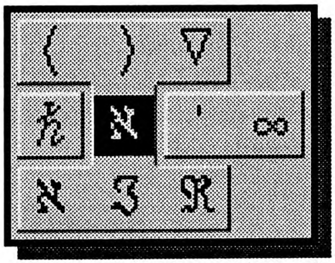
Pops up a sub-palette containing buttons for inserting wildcards. Release the mouse button with the pointer on the desired character to insert it in the notebook.



wildcard sub-palette

Special Symbols pop-up

Pops up a sub-palette containing miscellaneous symbols. Release the mouse button with the pointer on the desired character to insert it in the notebook.



special symbols sub-palette

Appendices

Appendix A: Keystrokes

Greeks and Symbols

The following tables list the keystrokes to use for upper- and lower-case Greek letters and special characters.

Universal Keystrokes

Type apostrophe then keystroke listed on Macintosh or Windows

modifier ↓ key→	A	B	C	D	E	F	G	H	I	J	K	L	M
	α	β	χ	δ	ε	ϕ	γ	η		φ	κ	λ	μ
shift	ℵ	⟨		Δ		Φ	Γ	ℏ	ℑ	ϑ	⟩	Λ	

modifier ↓ key→	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
	η		π	θ	ρ	σ	τ	υ		ω	ξ	ψ	ζ
shift	∞		Π	Θ	ℜ	Σ		ϒ		Ω	Ξ	Ψ	∇

Macintosh Shortcut Keystrokes

Type modifier(s) and key together

modifier(s) ↓ key→	A	B	C	D	E	F	G	H	I	J	K	L	M
option	α	β	χ	δ	ε	ϕ	γ	η		φ	κ	λ	μ
option shift	ℵ	⟨		Δ		Φ	Γ	ℏ	ℑ	ϑ	⟩	Λ	

modifier(s) ↓ key→	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
option	η		π	θ	ρ	σ	τ	υ		ω	ξ	ψ	ζ
option shift	∞		Π	Θ	ℜ	Σ		ϒ		Ω	Ξ	Ψ	∇

Windows Shortcut Keystrokes

Type modifier(s) and key together

modifier(s) ↓ key→	A	B	C	D	E	F	G	H	I	J	K	L	M
alt Ctrl	α	β	χ	δ	ε	ϕ	γ	η		φ	κ	λ	μ
alt Ctrl shift	ℵ	⟨		Δ		Φ	Γ	ℏ	ℑ	ϑ	⟩	Λ	

modifier(s) ↓ key→	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
alt Ctrl	η		π	θ	ρ	σ	τ	υ		ω	ξ	ψ	ζ
alt Ctrl shift	∞		Π	Θ	ℜ	Σ		ϒ		Ω	Ξ	Ψ	∇

Menu Items and Palette Buttons

The following tables list the keystrokes to use for menu items and palette buttons, and other editing functions

You can perform a manipulation in place by holding the ⌘ and option keys (Macintosh) or Ctrl and alt keys (Windows) while pressing the manipulation keystroke (number keys on Macintosh or functions keys on Windows).

Macintosh Keystrokes

Menu/Submenu	Menu Item	Palette Button	Keystroke
File	New Notebook	*n/a*	⌘ N
File	Open Notebook	*n/a*	⌘ O
File	Close Notebook	*n/a*	⌘ W
File	Save Notebook	*n/a*	⌘ S
File	Print Notebook	*n/a*	⌘ P
File	Quit	*n/a*	⌘ Q
Edit	Undo	✓	⌘ Z
Edit	Cut	✂	⌘ X
Edit	Copy	⧉	⌘ C
Edit	Copy as Picture	*n/a*	⌘ F
Edit	Paste	📋	⌘ V
Edit	Clear	⌫	delete
Edit	Select In	X^2_1	⌘ E
Edit	Select Out	X^2_1	shift esc
n/a	*n/a*	X^2_1	tab
n/a	*n/a*	=mc	⌘ A
Edit	Comment Font	*n/a*	⌘ option C
Edit▸Comment Style	Plain	P	⌘ T
Edit▸Comment Style	Bold	**B**	⌘ B
Edit▸Comment Style	Italic	*I*	⌘ I
Edit▸Comment Style	Underline	*n/a*	⌘ U
Notebook	Clarify	*n/a*	⌘ option shift space
Notebook▸Windows	Palette	▣	⌘ option B
Notebook	Expose	*n/a*	⌘ `
Notebook	Collapse	*n/a*	⌘ \

