

Bright Software Switzerland

GSymbolix

The Simple Solution to Complex Math.

User's Manual

second printing

© 1990-93 by Bright Software Switzerland

program written by Henrik Gudat
manual written by Henrik Gudat and Janett Schickler

BRIGHT SOFTWARE
P.O. Box 18
4153 Reinach 2
Switzerland



Welcome to GSymbolix

Limited Warranty

This software and manual are sold *as is* without warranties as to performance. Neither Bright Software nor the authors of GSymbolix are liable or responsible to the purchaser for loss or damage caused directly or indirectly by this software product. The user, and not the manufacturer, retailer, or distributor, assumes the entire cost of all necessary servicing, repair, or correction and any incidental or consequential damages.

Copyright

This software product (program and manual) is fully copyrighted and all rights are reserved by Henrik Gudat. Copying, selling, duplicating, or otherwise distributing this product is a violation of the law. The manual may not in part or in whole be copied or transferred to any electronic medium or machine readable form without the prior written permission of Bright Software Switzerland.

Apple, the Apple logo, Apple IIGS, GS, SANE, Finder, and GS/OS are (registered) trademarks of Apple Computer, Inc.
The apple is a fruit and is registered in many cookbooks.

Credits to

the Bright Software staff (especially André...)
Lawrance Schneider
Urs Brunner from MusiComp
Dave Lyons from Apple Computer, Inc.
Dave Hecker from Seven Hills Software

Also many thanks to the non-existent Apple II department of Apple Switzerland

“Make it as simple as possible, but not simpler.” - A.E.

Registration Card

Bright Software congratulates you on your purchase of GSymbolix! This program will help you master complex math problems and will also provide you with years of fun with your computer.

We are working on improvements and are always open for ideas. For this reason, we have included this registration form and would like to hear from you. Please fill out this short questionnaire and mail it to

Bright Software Switzerland
P.O. Box 18
4153 Reinach 2
Switzerland

Once you are a registered owner you will be notified of upgrades and new products. If you don't send us this form, you'll miss out on some good opportunities!

(please print clearly)

Form A2-GS17+

Name (first, last):

Profession:

Full address:

Phone:

Fax:

e-mail address:

Program version: _ . _ _ _ Serial number (see disk label): # _ _ _ _

Purchased from:

Date of purchase:

Where did you first learn about GSymbolix?

Do you own other programs from Bright Software?

Bright Software Switzerland

Do you have any suggestions for changes in the next version of GSymbolix?

What do you like best about the program and/or manual?

What do you like least about the program and/or manual?

Do you intend to buy and/or own other computers?

Which magazines do you read regularly?

Any other comments:

Thank you very much for filling out this questionnaire and mailing it in.

Welcome to GSymbolix



Table of Contents

Introduction

Foreword	0
Introduction	1
If You're in a Hurry.....	1-1
Things You Need to Know	2
Internal Precision and Data Types.....	2-1
Special Features.....	2-2
Conventions in this Manual.....	2-3
Figures.....	2-5
A Quick, Guided Tour	3
Let's Get Started!.....	3-3
Entering a Formula.....	3-3
Manipulating the Formula.....	3-5
Displaying the Formula.....	3-6
Finding Roots.....	3-7
Calculating Integrals or Areas.....	3-8
Calculating with GSymbolix.....	3-9
The Desktop	4-1
Controls	4-1
Window.....	4-1
Check Box.....	4-1
Radio Buttons.....	4-1
Lists.....	4-1
Buttons.....	4-2
Pop-Up Menus.....	4-2
Menu Bar	4-2
The File Menu.....	4-2
The Edit Menu.....	4-3



The Rules Menu.....	4-3
The Formula Menu.....	4-3
The Window Menu.....	4-3
The Extensions Menu.....	4-3
Windows.....	4-4
The Formula Window.....	4-4
The Stack Window.....	4-4
The Functions Window.....	4-4
The Constants/Variables Window.....	4-4
Formula Tables.....	4-5
The Scrapbook.....	4-5
The Evaluate Window.....	4-5
The Solver Window.....	4-5
The Integrate Window.....	4-5
Trigonometric Conversions.....	4-5
The Preferences Window.....	4-5
The 2D Control Window.....	4-5
The 3D Control Window.....	4-6
The 2D/3D Display Window.....	4-6

Tutorial

Numeric Math.....	5
The Stack Window.....	5-2
How the Calculator Works.....	5-2
The Stack.....	5-2
Using the Stack for Calculations.....	5-3
One-Argument Commands.....	5-4
Two-Argument Commands (Dyadic Operators).....	5-4
The "No-Argument" Function.....	5-5
Stack Commands.....	5-5
Variables and Symbolic Constants.....	5-8
Defining New Variables.....	5-8
Deleting Variables.....	5-9
Chain Calculations.....	5-9
Complex Numbers.....	5-10
More Chain Calculations.....	5-12



Symbolic Math - Basics	6
The Formula.....	6-1
Objects or the Structure of a Formula.....	6-2
Complex Formulae.....	6-2
Functions for Complex Math.....	6-3
Entering a Formula.....	6-4
Defining Variables.....	6-5
Checking the Syntax of a Formula.....	6-6
Editing a Formula.....	6-8
Switching Formulae.....	6-9
Nesting Formulae.....	6-10
Numeric Math with Formulae	7
Evaluating a Formula.....	7-1
The Solver.....	7-2
Numeric Integration of Functions.....	7-4
2D Plot Module	8
The 2D Control Window.....	8-1
Labels.....	8-2
Superimposing Graphs.....	8-2
Coloring Graphs.....	8-2
Choose Font.....	8-3
Graphs and Formulae.....	8-3
The Graph.....	8-4
The Graph Menu.....	8-4
Finding Roots.....	8-4
Zoom.....	8-5
Show Slope.....	8-5
Show Coordinates.....	8-5
Show Range.....	8-6
Save Graph.....	8-6
Copy Graph to Clipboard.....	8-6
Print Graph.....	8-6
Purge Graph.....	8-6



3D Plot Module	9
Plotting a Graph.....	9-1
Data Types / Precision.....	9-5
The Graph Menu.....	9-5
Saving Graphs.....	9-6
Copying Graphs to the Clipboard.....	9-6
Print.....	9-6
Symbolic Operations on Formulae	10
Expand.....	10-1
Sort.....	10-2
Collect.....	10-2
Simplify.....	10-3
Trigonometric Conversions.....	10-3
Derive (All).....	10-3
Limitations of the Algorithm.....	10-4
Symbolic and Numeric Results.....	10-4
Block Protection Brackets ("[...]").....	10-6
System Extensions	11
Installing Extensions.....	11-1
Executing Extensions.....	11-1
Loaded or Not Loaded?.....	11-2
Problems with Extensions.....	11-2
Sample Programs.....	11-3
More About Extensions.....	11-5
Display Modes	12



Reference Section

The Menu Bar	A
The Apple Menu.....	A-1
The File Menu.....	A-1
The Edit Menu.....	A-3
The Rules Menu.....	A-4
The Formula Menu.....	A-5
The Windows Menu.....	A-6
Preferences	B
Scientific Functions	C
Installation	D
Installing GSymbolix on a Hard Disk.....	D-1
Installing GSymbolix's Icons.....	D-1
Files Created by GSymbolix.....	D-2

Index

Figures



Chapter 0: **Foreword**

Welcome to GSymbolix 1.7

© 1992 by Bright Software Switzerland

program written by Henrik Gudat
manual written by Henrik Gudat and Janett Schickler

September 1992





Foreword

GSymbolix - an abbreviation for GS-Symbolics - is the first, large commercial program for numeric and symbolic math for Apple IIGS computers. *GSymbolix* closes a big and important software gap and gives the IIGS new functionality. It also has set a new standard in what we call "user friendly".

GSymbolix's kernel, interface and auxiliary routines are melt into one single module which make it impossible to port the program over to other platforms. Of course, the complete program is written in clean and fast assembly code. Almost every part has been rewritten in the course of time and new techniques have allowed for more complex functions.

There are always several ways to write a program: different approaches, from brute force to careful software engineering, produce the desired result - but how?

GSymbolix contains modern algorithms (binary trees, dynamic data structures, routines with linear expenditure) and a complete rule-based system keyed to symbolic math. Although recursion is often an elegant solution for "analyzing" well-defined languages (in a broad sense), *GSymbolix* dispenses with recursion: even evaluating or parsing a formula is done in a self-contained loop. The often not very simple conversion from recursion to iteration is one of the secrets to fast and efficient code.

A word about the Apple IIGS market

It's a funny situation: Apple invests money (more or less intentionally) in improving the system software - but nobody uses it. Most programs for the GS are small utilities or programs that have never been updated. The current situation as well as Apple's strange marketing strategy do not encourage developers to write large applications.

The Apple IIGS is a great computer with an enormous potential. Sure, the hardware doesn't meet the need for speed, but it's especially the software that could stand improvement. Blame Apple when something



works slowly, but don't blame them when it doesn't work at all! Most programs are full of teething troubles and accomplish a certain task with a lot of imagination. Don't look for a simple wordprocessor with auto-indexing or even multiple footnotes - though these are important ingredients for anyone who wants to write a text. The above example is not an exception: there are many other jobs the IIGS simply cannot do because the appropriate software is missing.

Appeal to ...you!

I've spent about two years in creating GSymbolix, and what I've said above *definitely* doesn't mean that GSymbolix is the best and most complete program for the GS. But GSymbolix is a carefully prepared project that has taken a rather long time to complete. Other programmers from Bright Software, especially Yann, helped me develop complex algorithms. And needless to say, almost everybody here tested GSymbolix, suggested changes or gave advice.