Notebook	Indent Left	n/a	⌘ [
Notebook	Indent Right	n/a	⌘]
Notebook▸Insert	Page Break	n/a	⌘ option P
Notebook▸Insert	Case Theory	n/a	⌘ option T
Notebook▸Insert	Transform Rule	n/a	⌘ Y
Notebook	Make Working Stmt	n/a	⌘ D
Notebook	Stop Working Stmt	n/a	⌘ option D
Notebook	Get Info	n/a	⌘ option J
Notebook▸Display Precision	2 digits	n/a	⌘ option shift 2
Notebook▸Display Precision	3 digits	n/a	⌘ option shift 3
Notebook▸Display Precision	4 digits	n/a	⌘ option shift 4
Notebook▸Display Precision	5 digits	n/a	⌘ option shift 5
Notebook▸Display Precision	6 digits	n/a	⌘ option shift 6
Notebook▸Display Precision	7 digits	n/a	⌘ option shift 7
Notebook▸Display Precision	8 digits	n/a	⌘ option shift 8
Notebook▸Display Precision	10 digits	n/a	⌘ option shift 0
Notebook▸Display Precision	15 digits	n/a	⌘ option shift 9
Notebook	Notebook Font	n/a	⌘ option F
Manipulate	Calculate	🖩	⌘ 5
Manipulate	Simplify		⌘ 1
Manipulate	Expand		⌘ 6
Manipulate	Collect		⌘ 7
Manipulate	Factor	🧲	⌘ 8
Manipulate	Apply	⚖	⌘ 0
Manipulate	Transform	⚡	⌘ 9
Manipulate▸Other	Int. by Parts	∫udv	⌘ 4
Manipulate▸Table	Generate	▦	⌘ option G
Manipulate▸Table	Int. Differential Eq.	n/a	⌘ option O
Manipulate	ReManipulate Now	n/a	⌘ =
Manipulate▸Preferences	Auto Simplify	n/a	⌘ 2
Manipulate▸Preferences	Auto Casing	n/a	⌘ 3
Graph▸y = f(x)	Linear	∿	⌘ G

Graph▸y = f(x)	Complex 3D	*n/a*	⌘ option K
Graph▸z = f(x, y)	Contour 2D	◎	⌘ M
Graph▸z = f(x, y)	Color 3D	▨	⌘ K
Graph▸z = f(x, y)	Illum. 3D	*n/a*	⌘ option U
Graph▸z = f(x, y)	Spherical 3D	*n/a*	⌘ option S
Graph▸z = f(x, y)	Cylindrical 3D	*n/a*	⌘ option Y
Graph▸Other	Parametric	*n/a*	⌘ option R
Graph▸Other	Space Curve	*n/a*	⌘ option Z
Graph▸Other	Implicit	*n/a*	⌘ option H
Graph▸Scatter	Linear	⋰	⌘ option V
Graph▸Additional	Add Line Plot	*n/a*	⌘ option L
Graph▸Additional	Add Scatter Plot	*n/a*	⌘ option W
Graph▸Animation	Start	*n/a*	⌘ option M

Windows Keystrokes

Menu/Submenu	Menu Item	Palette Button	Keystroke
File	New Notebook	*n/a*	Ctrl N
File	Open Notebook	*n/a*	Ctrl O
File	Close Notebook	*n/a*	Ctrl W
File	Save Notebook	*n/a*	Ctrl S
File	Print Notebook	*n/a*	Ctrl P
File	Exit	*n/a*	alt F4
Edit	Undo	✓	Ctrl Z
Edit	Cut	✂	Ctrl X
Edit	Copy	⧉	Ctrl C
Edit	Copy as Picture	*n/a*	Ctrl F
Edit	Paste	📋	Ctrl V
Edit	Clear	⌫	backspace *or* delete
Edit	Select In	X^2_1	Ctrl E
Edit	Select Out	X^2_1	shift esc
n/a	*n/a*	X^2_1	tab

n/a	*n/a*	$E=mc^2$	Ctrl A
Edit	Comment Font	*n/a*	alt Ctrl C
Edit▸Comment Style	Plain	P	Ctrl T
Edit▸Comment Style	Bold	**B**	Ctrl B
Edit▸Comment Style	Italic	*I*	Ctrl I
Edit▸Comment Style	Underline	*n/a*	Ctrl U
Notebook	Clarify	*n/a*	Ctrl shift space
Notebook▸Windows	Palette	▣	Ctrl shift B
Notebook	Expose	*n/a*	Ctrl `
Notebook	Collapse	*n/a*	Ctrl \
Notebook	Indent Left	*n/a*	Ctrl [
Notebook	Indent Right	*n/a*	Ctrl]
Notebook▸Insert	Page Break	*n/a*	Ctrl shift P
Notebook▸Insert	Case Theory	*n/a*	Ctrl shift T
Notebook▸Insert	Transform Rule	*n/a*	Ctrl Y
Notebook	Make Working Stmt	*n/a*	Ctrl D
Notebook	Stop Working Stmt	*n/a*	Ctrl shift D
Notebook	Get Info	*n/a*	Ctrl shift J
Notebook▸Display Precision	2 digits	*n/a*	Ctrl F2
Notebook▸Display Precision	3 digits	*n/a*	Ctrl F3
Notebook▸Display Precision	4 digits	*n/a*	Ctrl F4
Notebook▸Display Precision	5 digits	*n/a*	Ctrl F5
Notebook▸Display Precision	6 digits	*n/a*	Ctrl F6
Notebook▸Display Precision	7 digits	*n/a*	Ctrl F7
Notebook▸Display Precision	8 digits	*n/a*	Ctrl F8
Notebook▸Display Precision	10 digits	*n/a*	Ctrl F10
Notebook▸Display Precision	15 digits	*n/a*	Ctrl F9
Notebook	Notebook Font	*n/a*	Ctrl shift F
Manipulate	Calculate	🖩	F5
Manipulate	Simplify		F11
Manipulate	Expand		F6
Manipulate	Collect		F7

Manipulate	Factor	🧲	F8
Manipulate	Apply	⚖	F10
Manipulate	Transform	⚡	F9
Manipulate▸Other	Int. by Parts	∫udv	Ctrl 4
Manipulate▸Table	Generate	▦	Ctrl shift G
Manipulate▸Table	Int. Differential Eq.	*n/a*	Ctrl shift O
Manipulate	ReManipulate Now	*n/a*	Ctrl =
Manipulate▸Preferences	Auto Simplify	*n/a*	Ctrl 2
Manipulate▸Preferences	Auto Casing	*n/a*	Ctrl 3
Graph▸y = f(x)	Linear	〰	Ctrl G
Graph▸y = f(x)	Complex 3D	*n/a*	Ctrl shift K
Graph▸z = f(x, y)	Contour 2D	◎	Ctrl M
Graph▸z = f(x, y)	Color 3D	▨	Ctrl K
Graph▸z = f(x, y)	Illum. 3D	*n/a*	Ctrl shift U
Graph▸z = f(x, y)	Spherical 3D	*n/a*	Ctrl shift S
Graph▸z = f(x, y)	Cylindrical 3D	*n/a*	Ctrl shift Y
Graph▸Other	Parametric	*n/a*	Ctrl shift R
Graph▸Other	Space Curve	*n/a*	Ctrl shift Z
Graph▸Other	Implicit	*n/a*	Ctrl shift H
Graph▸Scatter	Linear	⁘	Ctrl shift V
Graph▸Additional	Add Line Plot	*n/a*	Ctrl shift L
Graph▸Additional	Add Scatter Plot	*n/a*	Ctrl shift W
Graph▸Animation	Start	*n/a*	Ctrl shift M

APPENDIX B: USING EXPRESSIONIST WITH THEORIST

Expressionist is a typeset quality equation editor developed by Prescience Corporation and published by Waterloo Maple Software.

When you use **Copy as Picture** on the **Edit** menu to copy an equation or expression, Theorist generates a Macintosh QuickDraw image or Windows Metafile image of the current selection. This image can be moved to almost any program, and is satisfactory for screen display or for informal printed presentations.

The picture created by Copy as Picture also includes special hidden Expressionist comments which allow you to paste the figure into Expressionist for high quality equation typesetting. You can edit the equation in Expressionist, just as though you had typed it in directly. You can then copy and paste the image into a word processor or desktop publishing program for a typeset mathematical expression.

You can also copy equations from Expressionist into Theorist. To copy an Expressionist equation into Theorist, first choose the Theorist.CAT translation table and set copying for Copy As Text in Expressionist. Select an expression in Expressionist, then choose **Copy** from the **Edit** menu, click on the Theorist window and paste it in.

The translation may not be exact because Expressionist equations use very few syntax rules. Some editing may be necessary. In particular, names that should be multiplied may be concatenated, creating unintended new names. For example, if the two names x and b are next to each other in an expression in Expressionist (without a separating space), when brought into Theorist these two names become the single name “bx”. To correct this, click between the two characters and press space or *.

Appendix C: Distribution Disks

Theorist for Macintosh

The following files and folders are included on the floppy disks included in the Theorist for Macintosh package.

Standard Application Disk

Theorist®
This application is the standard edition of Theorist, which runs on all Macintosh models from the Macintosh Plus onward. However, one of the other application editions may be optimal for different Macintosh models.

New Notebook
A notebook containing many declared names and transformation rules, opened automatically when Theorist is started and when new notebooks are created. This must be in the same folder as the Theorist application.

TheoText.CAT
A text translation file. This must be in the same folder as the Theorist application.

FPU Application Disk

Theorist®/FPU
This application is the FPU edition of Theorist, made to take advantage of the 68020 and 68030 CPUs and the 68881 and 68882 math co-processors (and the 68040 CPU with built-in co-processor emulation). Use this version if you work on a Mac II, Mac IIx, Mac IIcx, Mac IIci, Mac IIfx, Mac SE/30, or any Macintosh equipped with a 68020, 68030, or compatible CPU, *and* a 68881, 68882 or compatible floating point co-processor chip (or just a 68040). This includes most Quadra and Centris Macintosh models. Do not use this edition on PowerMacintosh models.

New Notebook
A notebook containing many declared names and transformation rules, opened automatically when Theorist is started and when new notebooks are created. This must be in the same folder as the Theorist application.