To make a long story short, many experienced programmers have spent a considerable amount of time in making GSymbolix what it is today. And when we are talking about time, we shouldn't forget the money.

Please keep in mind that powerful *and* reasonably priced programs such as this one can only continue to find their way into the marketplace if users support the products by buying and not copying them. We'd like to be able to work on additional projects and upgrade existing products, but this we can only do if you don't vote against such efforts by copying and distributing this program.

Remember, in the long run your loyalty will pay off with well, extraordinary products.

Thanks...

Thanks to all those who've sent letters. As you can see, I have implemented most of your suggestions and have enjoyed corresponding with you. I'm open and responsive to your ideas and welcome your mail wholeheartedly.



I appreciate your consideration and hope you now better understand why the IIGS market is in a precarious situation. But now enjoy calculating with GSymbolix.

Henrik Gudat

author of GSymbolix and Gate
chairman of Bright Software
September 29, 1992

PS: Apple Computer encouraged us to change the name of this product because it contains one of their trademarks. For this reason, GSymbolix 1.7 will be the last "GSymbolix". Its successor will be called

Ευρεκα! Symbolix

and will be better, faster, and simpler to use.



Chapter 1: Introduction

Wait!
*Read this chapter and you'll find out
how you can become a GSymbolix expert
in a flash*

Introduction

Welcome to GSymbolix, the first math program on the Apple IIGS that does not distinguish between pure numbers and algebraic expressions. Completely programmed in ultra-fast assembly language using the latest techniques in computer algebra, this unique program sets a new standard in math software for the Apple IIGS. An easy to use desktop interface allows you to concentrate on the math problem and not on the program itself. Enjoy GSymbolix's simplicity, its power, its capabilities.

Some of the Highlights of GSymbolix are:

desktop application • resources allow easy customization • written in assembly language • scientific RPN calculator • formula module with many symbolic commands • object oriented graph modules • integrated formula tables and scrapbook • powerful print commands • superb error handling • and much, much more...

This manual will help you master GSymbolix in a very short period of time. It has three parts:

- In part one (Chapter 3), you'll learn some of GSymbolix's functions and how the different modules are linked together.
- The second part (all remaining chapters) explains all of GSymbolix's features in detail. You should read these chapters after you've completed Chapter 3.
- The last part (the appendices) summarizes commands, functions, and flags. Refer to this section if you have any specific questions.

This manual has been designed to familiarize you with all the necessary functions very quickly. However, it does *not* cover mathematic fundamentals. GSymbolix is a collection of tools that

will allow you to do things more easily and quickly that normally are tedious and time-consuming.

If You're in a Hurry...

You probably just got your copy of GSymbolix and are eager to use it. However, it's crucial that you first read at least the following section "Things You Need to Know" and Chapter 3 "A Quick Tour" because you'll need at least a fundamental knowledge of the structure of this program before you get started. If you try to discover GSymbolix on your own, you'll risk getting frustrated and may need even more time to understand how it works.

So: it pays to read the recommended chapters!

Chapter 2: **Things You Need to Know**


GSymbolix - behind the scenes...



Things You Need to Know

GSymbolix runs on standard Apple IIGS computers with at least 1.5 Mb of free memory (2 to 3 Mb recommended). One 3.5" disk drive is sufficient though we strongly recommend you use a hard disk.

GSymbolix is compatible with system 5.x and later. Please note that menu icons and the built-in bell work only with GS/OS system 6.0 (or later).

 GSymbolix allows you to start it up even if the printer drivers are not correctly installed. If the print commands are disabled and can't be selected, make sure that you've configured your printer software properly.

2.1. Internal Precision and Data Types

GSymbolix uses Apple's SANE (Standard Apple Numerics Environment) tool set for all computations. Instead of writing our own routines for GSymbolix, we decided to make use of SANE because only this tool offers:

- extremely accurate results
- careful rounding (direction selectable)
- gradual underflow detection
- exception handling

In addition, GSymbolix will profit from improvements on the SANE tool or the hardware. Future Apple IIGS models will hopefully already have a built-in floating-point coprocessor, which will increase GSymbolix's performance dramatically.

We've chosen the extended data type for GSymbolix because



the precision of calculations has the highest priority in a math program. The following table describes the range and precision of this data type:

- Size: 10 bytes
- Significant precision: 64 bits or 19 decimal digits
- Maximum value: +/- 1.1E+4932
- Minimum value: +/- 1.9E-4951

The only problem in choosing such a precise data type is that you sacrifice speed. A few functions can be very slow (especially 3D Plot), and adding an accelerator board or a floating-point coprocessor (with SANE patch) to your system might be a good investment. However, we're convinced that in a math program like GSymbolix accuracy is more important than speed, especially if future SANE upgrades will solve this problem.

Special Features

In addition to "normal" numbers, SANE is able to calculate with "NaNs" (NaN is an abbreviation for "Not a Number") and infinities. Although these special "numbers" do not represent a value, SANE allows you to calculate with them.

NaNs are produced when an operation can't be performed due to invalid arguments (example: LN(-1)). SANE distinguishes between 10 different NaNs, which are listed in the following table:

- NaN(1) invalid square root
- NaN(2) invalid addition
- NaN(4) invalid division
- NaN(8) invalid multiplication
- NaN(9) invalid modulo
- NaN(17) invalid ASCII string
- NaN(33) invalid argument to trig routine
- NaN(34) invalid argument to inverse trig routine



NaN(36) invalid argument to log routine
NaN(37) invalid argument to power routine

NaNs are always displayed - as shown in the table - with the NaN code in parentheses. This enables you to locate the invalid expression.

Infinities are used when an operation produces a mathematical infinity (example: $1/0$). As with NaNs, INFs can be used like any other number and always give the correct result. You must, however, distinguish between two infinities, +INF and -INF. The first one stands for an infinite positive number, the second keyword for a infinite negative number.

For more information please refer to the *"Apple Numerics Manual"*.

2.2. Conventions in this Manual

Sometimes it is quite difficult to make clear what button should be pushed or which menu has to be selected. Therefore, the following conventions are used throughout this manual:

FORMULA / PREFERENCES

..means that you should select the menu item PREFERENCES in the FORMULA menu.

[FORMULA / STACK / ENTER]

The brackets indicate that you should press or click on a control (i.e. a button, check box, or radio button). To make the control in question appear, choose STACK from the menu FORMULA. If the window "STACK" is already open (on the desktop), the command [STACK / ENTER] produces the same result. And, if it's clear from the context that the STACK window is meant, only [ENTER] may appear in the text.



{StdP}

The curly braces are used to specify symbolic constants. They are all placed on the left side of the window of constants and variables. Five constants can be accessed via buttons, all others are listed on a pop-up menu (the large, empty button). Clicking on this control opens the library.

<RETURN>

..tells you to press RETURN on the keyboard.

OA

..is an abbreviation for the Apple key (Open Apple). This form is often used when referring to the keyboard equivalent of a menu item. For example, when you read “..then select WINDOW / FORMULA (OA-1)”, you can either select FORMULA from the WINDOW menu using the mouse, or hold down the Apple key and then press <1>. Both actions will have the same effect.

Clicking

..means that you should move the cursor to the indicated control or menu item and push the mouse button once. Don't move the mouse while you are holding down the button.

Pressing

...means the same as clicking.


Double-Clicking


..means you have to push the mouse button twice in rapid succession. You can change the interval in which the IIGS interprets two mouse clicks as a double-click by entering the control panel (OA-control-escape) and opening the submenu “options” (ROM 1.0) or “mouse” (ROM 3.0).




Figures

At the end of this manual you'll find several pages with figures (i.e. screen dumps). All "items" (controls) in a window as well as the windows themselves are individually numbered. *Please remove these pages from the ring binder and have them ready* when reading this manual since there are many references to these illustrations (for example "4.1").

 Such paragraphs inform you of important details you must pay attention to under all circumstances.

 These paragraphs contain tips and additional information.

 New features (introduced in GSymbolix 1.7 or later) are marked like this.



Chapter 3: **A Quick Tour**

Now get ready...

Launching GSymbolix:

- *To start GSymbolix from your working disk, boot into the Finder. Then insert GSymbolix and double-click on the calculator-icon.*
- *To start GSymbolix from your hard disk, boot into the Finder. Create a new directory. Then insert GSymbolix and drag all files into the folder you've just created. (Double-click on the disk icon "GSymbolix", select all files, then drag them into the directory.) Double-click onto the file "GSYMBOLIX" in this folder to start the program.*

*Note: use the original program disk only for making working copies. After making copies (**for yourself only!**), store the original disk in a safe place so if a problem develops with your working copy another can be made.*

Bright Software, P.O. Box 18, 4153 Reinach 2, Switzerland
voice +41 61 711 23 83 fax +41 61 711 52 63
InterNet gudat@avalon.physik.unizh.ch



A Quick, Guided Tour

This chapter will give you an idea of GSymbolix's capabilities. After having finished this guided tour you'll be familiar with most of its functions and will be able to work with GSymbolix on your own. When you have any questions, refer to the corresponding Reference Section where you can find detailed descriptions.

To run GSymbolix, start *GS/OS™* and the *Finder™*. Then insert the GSymbolix program disk and double-click on the calculator icon.

As GSymbolix loads into your IIGS, a short animation will appear on the screen. After a while, the main screen replaces this picture and GSymbolix opens three windows.