TheoText.CAT
A text translation file. This must be in the same folder as the Theorist application.

PowerMac Application Disk

Theorist®/PowerMac
This application is the PowerMac edition of Theorist, made to take advantage of the PowerPC processor in PowerMacintosh models. Do not use this edition on any other Macintosh models.

New Notebook
A notebook containing many declared names and transformation rules, opened automatically when Theorist is started and when new notebooks are created. This must be in the same folder as the Theorist application.

TheoText.CAT
A text translation file. This must be in the same folder as the Theorist application.

Notebooks Disk

Graphics — A folder containing several notebooks with sample graphs demonstrating some of the graphs you can create.

Conformal Mapping
3D Lissajous
Electron Orbital
Elliptic Coordinates
Gibb's Phenomenon
Spherical Contours
Spherical Family
The Planets
Two Sqrt Leaves

Color Graphics — A folder containing several notebooks with sample color graphs.

Custom Color Sets
Direct Color Models
Five Roots of One
Spherical Plate
Surface Color Legends
Surface Styles
Yellow Seashell
Twisted Wire

Mathematics — A folder containing several notebooks of mathematical rules including folders with more notebooks.

3D Calculus — Rules for three-dimensional linear algebra and vector calculus
Finite & Infinite Series — Rules for summations
Laplace Transforms — Rules for Laplace and Inverse Laplace transforms
Number Theory — Rules for working with integers
Re( ) & Im( ) — Rules for real and imaginary functions
Statistics — Rules for calculating means, standard deviations and linear regression over matrix numbers
Units — Rules defining a wide variety of units

Integral Tables — Folder of notebooks containing integration rules

Exponential Integrals

Exponential Trig Integrals
Hyperbolic Integrals
Integration
Inverse Hyperbolic Integrals
Inverse Trig Integrals
Irrational Integrals
Logarithmic Integrals
Trigonometric Integrals

Special Functions — Folder of notebooks containing numerical definitions of special functions

Gamma and related functions
Orthogonal Polys P, T, H, L
Spherical Bessel j, y, k, i
Spherical Harmonics
Zeta function

Transcendental Functions — Folder of notebooks containing rules for transcendental functions

Hyperbolic Functions
Log & Power Functions
Trig Functions

Samples — Folder of sample interactive notebooks for exploring Theorist

Theorist for Windows

The Theorist for Windows package includes a floppy diskette labeled Install 1. When the Install program is executed from this disk and its instructions are followed, the following files and directories are installed on your hard disk. (Note: other files related to the operating system may also be installed but are not directly related to Theorist.)

NEWNOTE.THE — A notebook containing many declared names and transformation rules, opened automatically when Theorist is started and when new notebooks are created. This must be in the same directory as the Theorist executable file.

THEOTEXT.CAT — A text translation file. This must be in the same directory as the Theorist application.

Graphics — A directory containing several notebooks with sample graphs demonstrating some of the graphs you can create.

CONFMAP.THE	Conformal Mapping
3DLISS.THE	3D Lissajous
ELECORB.THE	Electron Orbital
ELLIPCOR.THE	Elliptic Coordinates

GIBBPHE.THE	Gibb's Phenomenon
SPHCONT.THE	Spherical Contours
SPHFAM.THE	Spherical Family
PLANETS.THE	The Planets
TWOSQRT.THE	Two Sqrt Leaves
CGraphs	A directory containing several notebooks with sample color graphs demonstrating some of the graphs you can create.
CUSTOMC.THE	Custom Color Sets
DIRECTC.THE	Direct Color Models
FIVEROOT.THE	Five Roots of One
SPHPLATE.THE	Spherical Plate
SURFCLEG.THE	Surface Color Legends
SURFSTY.THE	Surface Styles
YELLSEA.THE	Yellow Seashell
TWISTW.THE	Twisted Wire
Math	A directory containing several notebooks of mathematical rules including subdirectories with more notebooks.
3DCAL.THE	3D Calculus, rules for three-dimensional linear algebra and vector calculus
FININF.THE	Finite & Infinite Series, rules for summations
LAPTRANS.THE	Laplace Transforms, rules for Laplace and Inverse Laplace transforms
NUMBTHEO.THE	Number Theory, rules for working with integers
RE&IM.THE	Rules for real and imaginary functions
STATIST.THE	Statistics, rules for calculating means, standard deviations and linear regression over matrix numbers
UNITS.THE	Rules defining a wide variety of units
IntegTab	Integral Tables, directory of notebooks containing integration rules
EXPINT.THE	Exponential Integrals
EXPTRIG.THE	Exponential Trig Integrals
HYPINT.THE	Hyperbolic Integrals
INTEG.THE	Integration
INVHYP.THE	Inverse Hyperbolic Integrals
INVTRIG.THE	Inverse Trig Integrals
IRRALG.THE	Irrational Integrals
LOGINT.THE	Logarithmic Integrals
TRIGINT.THE	Trigonometric Integrals
SpecFn	Directory of notebooks containing numerical definitions of special functions
GAMMA.THE	Gamma and related functions
ORTHO.THE	Orthogonal Polys P, T, H, L

SPBESS.THE	Spherical Bessel j, y, k, i
SPHARM.THE	Spherical Harmonics
ZETAFN.THE	Zeta function
TransFn	Directory of notebooks containing rules for transcendental functions
HYPFN.THE	Hyperbolic Functions
LOGPOW.THE	Log & Power Functions
TRIGFN.THE	Trig Functions
Samples	Directory of sample interactive notebooks for exploring Theorist

GLOSSARY

This glossary defines some general mathematical terms, and terms specific to Theorist.

Adjoint The transpose of the conjugate of a matrix, or the conjugate of a scalar.

Algorithm A mechanical procedure for solving a problem in a finite number of steps.

Animation A sequence of images displayed in quick succession to simulate movement.

Apply A Theorist manipulation whereby two parts of an equation or expression can be augmented simultaneously so that the two parts cancel out.

Arbitrary constant A constant included as part of a result when multiple solutions are possible, each of which is described by substituting any value for the arbitrary constant. Also called the constant of integration.

Arbitrary integer An integer constant included as part of a result when multiple solutions are possible, each of which is described by substituting any integer value for the arbitrary constant.

Argument An expression passed to a function or op, sometimes enclosed in parentheses.

Associative A property of a binary operator. When three or more elements are combined with a binary operator that is associative for those elements, the result does not depend on how the elements are grouped into pairs. If multiplication is associative across the names A, B and C then $A(BC) = (AB)C$.

Auto Casing A Theorist option. If Auto Casing is on, manipulations that generate more than one possible solution create case theories or arbitrary constants.

Auto Collapsar A Theorist option. If Auto Collapsar is on, large expressions generated by manipulations are displayed in abbreviated form.

Auto Simplify A Theorist option. If Auto Simplify is on, most manipulations conclude by executing the Simplify command.

Axis A Theorist plot proposition that displays a labeled axis (with tick marks) in a graph viewport.

Base A name, number, or expression that is raised to a power (with a superscript) or qualified with an index (or subscript).

Behavior Properties of names. Theorist uses fifty seven predefined behaviors; you can also define and declare the behavior of a names with an equation.

Boolean A value that is True or False (or 1 or 0).

Case theory A Theorist proposition. Case theories enclose side derivations. Whatever is true in enclosing theories is true in the contained case theory; whatever is true inside a case theory is ignored outside of that theory.

Ceiling A Theorist predefined name that acts as a function. It rounds a given argument to the next highest integer. See floor.

Class The Theorist categorization system of names and expressions that determine how they behave algebraically. There are five classes: Constant, Variable, M-Linear Operator, D-Linear Operator, Function.

CMY A color encoding system that describes colors by their Cyan, Magenta, and Yellow components.

Collapsar A Theorist expression that has been abbreviated so that some terms or factors are replaced by an ellipse (...).

Collapse A Theorist command to abbreviate long expressions (into collapsars) or conceal the daughter propositions of a proposition.

Collapse All A Theorist command to collapse all propositions in the current notebook.

Collect A Theorist manipulation that collects common factors out of a sum or product and other similar tasks.

Commutative A property of a binary operator. When two or more elements are combined with a binary operator that is commutative for those elements, the result does not depend on the order of the elements. If multiplication is commutative over the names A and B then $AB = BA$.

Commute A Theorist manipulation used to interchange two expressions in a product or sum, or reverse the order of all (or some) of the elements in a sum or product.

Complex arithmetic Arithmetic of complex numbers.

Complex numbers Numbers that have real and imaginary components. The imaginary component is designated with the constant i, the square root of negative one (e.g., a + bi).

Complex plane

A plane with pair of orthogonal axes such that one axis corresponds to the real part of a complex number (a) and the other to the imaginary part (b). Also called an Argand diagram.

Conjugate

The conjugate of a complex number a+bi is the complex number a–bi. These two form a conjugate pair and each is the conjugate of the other. The complex conjugate of a matrix is formed by replacing each element of the matrix with its conjugate.

Constant

A Theorist class. Names of class constant have fixed numerical values that do not change, unless you change them. Graph bounds (top, bottom, etc.) are constants; using the rocket or knife changes them.

Contour plot

A Theorist plot proposition. Contour plots represent a surface with a series of labeled elevation lines. Contour plots are drawn in panels like surface plots.

Copy as Picture

A Theorist command that copies the current selection to the clipboard as a picture (using the Macintosh PICT format or Windows MetaFile format).

Cross product

A Theorist op. A cross product is the product of two vectors with either two or three elements. For two-vectors, the result is a scalar; for three-vectors, the result is another vector. If A and B are three vectors and C is the cross product of A and B, the length of C is the product of the lengths of A and B, and the sine of the angle between the two vectors.