At the top of the screen you see the menu bar. Clicking on one of the words on this line opens a "menu". A **menu** is a group of related commands. To select an item from one of the six menus, click on a name in the menu bar and hold down the mouse button. Then move the cursor to the command you want to execute and release the button. If you want to learn more about the menu bar and the desktop interface, please refer to the *Apple IIGS User's Manual*.

Three windows are automatically placed onto the desktop as soon as GSymbolix has been completely loaded into memory. The upper window (see fig. 1) is called the formula window and holds the formula. The formula has to be entered a bit differently from what you might expect since the current version of GSymbolix does not support the international mathematical notation of formulas. Instead, it uses another system found on many calculators and computers: a formula



must be written like a text in a word-processor, i.e. as a string of characters. The formula window is thus like a little word processor: you can cut, copy, paste, clear, and edit any parts of the text that represent a formula. But more about that later.

The window to the right (see fig. 3) is a window with a lot of buttons in it. Each **button** represents a mathematical function (tangent, cosine etc.), and pressing one of them inserts the function in the formula at the current insertion point or, depending on the selection of the controls explained next, executes the function on the calculator.

Two other controls are located in the lower right corner of this window (see fig. 3.1). They are called **radio buttons**. They retain and display an on-or-off setting and are organized into so-called "families". Only one radio button in a family can be on at a time. In this example, the family consists of only two buttons, "Formula" and "Stack". These two controls determine to which window variables or built-in functions and constants are linked. If you select "Formula" (the small circle is then filled out with black color), GSymbolix links them to the formula window and puts the program in symbolic math mode. If "Stack" has been chosen, GSymbolix will calculate like ordinary calculators. In this case, the numeric math mode is active and selecting either a constant, variable, or function stores a copy of the value of the constant or variable in the calculator window or executes the function. *Please make sure that this switch is set to "Formula" before you continue*, otherwise the following examples won't work properly.

The window at the bottom of the screen (see fig. 4) is the window of constants and variables. To the left you can see six buttons (the large one is empty). They give you full access to the built-in library of constants. When you press one of these buttons, their names (like Pi, e or +INF) will be inserted at the insertion point of the formula window, very similar to pressing the buttons in the middle window.

To the right is the section of variables. Two buttons, "Delete" and "Define" (see fig. 4.2, 4.3), let you add and remove variables.



If you add a variable, it will appear in the list to the right (see fig. 4.1). Clicking on a variable in this list highlights or selects the variable. Some menu commands work only when you've selected one or two variables before executing the command.

Let's Get Started!

We'll begin with the following simple formula:

$$FN(x) = (a+b)^2 + 5a^2 - b^2$$

whereby a is a parameter ($a=15$) and b is a variable.

3.1. Entering a Formula

First enter this formula into the topmost window (see fig. 1):

- Click in the content region of the window
- Enter the formula

In fact, there are two ways to enter this formula:

1) Type in all characters via the keyboard (without entering spaces):

$$(a+b)^2+5*a^2-b^2$$

Now select FORMULA / CHECK SYNTAX OF FORMULA (see fig. 10.1). An alert box appears telling you that there is something wrong with the formula you just typed. CHECK SYNTAX OF FORMULA has highlighted the position where GSymbolix stopped parsing the formula.

Obviously, something is wrong with the 'a'. GSymbolix stopped there because it doesn't recognize 'a' as a variable. If you now choose FORMULA / ADD VARIABLE, GSymbolix converts the highlighted text in the active window (in this case 'a') to a variable and adds it to the list of variables. Since you also use a



second variable in the formula, repeat this procedure for 'b': click to the left of one of the *b*'s in the formula, hold down the mouse button and move the cursor over the variable. Then release the mouse button and select FORMULA / ADD VARIABLE again. The list of variables (see fig. 4.1) should now contain the variables *a* and *b*.

However, *a* does have a value ($a=15$) assigned to it. To set *a* to the correct value, click on its entry in the list of variables and push the button DEFINE (see 4.2) to the left of the list. Now enter '15' in the value field and press OK. That's it.

This is the other way to enter the same formula:

2) First define all variables you'll need in your formula. To do so, press DEFINE (see 4.2) in the lower window and enter a name (*a*) for the variable and a value (15). Then push OK. Follow the same steps to define the new variable *b*.

Now let's enter the formula:

- Push the [(.)]-button in the window of functions (see 3)
- Double-click (click the mouse button twice in rapid succession) on 'a' in the list of variables (see 4.1)
- Push the [+]-button in the window of functions
- Double-click on 'b' in the list of variables in the lower window
- Push the red arrow button in the window of functions
- Push the [y^x]-button in the window of functions
- Enter <2> (press "2" on the keyboard)
- Push the red arrow button in the window of functions
- Push the [+]-button in the window of functions
- Enter "5"
- Push the [*]-button in the window of functions
- Double-click on 'a' in the list of variables in the lower window
- Push the [y^x]-button in the window of functions
- Enter <2>
- Push the red arrow button in the window of functions
- Push the [-]-button in the window of functions
- Double-click on 'b' in the list of variables in the lower window
- Push the [y^x]-button in the window of functions



- Enter <2>

Finished! You see that you can enter a formula almost without using the keyboard. It's up to you to decide which way works better for you. Of course, you can also use a combination of both methods.

3.2. Manipulating the Formula

Perhaps you are a mathematician and notice immediately that our formula can be written in a much shorter form. Don't even think any longer about how this could be done, GSymbolix will accomplish this task for you - within seconds.

Because there are some exponents in the formula, it's a good idea to expand everything first. Select RULES/EXPAND (F9). The formula should now look like this:

$$a*a+a*b+b*a+b*b+5*a*a-b*b$$

After executing EXPAND you should normally sort the formula every time. Choose RULES / SORT and after a second you'll see...

$$a*a+a*b+a*b+b*b+5*a*a-b*b$$

Only a single term ($b*a$) has been changed. Once you work with longer and more complex formulae, you'll soon discover how useful this command can be.

Of course, our formula is not yet in its final form. Select RULES / COLLECT and watch how GSymbolix collects the expression:

$$6*a^(2)+2*a*b+0*b^(2)$$

You think this formula could be further simplified? Then execute RULES / SIMPLIFY to get rid of the quite useless term $0*b^(2)$. The upper window should now contain...



$$6 * a ^ { 2 } + 2 * a * b$$

which is equivalent to the original formula!

3.3. Displaying the Formula

Now we want to plot the graph of $6a^2 + 2ab$:

- Highlight the independent variable (in our example b) by clicking on it in the list of variables (☞ 4.1)
- Select FORMULA / PLOT 2D GRAPH (☞ 10.2)

A new window appears. The interval is set to $[-6;6]$ and grid lines and labels are turned on. Press [Autoscaling] (☞ 6.1) to accept these settings and to let GSymbolix calculate the y-range automatically. After a while the graph in the interval $[-6;6]$ will be displayed in another window (☞ 6.6).

Now we want to take a closer look at the graph by zooming in, let's say from -1 to 2 . You can do this by dragging a *selection box* (☞ 6.5) in the system of coordinates:

- Move the cursor to the x-coordinate -1 . (First scroll down the window, then orientate yourself by the labels below the system of coordinates.)
- Click the mouse button and hold it down
- Move the cursor to the x-coordinate 2
- Release the button

i The height of the selection box doesn't matter since GSymbolix will calculate the vertical display range.

Now move the cursor to the yellow *information bar* and hold down the mouse button. A pop-up menu (☞ 6.6) appears offering you several operations on graphs. Select the menu item ZOOM WITH AUTOSCALING (☞ 6.8).

The screen is cleared and the new graph is drawn after a few seconds.



Now we want to know the slope at $x=1$. Move the cursor to that point on the screen (the vertical position does not matter) and click the mouse button. Open the pop-up menu again and choose SHOW SLOPE (⇧ 6.7). GSymbolix displays the slope at $x=1$ in the information bar:

Slope= 3.00000000086e+1

i Because GSymbolix can give you only an approximation of the exact result, this value is very close but probably not equal to 30 (which is the correct result).


To learn more about the many other functions of the plot module turn to the corresponding chapter in the Tutorial.

3.4. Finding Roots

Many mathematical problems can only be solved if you know roots (i.e. points where the formula crosses or touches the x -axis) of a formula. GSymbolix has already built-in a fast and flexible algorithm that finds any root, provided you specify the appropriate start value. In our example, though, it doesn't matter what value you enter.

- Select FORMULA / SOLVE (⇧ 10)
- Press SOLVE several times and watch the value shown above this button.

The value should eventually balance out at -45 (= -4.5e1). This means that GSymbolix can't find a better approximation of this root. Then press CANCEL to close the solver window.

i  The 2D module we have used above also features a solver but does not show intermediate results.



3.5. Calculating Integrals or Areas

Suppose you want to calculate the area between the graph and the x-axis in the interval from -50 to -43, then multiply the result by 1.5 and finally subtract the sine of $\pi/6$. The result is to be stored in 'P'. Or in more mathematical terms:

$$P = (\text{Area of } 6a^2 + 2ab \text{ from } -50 \text{ to } -43) * 1.5 - \text{SIN}(\pi / 6)$$

First you need to calculate the area:

- Select FORMULA / INTEGRATE (⌘ 10)
- Enter -50 in the topmost edit line ("From") and -43 in the box below.
- The field [Initial Accuracy] is set to 20 and indicates that 120 data points will be evaluated. You can leave this field as it is.
- Press AREA. GSymbolix now calculates the area between the graph and the x-axis. GSymbolix first calculates an approximation of the area, then increments INITIAL ACCURACY by one and calculates the area again. The program repeats this loop until you hold down the Open Apple (OA) key.
- After the first loop has been terminated and GSymbolix inputs the results into the blank space at the bottom of the window, press Open Apple to stop this loop. Since we need this value for further calculations, move the area to the calculator by pressing TO STACK.