D-Linear Operator

A Theorist class. Names of class D-Linear Operator are derivatives or other ops that are neither commutative nor associative under multiplication.

Dependent variable

A variable in an equation whose value is contingent on the independent variables in that equation, and the equation itself.

Derivative

The rate of change of a dependent variable (or function) with respect to an independent variable. Theorist works with derivatives symbolically, and calculates them numerically.

Details

Part of a Theorist graph. The details box contains a description of each of the graphic elements (plots) shown in the graph.

Determinant

A sum of products of matrix elements in a particular pattern (some terms are negative). The determinant of $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ is $ad - bc$.

Differential

A product of the differential operator d with any variable. Its value is infinitesimal; you get derivatives by dividing different differentials.

Differentiation

The process of finding the derivative of a function. Differentiation is the inverse process of integration.

Display precision
A Theorist option that determines how many digits are displayed for numeric values.

Dot product
A Theorist op. A dot product is the scalar product of two vectors (with either two or three elements). If C is the dot product of A and B, the length of C is the product of the lengths of A and B, and the cosine of the angle between the two vectors.

Entry palette
Part of Theorist's main display. Use the palette to enter names and functions. The palette consists of the variable palette (for entering variables and constants) and the function palette (for entering ops and functions).

EPS
Acronym for Encapsulated PostScript. EPS files use the PostScript language to describe computer generated images, and are used to move images between programs and systems.

Equation
A Theorist expression consisting of two subexpressions connected by the equals op.

Expand
A Theorist manipulation used to break up an expression into a sum of smaller pieces, and other similar tasks.

Exponent
A number or expression placed as a superscript to a base indicating repeated multiplication.

Expression
A Theorist entity. An expression is a name, a number, or an op that encloses one or more name or number.

Factor
A Theorist manipulation used to consolidate polynomials into factored form, and other similar tasks.

Factorial
A Theorist op. The factorial of an integer is the number obtained by multiplying all the positive integers less than or equal to the given integer. The factorial of other real and complex numbers is defined by using the Gamma function.

Floor
A Theorist predefined name that behaves as a function. It rounds a given argument to the next lowest integer. See ceiling.

Fortranish
A Theorist option. If Fortranish is on, expressions are entered from the keyboard in accordance with the precedence rules for operators established by FORTRAN and other programming languages.

Function
A Theorist class. Names of class function use the function op to enclose one or more input arguments, and they return a value determined by the definition of the particular function.

Functions palette
One side of the Theorist entry palette, used to enter ops and functions. This palette displays a large $f(x)$ on the left side.

Gamma function A function used in advanced mathematics. For any real or complex number:

$$\Gamma(z)\int\int_0^{\infty} t^{z-1}e^{-t}dt$$

For integers, $\Gamma(z) = (n-1)!$

Graph A frame work for displaying plots and those plots. In Theorist, a graph consists of a graph theory (a two- or three-dimensional shell) and a set of plots.

Graph Theory A Theorist proposition. Graph theories display Line, Contour, Scatter, and Surface plots, and possibly Axes and Grid lines, in a viewport.

Greek subpalettes The ψ pop-up on the variables palette for entering lower case Greek letters and the Ψ pop-up on the variables palette for entering upper case Greek letters.

Grid lines A Theorist plot proposition that displays a set of Grid lines in a graph theory.

HLS A color encoding system that describes colors by their Hue, Lightness, and Saturation components

HSV A color encoding system that describes colors by their Hue, Saturation, and Value components.

Hyperbolic functions Functions akin to trigonometric functions, related to hyperbolas as trigonometric functions are related to circles.

Independence declaration A Theorist proposition that declares one or more variables independent of each other or independent of all variables. By default, each variable is dependent on all other variables.

Independent variables A variable in an equation whose value is allowed to assume values freely over a specified domain. The values of the independent variable(s) (and one or more equation) determine the values of the dependent variable(s).

Insertion point A blinking vertical (or occasionally horizontal) bar that indicates where the next entry is placed. An insertion point does not contain a selection. You create an insertion point with the mouse, and can move it with the arrow keys and the tab key.

Integral A Theorist op. The integral op encloses an expression and represents a summation of infinitesimally small values.

Integration The process of finding an integral. Integration is the inverse process of differentiation (i.e., finding a function whose derivative is the original function).

Isolate A Theorist manipulation used to rearranges an equation so that the selected expression is separated out and placed on one side of the equals sign and the rest of the equation is on the other.

Knife A Theorist editing tool found on the right-hand side of two- and three-dimensional graph theories. Use the Knife to slice out a portion of a plot for closer inspection.

Line A Theorist plot proposition that displays a curve (based on a selected equation) in a graph theory.

Linear An equation or an expression that is of the first degree. The graph of a linear relationship produces a straight line (or a higher dimensional equivalent, e.g., a plane). The relationship between two expressions is linear if the derivative of one with respect to the other is constant.

Log A Theorist predefined name that acts as a function. It returns the common logarithm (base 10) of a given argument.