⚠ If you do not stop GSymbolix by pressing the Apple key after the first loop has been terminated, results of other calculations in the rest of this chapter may differ (+/-0.000001).

Press CANCEL to close the window.



3.6. Calculating with GSymbolix

Until now you've seen only three windows: one for the formula, one that contains all built-in functions, and one that gives you access to constants and variables. However, there is a fourth and important window: the *calculator* in the *stack window* (☞ 2a).

i It's a good idea to close the top window by pressing OA-1 before you continue because your desktop may otherwise get jammed.

First select WINDOWS/STACK, then WINDOWS/STACK COMMANDS. These two windows enable you to calculate "numerically", i.e. similar to pocket calculators. The stack window displays a few numbers (☞ 2.3) and an edit line (☞ 2.1a). The second window you've just opened (☞ 2.4) offers a couple of commands for manipulating, saving, or loading stacks.

A "stack" consists of 21 levels, or planes, and each level can hold exactly one number. How to calculate with the stack and the RPN system is explained later in this manual. For now, just follow the instructions and everything will work fine.

The value at the bottom of the stack window is the area you have calculated earlier in this chapter. The next step to take is to multiply this value by 1.5:

- Enter '1.5' in the edit line (☞ 2.1a)
- Press <Enter> or <Return> on the keyboard

The area we calculated before has just been moved one level higher. '1.5' is now located in the first level of the stack.

How can we multiply these two values?

- Activate the numeric math mode by clicking on the lower radio button ("Stack") of the module link switch in the middle window (☞ 3.1).

- Press [*] in the same window

(If you had not set the module link switch to "Stack", GSymbolix would have inserted the '*' into the formula.)



The result of this multiplication is 652.5, but the value you see on the stack will probably differ slightly from this mathematically correct result. That's because GSymbolix has returned only an approximation of the area (about 435) figured earlier in this chapter.

Finally we have to subtract the sine of $\pi/6$:

- Push the button with the π constant in the lower left hand corner of the screen (☞ 4). A numeric approximation of this symbolic constant is pushed onto the stack.
- Enter '6' in the edit line (☞ 2.1a) and press <Return>.
- Click on [/] in the window of functions.
- Push [SIN] in the same window to get the sine of .523598...
- Press [-] again in the same window.

Our last step is to store the final result in P , a yet undefined variable:

- Press DEFINE in the bottom window (☞ 4.2)
- Name the new variable 'P'
- Press [OK]
- Highlight P in the list of variables (☞ 4.1) by clicking on it once
- Execute [STO] in the stack commands window (☞ 2.4) to move the result to P
- Double-click on P the list of variables to verify that the variable has been set correctly. The result should appear in level 1 of the calculator.

That's it!

We hope this chapter has given you a good idea of GSymbolix's capabilities. Some important functions (Derive, 3D graphs) have not been mentioned in this quick tour but the following chapters will give you a complete and detailed explanation of all features.

Chapter 4: The Desktop

Let's scratch the surface...



Chapter 4: The Desktop

Before you can efficiently work with GSymbolix you need to be familiar with the desktop interface. This chapter explains the different types of controls, the menu bar, and the various windows.

4.1. Controls

Window (☞ 1)

A window is a rectangular area that displays information on the desktop. It can be closed or moved around the desktop. Some windows in GSymbolix have a variable size and a scroll bar to the right.

Check box (☞ 6.2)

A small box associated with an option in a dialog box or window. Checking a box usually activates a certain function. Clicking the check box a second time turns it off again.

Radio buttons (☞ 3.1)

Small circles organized into families. Clicking on a radio button turns off all others of this family.

Lists (☞ 4.1)

Scrollable, vertical arrangements of similar items (variables, for example) on the screen. You can select one item by clicking on it. Using the Apple key and shift, it's possible to highlight several variables at once:

- Holding down the Apple key and clicking on a variable selects/deselects it without clearing other selected variables
- Range selection: clicking on a variable, then holding down



shift and clicking on a second one selects all variables between these two.

Some functions in GSymbolix require that you select or highlight a variable before executing it. If you highlight more variables than the function needs to work properly, GSymbolix takes the top variables in the list first.

Buttons (☞ 6.1)

Controls that execute a certain routine in a program when pressed (by clicking on it). Normally they are used to confirm, cancel, or designate an action.

Pop-Up Menus (☞ 1.2, 6.6, 7.4)

GSymbolix features several "hidden" menus which are opened when clicking in certain regions. For example, holding down the mouse button in the yellow information bar of the formula window opens such a menu. To select a command, hold down the mouse button, highlight the desired item, then release the button.

4.2. Menu Bar

The menu bar is organized into seven sections: File, Edit, Rules, Formula, Window, and Extensions. The menu on the far left is the Apple menu and appears in every desktop application. It gives you access to desk accessories like the *Control Panel NDA* and - in GSymbolix - activates the online help.

i The menu icons appear only if you are using GS/OS system 6.0 or later.

The File Menu (☞ 8)

When you click on this menu, you should see a list of choices like in the figure. You can load or save a formula, mix lists of variables, insert formulas into others, print screens and formulas, or quit GSymbolix.



The Edit Menu

This menu contains a few edit functions (cut, copy, paste, clear, select all), a command which undoes the last symbolic operation, and two functions that transfer data from the symbolic to the numeric module and vice versa.

The Rules Menu (☞ 9)

This is the core of GSymbolix. Several extremely powerful commands that can manipulate formulas are hidden behind this menu. "Bell" lets you turn on or off an audio signal which sounds as soon as GSymbolix has finished a complicated calculation. The last item tells GSymbolix to always sort a formula before simplifying it.

i The bell option is turned on only when using GS/OS system 6.0 or later. Also note that you must have assigned a sound to the "Task Complete" event (in the *Sound Control Device* of the *Control Panel* desk accessory).

The Formula Menu (☞ 10)

Some versatile tools are located in the Formula menu. They allow you to evaluate, solve, integrate a formula, plot functions, change the formula's font, and add variables to the list "on the fly". The item PREFERENCES opens a window which enables you to customize GSymbolix (☞ 13).

The Window Menu (☞ 11)

Selecting "Window" opens a menu that lets you show or hide most of the windows.

The Extensions Menu (☞ 12)

The items in this menu depend on you to add even more functions to the main program. Each command activates a stand-alone program which is able to communicate with GSymbolix.



4.3. Windows

The Formula Window (☞ 1)

This window holds a formula and works like a word-processor. Symbolic operations manipulate the formula directly in this window, as if you had changed it by hand. Note that this window holds the currently active formula. It's possible to have up to eight formulae in memory, but normally you can work with only one formula at a time.

The Stack Window (☞ 2a/2b)

This is GSymbolix's built-in scientific calculator. Although it doesn't use the ordinary algebraic enter system, it's basically a simple calculator, with some program specific improvements.

The Window of Functions (☞ 3)

This window is made up of two parts: standard buttons and two radio buttons. The former represent all built-in functions (like logarithms and trigonometric functions), the two radio buttons determine to what module these functions are actually linked. Setting this switch to "Formula" inserts the corresponding function into the formula when pressing a button, while setting it to "Stack" executes the function on the calculator.

The Window of Constants and Variables (☞ 4)

This gives you access to all variables and to the built-in library of constants. It also helps you to add to or delete from the list of variables (DEFINE and DELETE). What exactly happens when you select a constant or double-click on a variable depends, like in the window explained above, on the setting of the switch in the window of functions (☞ 3.1). If the radio buttons are set to "FORMULA", GSymbolix inserts the symbolic constant or variable in the formula, otherwise it pushes its numeric counterpart onto the calculator's stack.



Mathematics, Geometry, Elements (⌘ WINDOWS menu)

These windows show you an alphabetically sorted list of all elements of the periodic system. Or a table containing all important formulas. By copying values/formulas to the calculator/formula you can use them for your own calculations.

The Scrapbook (⌘ WINDOWS menu)

This is an area where you can write anything you like. Store a note, an entire formula, an important result - everything is allowed. And best of all, GSymbolix saves the scrapbook each time you quit the program, and loads it automatically when you launch GSymbolix again.

The EVALUATE Window (⌘ 5)

This window allows you to evaluate a formula. It offers several choices as to where to store the result.

The SOLVER Window (⌘ FORMULA menu)

Activates the SOLVER routine which scans for roots in the current formula.

The INTEGRATE Window (⌘ FORMULA menu)

Opens a dialog box which lets you calculate integrals and areas of a formula.

Trigonometric Conversions (⌘ RULES menu)

Open this window when you want to replace one trigonometric function (SIN, TAN, COS, COT) with another one.

Preferences (⌘ 13)

This window enables you to turn on and off some of the internal settings.

2D Control Window (⌘ 6)

Displays a window with the current settings of the 3D module.



3D Control Window (☞ 7)

Displays a window with the current settings of the 3D module.

2D / 3D Display Windows (☞ 6.4/WINDOWS menu)

These two windows display graphs. Both feature a “pop-up menu” that can be accessed by clicking in the yellow information bar in the 2D Display Window or in the content region of the 3D Display Window. Note: both menus are not accessible when the windows are empty.

Chapter 5: Numeric Math

$$\mathit{INF} * 0 = \mathit{NaN}$$



Numeric Math

The core of the numeric math module is GSymbolix's built-in scientific RPN (*Reverse Polish Notation*) calculator. It offers:

- 15 stack commands
- 39 functions
- 21 levels (6 visible)
- complex numbers
- 39 numeric constants
- access to up to 100 (complex) variables
- full control over SANE environment and display format
- full integration into other modules

We recommend that you have GSymbolix running while you're working through the following chapters. If you've just started GSymbolix, close the top window (select WINDOW/HIDE FORMULA), otherwise the desktop gets too cluttered. The function of the formula window will be further explained in the next chapter.

An entire window, the "stack" window (Fig.), is dedicated to the calculator. You can bring it to the front by pressing OA-4 or by selecting WINDOW / SHOW STACK from the menu bar. Other windows that should be open now are:

- WINDOW / FUNCTIONS (OA-2)
- WINDOW / CONSTANTS AND VARIABLES (OA-3)
- WINDOW / STACK COMMANDS (no equivalent)

⚠ Before you can perform numeric calculations with built-in functions, constants or variables, you must tell GSymbolix that you want to use the numeric math module. To do this, set the



module link switch in the window of functions (OA-2) to “Stack” (☞ 3.1). If you skip this step, some examples in this chapter won’t work.

5.1. The Stack Window

What you see in the stack window (☞ 2a) is the “level” display. The six numbers you see there (☞ 2.3) represent the “head” of the stack. The remaining but invisible eleven levels can be found on top of this window. There’s also an edit line at the bottom where you can enter values (☞ 2.1a).

☞ The display format of the numbers is, by default, the *auto* format. GSymbolix supports several display formats (scientific, fixed, engineering, and auto mode). See chapter 12 (“Display Modes”) for a detailed explanation.

5.1.1. How the Calculator Works

The calculator uses a special input system commonly called Reverse Polish Notation (*RPN*). This system is much more efficient than the ordinary algebraic system because there is no need for parentheses. The RPN is therefore easier and faster to use when entering very long formulae. If you aren’t yet familiar with this system, it’ll take some time to get used to it. But you’ll soon realize how advantageous this system is.

5.1.2. The Stack

The stack is a group of 21 levels, where each level holds exactly one number. In the stack window you can only see the first six levels of the calculator’s stack, all others are located “on top” of the window (☞ 14).

The stack follows the simple rule of “last in, first out” (*LIFO*). When you enter a value, the previous value moves up to level 2 and no longer has the highest priority. Similarly, if you delete



level 1, all levels above move down one position.

The last value you entered will be used for the next calculation. If a certain action requires two arguments (like adding two values), the second level will also be used.

Since level 1 is always the target level and it may happen that you need a value located somewhere else, GSymbolix offers a couple of stack commands which let you change the order and contents of the stack. They are located in the small window "Stack Commands" (see 2.4).

Only one stack can be active at a time, but you have the opportunity to work with several stacks by saving or loading entire stacks to or from a disk.

To make it even more difficult, newer releases of GSymbolix even support complex numbers on the stack. You'll learn at the end of this chapter how to enter complex numbers and calculate with them.

5.2. Using the Stack for Calculations

Calculations are performed by

- first entering all arguments that are used by the command or function (for example, $\text{SIN}(1.5)$ has one argument, namely 1.5),
- then executing a command or function (SIN in this example)

(How to enter values and execute commands or functions is shown in the examples below.)

The basic concept of the "stack" is the following:

- 1) Commands (like sine, +, or FLOOR) take their arguments from the stack.
- 2) The arguments of such a command are removed from the stack when the command is being executed.
- 3) Results are pushed onto the stack and can be used for further calculations.



5.2.1. One-Argument Commands

Commands that need only one argument to operate take their argument from level 1 of the stack and replace it with the result. Diagram 14.1 shows how the stack changes while executing a one-argument command.

Here is a list of all one-argument functions:

SIN COS TAN ASIN ACOS ATAN SINH COSH TANH ASINH
ACOSH ATANH SEC CSEC COT LN BLOG LOG EXP ALOG
SQ CEILING FLOOR FAC INV NEG ABS IP FP SIGN STO
PICK RE IM

An example for an one-argument function:

Calculate $\text{LOG}(0.00001)$

- Push 0.00001 onto the stack:
Write "0.00001" into the edit line (☞ 2.1a). Press <Return> to copy the value from the edit line to the stack. You should now see the value in level 1 of the stack (☞ 2.3).
- Press on the [LOG] button in the window of functions. After you've executed this function, level 1 is set to -5.

5.2.2. Two-Argument Commands (Dyadic Operators)

As the name already implies this second group of functions takes two arguments (instead of one) from the stack. When the command is being executed, both arguments are removed from the stack. The levels above (level 3 and up) move down one level. The result, which is always a single value, is stored in level 1. (For a diagram see fig 14.2)

The only two-argument functions are:

XROOT Y^X MOD + - / *



An example:

Calculate $(\text{Pi})^{(5)}$

- Click on [Pi] in the window of constants/variables. Level 1 is set to the numeric approximation of Pi.
- Go to the edit line in the stack window and type "5"
- Press <Return>.

After you have entered all the arguments onto the stack, execute the appropriate function:

- Push [Y^X] to calculate 3.14159^5

The result (about 306.02 or $3.0602e2$) is located in level 1 of the stack.

The "No-Argument" Function

Yes, there is also a function that doesn't take any argument from the stack. Pressing [RAN] stores a random value between 1 and $2^{(31)}-2$ in level 1. Before putting the result onto the stack, [RAN] scrolls the entire stack up one level.

⚠ From now on, "entering values on the stack" means that you should activate the stack window (☞ 2a/b) and type the given number into the edit line (☞ 2.1a/b, 2.2). Then press <Return> on the keyboard.

5.3. Stack Commands

There are 15 stack commands which help you manipulate the stack (☞ 2.4):

*Last*

When executing a function (sine, +, SQ and so on), GSymbolix stores the argument in level 1 in a separate register called LAST. So if you've executed a function by mistake, or if you want to use the same argument for another function, press [LAST] to get a copy of the last argument. Note that the name of this button is replaced with the number the LAST register holds as soon as you press a function. So you always know to which value this register is currently set.

Drop

Clears level 1 by moving all levels above down one level.

Clear all

Clears the entire stack by setting all levels to 0. Because this is a very dangerous command it works only when you double-click on this button.

DUP (☞ 14ff.3)

Duplicates level 1 by scrolling the entire stack up one level and then copying level 2 to level 1.

Example: to calculate $3+3$, enter 3, press [DUP], and then [+].

DUP2 (☞ 14ff.4)

Duplicates levels 1 and 2. It moves the entire stack up two levels and then copies levels 3 and 4 to level 1 and 2, respectively.

Swap (☞ 14ff.5)

Switches levels 1 and 2. This command is quite helpful when you want to execute a two-argument function and both arguments are already on the stack but in the wrong order.

Over (☞ 14ff.6)

Moves the entire stack up one level and copies level 3 to level 1. This command is very similar to [Duplicate] but duplicates level 2 and not level 1.

*Up/Down*

These two commands scroll the stack up and down. When you scroll up, level 21 (the highest level) wraps to level 1. When you scroll down, level 1 moves to level 21.

Pick

Retrieves the contents of a level specified in level 1. For example, if level 1 is set to 5, then executing [PICK] would replace the "5" with the value that is stored in level 5.

STO (store in variable)

Stores level 1 in a variable that must be selected *before* you execute this command. See "Variables" below for more information.

RCL (recall variable)

Levels 1-20 are moved to the next higher levels and the contents of the selected variable is copied to level 1. See "Variables" below for more details.

Save

Stores a copy of the current stack in a stack file.

Load

Loads a stack file from a disk. The currently active stack will then be replaced.

Re \leftrightarrow Im

Swaps the imaginary and real parts of the complex number in level 1. How to work with complex stacks is discussed at the end of this chapter.

Copy to edit line

It's possible to copy a value from one of the six visible levels to the edit line - just move the cursor over the number and click the mouse button.



5.4. Variables and Symbolic Constants

The window of variables and constants (OA-3) lets you access built-in constants and already defined variables. When GSymbolix is in the numeric math mode, selecting a constant (☞ 4.4) pushes its numeric approximation onto the stack. Double-clicking on a variable in the list of variables to the right (☞ 4.1) copies the contents of that variable to level 1.

5.4.1. Defining New Variables

Press [Define] (☞ 4.2) in the variable window to define new variables or to change the value of an existing variable. As soon as you push this button, a dialog box appears (☞ 4.5) asking you for a value and a name.

⚠ GSymbolix does not distinguish between variables and parameters. If you want to define a parameter, enter a name and the associated value. When defining a new variable, however, you need not enter a value: just leave the value field as it is.

A name for the variable is obligatory and can be up to 15 characters long. Please *be very careful* when choosing a name: do not start a name with numbers or other characters that might confuse you and/or GSymbolix's parser.

⚠ In case a variable has exactly the same name as a built-in constant, the variable overrides the constant. You can therefore no longer use this constant in a formula. Delete the variable if you intend to use the constant.



5.4.2. Deleting Variables

[Delete] (\rightarrow 4.3) removes one or more variables. First highlight all variables you want to delete, then press [DELETE]. Please refer to chapter 4 ("List control") for information on how to select one or more variables.