Logarithm The power to which a base number must be raised to equal a given number. For a base of 10, log(1000) = 3; the number ten must be raised to the power of three to equal 1000.

M-Linear Operator A Theorist class. Names of class M-Linear Operator are matrices or other ops that are associative but not commutative under multiplication.

Manipulations Theorist commands invoked from a pull-down menu, from the keyboard, or with the mouse used to generate new, equally valid, expressions or equations from existing expressions or equations.

Matrix A Theorist op. A matrix is a set of elements arranged in a rectangular array of rows and columns. Row vectors are horizontal matrices consisting of only one row. Column vectors are vertical matrices consisting of only one column.

Matrix subpalette A pop-up menu available on the function palette for entering matrices of up to four-by-four elements. The menu displays the tiny grid of a three-by-four matrix.

MetaFile A Windows picture format, or a graphic in that format, or a file containing a graphic in that format.

Mod A Theorist predefined name that behaves as a function (modulus). It returns the first number mod the second, that is, the remainder from dividing the two given arguments.

Move over A Theorist manipulation used to move an expression from one side of an equation to the other while retaining the logical validity of the equation.

Name A Theorist designation for a mathematical entity consisting of Roman or Greek characters, and possibly special characters.

Notebook A Theorist file. Notebooks are displayed in windows, and contain theories, propositions, and expressions.

Notebook Font A Theorist option that determines the set of font and icon sizes to use for displaying expressions.

Number One or more digits (and possibly an expression used for scientific notation) indicating a numeric value.

Numerical manipulations Theorist manipulations that produce numeric (as opposed to symbolic) results. Calculate is the only numerical manipulation.

Op A Theorist object that represents a mathematical operation (e.g., +, $\sqrt{\ }$, =). Ops join together and enclose one or more expression to create larger, more complex expressions.

Operator The Theorist name for matrix or derivative expressions (or expressions that act like matrices or derivatives), that do not necessarily commute under multiplication.

Orientation icon A Theorist icon displayed in the lower left-hand corner of a three-dimensional graph viewport that indicates which side of the representation of three space is displayed in the viewport.

Palette Part of Theorist's main display. The palette is a rectangular window displayed above a notebook window and is used to enter expressions with the mouse.

Partial derivative A Theorist op that takes derivatives with respect to a variable. It acts as a true "partial derivative" only if the specified variable is included in an appropriate independence declaration.

Pattern expression A Theorist expression used to match the form of other expressions in order to substitute an equivalent expressions for the original.

Pi product A Theorist op that encloses an expression and indicates the product of a series of values that the main argument takes on as a variable steps through a given domain of values. See Summation.

PICT A Macintosh picture format (based on QuickDraw graphic specifications) or a picture in that format, or a file containing a picture in that format.

Plot A Theorist proposition that generates an image in the viewport of a two- or three-dimensional graph theory.

Polar A coordinate system in which the position of a point is determined by the distance from a central point, and the angle of rotation from a fixed line.

Polynomial A mathematical expression consisting of a sum of terms, each of which is the product of a constant and a variable (or variables) raised to various powers.

Power A Theorist op (displayed as an exponent or superscript) that indicates the number of times a base expression is multiplied together.

Predefined behavior A set of fifty-seven Theorist behaviors. Each behavior has an associated name (e.g., sin, cos), but any behavior can be assigned to any name.

Predefined name The Theorist name associated with a predefined behavior.

Principal value The first or primary value that results from an operation that produces more than one correct answer.

Rational function A fraction with polynomial expressions in both the numerator and denominator.

ReManipulation Theorist's ability to perform manipulations again to reconstruct or reevaluate conclusions when their assumptions change. ReManipulation can be manual or automatic, which is the default.

RGB A color encoding system that describes colors by their Red, Green, and Blue components.

Rocket A Theorist editing tool found on the right-hand side of two- and three-dimensional graph theories. Use the Rocket to zoom in (or out) on a graph.

Root A number (or numbers) that, when substituted into an equation for the variable(s) makes the equation a true statement. The numbers 2 and –2 are roots of the equation $x^2 = 4$.

Round A Theorist name that acts as a function returning the integer closest to the given value.

Scalar An expression that represents a single numeric value as distinct from a vector or matrix.

Select A Macintosh operation of choosing, usually with the mouse, part or all of an expression or other object.

Select In A Theorist command that selects individually all subexpressions of a selected expression.

Select Out A Theorist command that selects the next largest enclosing structure of the current selection.

Self-adjoint An expression that is equal to the adjoint of itself.

SemiLog
A Theorist graph type that displays a logarithmic vertical axis and a linear horizontal axis.

Simplify
A Theorist manipulation that performs many actions in an attempt to reduce a selected expression to its most fundamental form.

Statement
A Theorist proposition that contains an expression or an equation. Statements that you enter (assumptions) are displayed with a square icon. Statements that the program generates (conclusions) are displayed with triangular icons.

Substitute
A Theorist manipulation that replaces a given expression with another equivalent expression.