5.5. Chain Calculations

The stack is especially helpful when working with long formulas. You can store the result of an operation on the stack and use it as an argument for the next calculation.

⚠ Instead of working through a formula from left to right (as you do with a 'normal', algebraic calculator) always start with the innermost operation.

For example, calculate the formula " $FN(X) = 5 * (Pi + (4 / 2) ^ 3)$ " as shown below:

Calculate 4 / 2:

- Enter 4,2 (Remember: type '4' <Return> '2' <Return>)
- Click on [/] in the window of functions

Calculate 2 ^ 3:

- Enter 3
- [y^x]

Calculate Pi + 8:

- Press [Pi] in window of constants and variables
- [+]

*Calculate 5 * 11.1416*

- Enter 5
- [*]

You see, only nine steps were necessary to enter this formula. With an ordinary calculator it would have taken 13.



Here are some examples of chain calculations and how to enter them:

$$5 + (3 * NA)$$

Enter 3 {NA} (⌨ 4.4) [*] Enter 5 [+]

$$3 - ((6 - 4) ^ 6)$$

Enter 6,6,4 [-] [Swap] [y^x] Enter 3 [Swap] [-]

or:

Enter 3,6,4 [-] Enter 6 [y^x] [-]

i You see, starting with the innermost argument is not always the fastest way, but certainly the safest. If you try to nest too many operations you'll soon lose control over the stack.

$$15 ^ (c - SIN(Pi / 3) - 1 / StdP)$$

Enter 15 {c} [Pi] Enter 3 [/] [-] {StdP} [INV] [-] [y^x]

5.6. Complex Numbers

If you want to calculate with complex numbers on the RPN calculator, you must first set a flag that tells GSymbolix to allow such numbers. Open the window "Preferences" (in the FORMULA menu) and mark the check box "allow complex numbers" (⌨ 13).

GSymbolix then resizes the stack window automatically. To the right of the old stack you'll see a second stack as well as another edit line (⌨ 2b, 2.2).

Entering complex numbers

You enter complex numbers almost the same way as you enter



real values. First go to the right edit line (☞ 2.1b) and enter the real part of the complex number. Then proceed with the imaginary part by typing it into the left text box (☞ 2.2).

Calculating with complex numbers

For you there's no difference when working with complex numbers. For GSymbolix, however, several functions work quite differently.

Let's take a look at the cosine. For real arguments, this function outputs - you guessed it - the cosine. If the argument is complex ($x+iy$), then GSymbolix calculates $\text{COS}(x)*\text{COSH}(y)-i\text{SIN}(x)*\text{SINH}(y)$.

Three functions need to be explained in detail:

ABS returns the hypotenuse (or radius) of z . ($\text{SQRT}(x*x+y*y)$).

NEG returns $-x-iy$.

SIGN returns the unit vector ($r=1$) in the direction of its argument.

⚠ Also note that non-elementary functions affect the real part only. By using the [Re<>Im] button you can manipulate both parts individually.

The imaginary stack in real mode

Though the imaginary stack cannot be seen when complex numbers are disabled, the stack commands (DUP, Drop etc.) will change it anyway. Only scientific functions are restricted to the real stack.



5.7. More Chain Calculations

Here are (more tricky) examples of chain calculations:

$$3 - ((6 - 4) ^ 6)$$

Enter 6,6,4 [-] [Swap] [y^x] Enter 3 [Swap] [-] *or*

Enter 3,6,4 [-] Enter 6 [y^x] [-]

Result: -61

$$15 ^ (Vm - SIN(Pi / 3) - 1 / StdP)$$

Enter 15 {Vm} [Pi] Enter 3 [/] [-] {StdP} [INV] [-] [y^x]

Result: 2.14e25

$$\text{SQRT}(Pi / (2a)) * (\text{COS}(\text{SQ}(b)/a) - \text{SIN}(\text{SQ}(b)/a)) / 2$$

DEFINE variables "a" (a=0.4) and "b" (b=0.909)

Enter 2 [highlight variable a] [RCL] [*] {Pi} [SWAP] [/] [SQRT]
[highlight variable b] [RCL] [SQ] [variable a] [RCL] [/] [COS] [highlight
variable b] [RCL] [SQ] [variable a] [RCL] [/] [SIN] [-] [*] Enter 2 [/]

Result: -1.3425...

$$\text{ABS}(G ^ (.5) - 3.3 ^ (2)) ^ (3/2) / 3$$

[G] Enter .5 [y^x] Enter 3.3 Enter 2 [y^x] [-] [ABS] Enter 3, 2 [/] y^x[]
Enter 3 [/]

Result: 11.978986...

$$e ^ (-at/2) / (3a^2) * (e ^ (3at/2) - \text{COS}(\text{SQRT}(3)at/2) - \text{SQRT}(3) * \text{SIN}(\text{SQRT}(3)at/2))$$

DEFINE variables "a" (a=1.5) and "t" (t=4.88)

[e] [highlight variable a] [RCL] [highlight variable t] [RCL] [*] Enter 2
[/] [NEG] [y^x] [highlight variable a] [RCL] [SQ] Enter 3 [*] [/]
[highlight variable a] [RCL] [highlight variable t] [RCL] Enter 3 [*] [*]
Enter 2 [/] [e] [SWAP] [y^x] [highlight variable a] [RCL] [highlight
variable t] [RCL] Enter 3 [SQRT] [*] [*] Enter 2 [/] [COS] [-] [highlight
variable a] [RCL] [highlight variable t] [RCL] Enter 3 [SQRT] [*] [*]
Enter 2 [/] [SIN] Enter 3 [SQRT] [*] [-] [*]

Result: 223.7297...

Chapter 6: Symbolic Math - Basics

How to work with formulae...



Symbolic Math - Fundamentals

The symbolic math module is probably the most exciting part of GSymbolix. With GSymbolix you can:

- enter formulas of up to 48kB (or more than 17,500 characters) in length
- nest up to 127 parentheses
- nest any number of formulae
- use variables as well as symbolic and numeric constants in a formula

Before working through this chapter you should have started up GSymbolix and opened the following windows:

- WINDOWS / FORMULA (OA-1)
- WINDOWS / FUNCTIONS (OA-2)
- WINDOWS / CONSTANTS AND VARIABLES (OA-3)

All other windows are not needed and can be closed.

⚠ If you intend to use constants, variables, or functions in a formula you must switch GSymbolix to the symbolic mode. To do this now, click on the radio button [FUNCTIONS / FORMULA] (see 3.1). If you skip this step, the examples in this chapter won't work.

6.1. The Formula

The core of GSymbolix is the formula window (OA-1). This mini-word processor holds exactly one formula and offers sophisticated editing capabilities (see 1).



6.1.1. "Objects" or the Structure of a Formula

A formula may consist of any of the following "objects":

- up to 715 numeric constants (i.e. numbers) - such as: 1, 8e-10e, 1.87010E2001
- symbolic constants - such as: e, NA, alpha, Pi
- up to 100 (different) variables
- functions
- characters for formatting ("white space") - such as: return, tab, space
- calls to other formulas: <EquationFile>
- "i" when using complex formulae
- "=" when entering equations

Examples of valid formulas are:

```
SIN( Pi ^ ( 15 / X ) )
-2 * INV( NA XROOT( SQ( X ) ) -.5 )
1 + - + 3 - - X (same as 1-3+X)
FP( CEILING( RAN * 10 ) / 10 )
7 + 3 * i - 4 / i ^ ( 3 )
```

(X is a variable)

Of course, GSymbolix supports equations as well:

```
3 = X
15 / 3 = 7 - i * X * SIN( X )
```

 Complex Formulae

GSymbolix supports complex formulae. A pseudo-constant, "i", has been added for "marking" expressions as complex. GSymbolix manages this marker intelligently, which means that the program processes multiple occurrences of "i" correctly. Here are some examples:



$7+3*i$	evaluates to $7+3i$
$7+3*i*i$	evaluates to $4+0i$
$7*i+3*i*i^2$	evaluates to $-3+7i$

As you can see, one single term may contain several "i"s. Additionally, the power function may be used to raise "i" to any power.

⚠ The power function is the only function that can be applied to "i". GSymbolix evaluates its argument and converts it to an integer. As a result, fractions in expressions such as $i^{(x)}$ may occur *but are ignored*. Consequently, the two expressions $3*i^{(3)}$ and $3*i^{(3.5)}$ are exactly the same.

GSymbolix also distinguishes between $*i$ and $/i$. Take a look at

$3/i*i$	$3/i^{(2)}*i*i$	$3*i^{(5)}/i$
---------	-----------------	---------------

They all produce the same result, namely 3.

Functions for Complex Math

Two functions allow you to "mask" complex expressions: $RE()$ and $IM()$. These functions "split" their argument and output either the real (RE) or imaginary (IM) part. The other part is set to zero.

Both functions cannot be differentiated since they are not elementary.

Example:	Result:
$IM(3*i)$	$3i$
$RE(3*i)$	0
$IM(4i+5*i*i)$	$4i$
$RE(4i+5*i*i)$	-5



6.1.2. Entering a Formula

- ⚠** • GSymbolix is case sensitive, i.e. "SIN" and "sin" or "StdP" and "Stdp" are not the same. This rule applies to all objects.
- All functions except "RAN" are immediately followed by an argument in parentheses. Consider the opened parentheses of these functions as part of the function and *do not separate the keywords and the parentheses with spaces, returns, or tabs*. For example, GSymbolix does not recognize "COS (Pi)" (note the space between "COS" and "(Pi)").

You enter a formula exactly as you would enter a text in a word processor. First make sure the formula window (press OA-1) is on the desktop (it need *not* be the foremost window) and then start entering the formula. However, some objects (functions, symbolic constants and variables) can be entered either directly using the keyboard or by clicking on a button or a list using the mouse:

- Symbolic constants can be inserted into the formula by selecting a constant from the built-in library.
 - To insert functions, push the corresponding buttons in the window of functions.
 - Double-click on a variable in the list of variables (window OA-3) to let GSymbolix insert it into the formula at the insertion point. Of course, this works only if you've already defined the variable before (press [DEFINE] to add variables to the list of variables). See also "Defining Variables" below.
- i** When pushing a function button after having highlighted (selected) characters in the formula, GSymbolix will put the highlighted text into the function's parentheses.

Although you may not immediately see the advantages of this alternative input method, you should consider using it because:



- GSymbolix always inserts the correct number of parentheses into the formula (a frequent error when entering formulas directly via the keyboard),
- there is no chance to misspell functions, constants or variables, and, finally
- you can keep working with the mouse.

6.1.3. Defining Variables

⚠ Before GSymbolix can work with algebraic expressions containing variables, you must have already defined them.

Press [Define] (see 4.2) to define new variables or to change the value of an existing variable. As soon as you push this button, a dialog box appears asking you for a value and a name.

i GSymbolix does not distinguish between variables and parameters. If you define parameters, enter a name and an associated value. When creating a new variable, just ignore the "value" field and specify the name only.

The name of the variable can be up to 15 characters long. Please be very careful when choosing a name: do not start a name with numbers or other characters that might confuse you and/or GSymbolix's parser. If the name you specified contains spaces, they're automatically replaced by underline characters ("_").

When you press [DEFINE] and a variable is highlighted, both edit lines ("value" and "name") are set to the name of that variable and its corresponding value. This is useful when you want to

- change the value of an existing variable (change the value and press [OK])
- duplicate a variable (change the name and press [OK]).

A variable appears only once in the list of variables. If you try to [DEFINE] a variable twice, the old one will be deleted first.



6.1.4. Checking the Syntax of a Formula

After you've entered an expression you'll most likely want to know whether GSymbolix accepts it in that form. Select **FORMULA / CHECK SYNTAX OF FORMULA** to let GSymbolix check the formula. One of the following situations will happen:

GSymbolix shows a syntax error message (see 15ff.8). This means that GSymbolix doesn't know what to do with the highlighted item. Common causes: undefined variables or misspelled functions, variables, or constants.

! Do not forget to put the second argument of the three functions MOD, XROOT and ^ in parentheses.

Example:	Cause:	Correct:
SINE (0 . 5)	unknown keyword	SIN (0 . 5)
12 ^ 3	GSymbolix does not know the function "^^" (but "^(")	12 ^ (3)
FLOOR (. 5)	space between "FLOOR" and ".5"	FLOOR (. 5)
3 XROOT 9	missing parentheses	3 XROOT (9)

GSymbolix gives you a Sense of expression not clear message

The formula you entered makes no sense. Common causes: forgotten items (example: 5+SIN()-3), items inserted by mistake, or incomplete expressions.

! There are situations where multiplication signs normally don't need to be written (for example: 4*a*b can be written as 4ab). Since this is, strictly speaking, against the rules, GSymbolix is not able to insert them in all cases. In some circumstances you may get a *Sense of expression not clear* error message when this happens.



(*GSymbolix shows a Number of open and closed brackets does not match error message* (☞ 15.1).

In this case, GSymbolix can't assign a closed parentheses to each opened one. Example: `SIN (COS (TAN (3))`

GSymbolix does nothing ([CHECK SYNTAX REWRITES FORMULA] is cleared, ☞ 13).

This means that the formula has been entered correctly and has been accepted.

(*GSymbolix rewrites (reformats) the formula* [CHECK SYNTAX REWRITES FORMULA] is set, ☞ 13).

This also means that the formula has been accepted. If the check box [INSERT SPACES WHEN REWRITING FORMULA] is set, GSymbolix inserts a space between two items.

GSymbolix outputs a buffer error

- **Table of numeric constants full:** you tried to exceed the limit of 715 numeric constants
- **Buffer too small to hold entire formula** (☞ 15.2 or 15.6): GSymbolix's internal buffers are too small to hold the formula.

An alert box bad der() expression appears

(You've used the der() pseudo-function incorrectly, i.e. the expression where GSymbolix stopped does not match der(*variable* : *expression*).

When you've entered nested formulas some error messages display the additional information "Last file included was xyz" (whereby xyz is the name of a file). This means that the error occurred somewhere after the <xyz> nest call. (GSymbolix can't highlight the exact position and therefore gives you a "hint" as to where to look for errors.)



6.1.5.Editing a Formula

In the formula window you can enter any formula up to 48kB in length in a word processor-like environment. It supports a number of editing keys which are summarized below:

- Left arrow: moves the insertion point one character to the left
- OA + left arrow: jumps to the beginning of the previous word
- Option + left arrow: moves insertion point to the beginning of the line
- Shift + left arrow: extends the selection back by one character, word (if OA is held down), or line (if option is held down).
- The right arrow key works analogously to the left arrow key
- Up arrow: moves the insertion point up one line
- OA + up arrow: jumps to the beginning of the page
- Option + up arrow: jumps to the beginning of the formula
- Shift + up arrow: extends the selection up by a line or a page (if OA is held down), or to the beginning of the formula (if option is held down)
- The down arrow key works analogously to the up arrow key
- Delete: deletes the character to the left of the insertion point, or removes a selection if text has been highlighted
- Clear: clears the current selection
- Control-F: deletes the character to the right of the insertion point, or removes the current selection of text that has been highlighted
- Control-Y: clears all characters from the insertion point to the end of the line
- Control-X: cuts any selection and puts it onto the clipboard
- Control-C: copies a selection and puts it onto the clipboard
- Control-V: inserts the contents of the clipboard at the insertion point, or replaces a selection with the contents of the clipboard if there is a selection
- Double-click: highlights a word
- Triple-click: highlights an entire line

You can also use the edit menu to cut, copy, paste, or clear a selection. These four commands are equivalent to the



keystrokes control-X, control-Y, control-V and <Clear>.

Pressing OA-A or choosing EDIT / SELECT ALL highlights the entire formula.

f Cut, copy, paste, and clear work in the active window only.

6.2. Switching Formulae

Though you can enter and view only one formula at a time in the window, GSymbolix can manage up to eight formulae simultaneously. Those you can't edit in the formula window are kept in a safe place in memory.

There are two commands for creating a new "document": FILE/NEW and OPEN. NEW creates an empty document. OPEN loads an existing formula.

When you are completely done with a formula, choose the FILE/CLOSE menu item. This deletes the formula (only in memory, not on the disk) and frees the memory that has been allocated for it. If there's also a graph linked to the formula (more about that in the chapter about 2D graphs), GSymbolix erases it as well. But if you need the formula or graph later, don't forget to save them to disk *before* closing the document.

The font of the formula is set to Shaston 8 point by default, but you can load another font at any time. It is, however, not recommended to change the font when working with the formula because the system font, Shaston 8 point, can be displayed much faster than any other font. When you load another font, commands such as Expand, Collect, and Sort will be delayed significantly.

To switch to another formula in memory, move the cursor to the information bar (☞ 1.1). Then push the mouse button and hold it down.

GSymbolix instantly opens a pop-up menu (☞ 1.2) which displays the names of all eight formulae. The checked formula is



the one you can see in the formula window (☞ 1.3). All commands that manipulate, save, load, or graph a formula will use this and only this formula. Only the formula in the window can be changed.

The last item of the pop-up menu is RENAME (☞ 1.4). This command renames the active formula: it opens a dialog box where you can enter any name you want (max. 15 characters) as long as you don't save the formula. If you want to save it to disk, *make sure that the name does not violate the naming rules of the operating system you're using!*

6.3. Nesting Formulae

The nest command is a very powerful and versatile function. Although its function is fairly simple - it inserts a specified formula from a disk at a certain position in the current formula - you'll find this command very useful. Frequently used expressions can be stored in separate files and "included" into other formulae at will, without modifying the original expression. You can even nest expressions in already nested formulae, or more strictly speaking, you can nest as many files as you want. This command not only frees you from typing the same expressions over and over again but also enables you to build libraries of frequently used expressions.

⚠ Never nest a file that is already called by another formula. If you do so, the program will run in an endless loop and will not stop until the internal buffers are full.

The syntax for the nest command is <Filename> whereby "Filename" can also be a partial or full pathname.



Examples:

<1/Equation>

nest file "equation" in the program directory

<Root>

nest "Root" in the current folder

<:Math:Library:Degree>

nest the file "Degree" in the folder "Library" on the disk "Math"

i As soon as you select a menu item from the RULES menu or CHECK SYNTAX OF FORMULA, the nest command is replaced with the contents of the nested file.

If you want to keep the nest command when checking the syntax of your formula, make sure that you have turned off the check box [FORMULA / PREFERENCES / CHECK SYNTAX REWRITES FORMULA] before.