Sum
The result of an addition.

Summation
A Theorist op that is multiplied on to an expression and indicates the sum of a series of values that the main expression takes on as a variable steps through a given domain of values.

Surface
A Theorist plot proposition that displays a surface (based on a selected equation) in a graph theory.

Symbolic manipulations
Theorist manipulations that produce symbolic (as opposed to numeric) results. All manipulations, other than Calculate, are symbolic manipulations.

Taylor series
A Theorist manipulation that produces a Taylor series expansion of a selected expression.

Theory
A Theorist work space containing propositions. Equations that are true in one theory may not be true in another. The main theory is essentially equivalent to a notebook. Case theories are subsets of a notebook as are graph theories.

Transform
A Theorist manipulation that executes whatever transformation rules have been created to execute "upon transform." This manipulation is for extending and programming Theorist.

Transpose
A matrix created from a given matrix by flipping the matrix along its upper-left-to-lower-right diagonal, thereby exchanging the rows for the columns.

Trigonometric functions
Functions of angles defined as ratios of sides of right-angled triangles inscribed in a unit circle. These functions are: sine, cosine, tangent, cotangent, secant and cosecant.

Trigonometry
The branch of mathematics concerned with the ratios of triangles.

User-defined behavior
A Theorist term indicating that the behavior of a name is not one of the predefined behaviors but is, instead, defined by you.

Variable
A Theorist class. Names of class Variable are scalars whose values can change dynamically without your intervention.

Variables palette
One side of the Theorist entry palette, used to enter variables and constants. This palette displays a large x on the left side.

Viewport
Part of a graph theory. The viewport is the window on theoretically infinite two-dimensional and three-dimensional spaces. All plot propositions are displayed in the viewport.

Vector
A one dimensional matrix. All vectors are either row or column vectors, depending on their orientation.

Wildcard sub-palette
The 𝕪 pop-up on the variables palette for entering wildcard variables.

Wildcard variables
Theorist expressions that act as placeholders in an expression. Each wildcard variable matches a specific set of expressions and is used as part of pattern expressions in Substitute manipulations and transformation rules.

Working statement
A marked Theorist statement. Once a statement becomes a working statement (by your designation or the program's), it is used (if needed) by all future manipulations, and for making graphs, without notifying you. Working statements can be either assumptions or conclusions.

Bibliography

Of the many texts available about mathematics and computing, these books were used the most while developing Theorist.

Abramowitz, M., Stegun, I. A., *Handbook of Mathematical Functions,* Dover Publications, Inc., New York, 1972.

The standard book for special functions, it contains extensive formulas and tables for all of the special functions you've ever heard of (except spherical harmonics). It also has lots of useful reference material in other areas of mathematics, including combinatorial and numerical analysis formulas and tables, and Laplace transforms.

Bronshtein, I. N., Semendyayev, K. A., *Handbook of Mathematics,* 20th Edition., English translation, Hirsch, K.A., Van Nostrand Reinhold Company, New York, 1985.

This book, about the size of a small dictionary, contains almost all of the basic applied mathematics you will ever need. The type is tiny and the language is sometimes strange, having been translated from Russian, through German, into English, but the book is an amazing compendium of all of those facts that you knew at one time, but forgot, in addition to many things you never knew, at an affordable price.

Buchburger, B., Collins, G. E., Loos, R., *Computer Algebra: Symbolic and Algebraic Computation,* 2nd Edition, Springer Verlag, New York, 1982.

A comprehensive survey of the state of the art in symbolic algebra research and implementation, circa 1982. Great starting point for research.

Davenport, J. H., Siret, Y., Tournier, E., *Computer Algebra: Systems and Algorithms for Algebraic Computation,* Academic Press, 1988.

A good introduction to how computers do symbolic algebra, including some of the latest algorithms.

Gradshteyn, I. S., Ryzhik, I. M., *Table of Integrals, Series, and Products,* 4th Edition, English translation, Jeffrey, A., Academic Press, Inc., San Diego, California, 1980.

As complete a table of integrals, series, and products as you could want. Over 1000 pages.

Knuth D. E., *The Art of Computer Programming: Volume 2, Seminumerical Algorithms,* 2nd Edition., Addison Wesley Publishing Company, Reading, Mass, 1981.

Donald Knuth must be the world's authority on how to do arithmetic, in addition to many other topics. If you have forgotten how to do polynomial division modulo 13, just pick up this book for a refresher. This book contains many good algorithms for numeric, and especially symbolic, computation.

Press, W.H., Flannery, B. P., Teukolsky, S. A., Vetterling, W. T., *Numerical Recipes in C: The Art of Scientific Computing,* Cambridge University Press, New York, 1988.

This is a great handbook for almost all areas of numerical analysis, and a great starting point for further research. These guys have been in the trenches and know the weapons that work, those that do not, and how and when they fail. Written in a friendly style that keeps you interested.

INDEX

This page and following pages left blank for your notes.