An example of a nest call

Let's assume you want to calculate the surface area of a sphere ($A = 4 * \text{Pi} * r^2$), and you have the following three files on the disk:

- the file "Circumference", containing the formula " $2 * \text{Pi} * r$ ",
- "Area", containing " $\text{<Circumference> * r / 2}$ ",
- "Sphere.Area", containing " $4 * \text{<Area>}$ "

Load the file "Sphere.Area" and then [DEFINE] the variable "r". Now select FORMULA / CHECK SYNTAX OF FORMULA. (If nothing happens, mark the check box [FORMULA / PREFERENCES / CHECK SYNTAX REWRITES FORMULA] and try again.) You'll now see in the formula window:

$4 * 2 * \text{Pi} * r * r / 2$

which is the same as $4 * \text{Pi} * r^2$.



Chapter 7: **Numeric Math with Formulae**

Numeric Math a little bit different...



Numeric Math with Formulae

This chapter shows in detail how to

- evaluate
- solve and
- numerically integrate a formula

All of these functions can be used on a formula stored in the formula window (☞ 1). Most of them produce a result which can be copied to the calculator and used as an argument for further calculations.

⚠ The routines for integrating and solving formulae do not support complex numbers. You can, however, use them even when the formula produces a complex result although they will ignore the imaginary part.

7.1. Evaluating a Formula

FORMULA / EVALUATE (☞ 5) evaluates a formula and stores the result in one of three possible places:

- [DISPLAY RESULT ONLY] writes the result in the window,
- [TO STACK] displays the result and pushes it onto the calculator's stack,
- [TO VARIABLE] displays the result in the window and stores it in the selected (highlighted) variable.

The fourth button, [TO VARIABLE, LOOP], does the same as [TO VARIABLE] but continues evaluating the formula and storing the result in the highlighted variable until OA is held down. (If you press Option, GSymbolix waits until you release



the key again.) This command is especially helpful when working on recursion formulae. An example:

To find roots of a formula, GSymbolix uses the algorithm

$$x = x - \text{FN}(x) / \text{FN}'(x)$$

(Newton)

By continually evaluating this formula and storing the result in x , GSymbolix approximates x to a root. We'll now apply this algorithm to find a root of the function $\text{COS}(X)$:

$$x = x - \text{COS}(x) / \text{NEG}(\text{SIN}(x))$$

($\text{NEG}(\text{SIN}(x))$ or $-\text{SIN}(x)$ is the first derivative of $\text{COS}(x)$)

[Define] the variable "x" (set the value to 2) and enter the formula above into the formula window. Then choose [FORMULA / EVALUATE / TO VARIABLE, LOOP]. After a few seconds GSymbolix will display the number 1.57079 at the bottom of the EVALUATE window (the correct result is $\text{Pi}/2$). Since this value doesn't change any more, you can be sure that GSymbolix has found the best approximation.

7.2. The Solver

SOLVE helps you to find roots of a functions (i.e. points where the function touches or crosses the x-axis). If the current formula is an equation, the SOLVER scans for the closest intersection of the two expressions. It uses Newton's algorithm, an uncomplicated and straightforward method.

Highlight an independent variable in the list of variables. When you now select FORMULA/SOLVE, GSymbolix opens a dialog box with the following items in it:



- an edit line where you can enter a starting value
- the [SOLVE] button which returns the next approximation
- the [LOOP] button which continually returns approximations until you press OA
- [TO STACK], which copies the current value to level 1 of the stack
- [CANCEL] which closes the window

Pushing [LOOP] is the same as pressing [SOLVE] many times. Normally you will use [LOOP], but if you want to write down intermediate steps then you'll probably prefer [SOLVE].

[TO STACK] serves as a link to the calculator. Pressing this button pushes the last approximation (i.e. the value in the edit line) onto the stack so that it can be used for further calculations.

The start value can be of any value, but the closer this value is to a root the faster it can be found.

- i** It's a good idea to take a look at the graph first, estimate a possible root and then use your own estimation as a starting value for the solver.

Interpretation of the Approximation

The following cases may occur:

- *The value doesn't change any more after a while*
GSymbolix has found a root.
- *The value changes quickly but is always between two fixed values*
The function has a local maximum or minimum. Try another start value.
- *The value increases or decreases permanently*
The function approximates to the x-axis but neither touches nor crosses it.

Cases two and three haven't been successful searches and you'll have to try other starting values.

(Also take a look at the root finder routine in the 2D graph module which features automatic interpretation of the results.)



7.3. Numeric Integration of Functions

The last function discussed in this chapter is the integration module. It calculates not only integrals but also areas using an extremely powerful and fast algorithm. You can specify the precision of the calculation and use the result for further operations.

First highlight an independent variable in the list of variables and then choose FORMULA/INTEGRATE from the menu bar. A dialog box with the following items in it appears:

- “from”, which is the lower limit of the integral/area to calculate,
- “to”, which is the upper limit,
- “initial accuracy”, which indicates how many data points have to be calculated (data points = 6 * initial accuracy),
- [INTEGRATE], which integrates a formula in the given interval,
- [AREA], which calculates the area of the formula in the specified range,
- [TO STACK], which copies the last result to the calculator,
- and finally [CANCEL], which closes the window.

The first thing you should do is set the upper and lower limits of the integral/area to be calculated. You can set any limit between -1E4932 and +1E4932.

Then specify the accuracy of the calculation. You can enter into this edit line any positive integer (1,2,3...). The higher the initial accuracy is, the more precise results you'll get - and the longer you'll have to wait for them. This factor is set to 20 by default which is a happy medium between accuracy and execution speed.

i As a rule, “complex” formulae (in the sense of complicated) have to be calculated at higher accuracy factors than simpler ones.



(Pressing either [INTEGRATE] or [AREA] starts the calculation of the integral or the area, respectively. Then GSymbolix displays the result at the bottom of the window, increases “accuracy” by one and calculates the same integral/area again, though at a higher precision. You can stop this endless loop by holding down OA at any time.

[TO STACK] enables you to use the last result for further calculations. It copies it to the calculator from where it can be used as an argument for built-in functions, stored in a variable, or even transferred to the formula.

(Press [CANCEL] to close the window.



Chapter 8: 2D Graphs

Draw Formulae - and more!



2D Plot Module

8.1. The 2D Control Window

2D Plot lets you graph a formula in any interval between -10^{4090} and $+10^{4090}$. Before starting 2D Plot, highlight the independent variable in the list of variables (usually "X"), then select FORMULA/2D PLOT or press OA-G. After you've chosen this function, a new window with the following items will be opened (see 6):

- two edit lines to the left specifying the horizontal range of the graph to be plotted
- two edit lines to the right of the screen which set the lower and upper limit of the vertical range
- a check box named "lock" which lets you superimpose graphs
- another check box, "draw grid", tells GSymbolix to draw thin grid lines, making it easier for you to locate points in the system of coordinates
- a third box titled "draw labels" which labels the x- and y-axes
- [PLOT], which clears the window and draws the graph
- [AUTOSCALING] which does nearly the same as [PLOT] except that it calculates the y-range automatically, ignoring any values in the y-range edit lines,
- [Choose Font], which allows you to change the font,
- "double height", a flag for improving high-density printouts
- and last but not least three color palettes

Under normal circumstances you'll only change the x-range ("x start" and "x end") settings and then press [AUTOSCALING] (see 6.1) which analyzes the graph in the specified interval and scales it so that the graph fills the entire screen.

The autoscaling feature is normally the fastest and easiest way



to obtain a good-looking graph. But it doesn't always produce the desired result if the function has one or several extreme values that would distort the graph too much (i.e. lower the vertical resolution in the remaining part of the graph). In such instances you should set the y-range "manually" by entering appropriate values into the "y start" and "y end" edit lines and then pressing [PLOT] (instead of [AUTOSCALING]).

◆ GSymbolix now always draws both sides of an equation. Previous versions allowed you to draw the left side only but this is no longer possible (and not very useful, either).

Labels

When you press the box "draw labels", GSymbolix will calculate appropriate units for the x and y axes separately and label them to the left and at the bottom of the system of coordinates. The units of the axes are always multiples of ten and divide the axes into 2 to 20 sections.

◆ Superimposing graphs

By checking the check box "lock graph" you can superimpose any number of graphs very easily. Instead of clearing the old graph first, GSymbolix combines both graphs - the new and the old one - into one single picture.

You might find this option useful when comparing graphs and their derivatives.

Of course, you should draw the grid and labels only once because the picture otherwise gets too cluttered.

◆ Coloring graphs

GSymbolix displays three color palettes (see 6.3): one for the left side of an equation, one for the right side, and a third for the axes. By clicking on a color box you can set the colors for each of the three objects.



Choose font

GSymbolix allows you to change the font of the labels. Previously, the program displayed them in Shaston 8 only. Unfortunately printing graphs with the “double vertical height” option (see “Print graphs” below) also doubles the font size of the labels. (GSymbolix can’t prevent the system software from doing so.) Consequently, the labels at the bottom often overlap which is quite an eye-sore. The new version therefore features a function for eliminating these justification problems. If you want to print a graph at “double vertical height”, make sure that the check box “double height” (see 6.2) is also checked. Note that the labels in the 2D Display Window will be much smaller than the chosen font size. But the print routine will draw them correctly and even without overlapping them.

Graphs and Formulae

GSymbolix manages up to eight formulae in memory. You can “connect” or link one graph to each formula. Consequently, GSymbolix allows you to work with up to eight graphs simultaneously.

It’s very important to know that formulae are only loosely linked to formula documents. If you change a formula and want to zoom into a previously calculated graph, GSymbolix can’t remember the original formula and displays an error message. Of course, functions for exporting, purging, printing, and saving graphs will still work.

To switch to another graph, just switch to its formula. When GSymbolix swaps the current formula and the to-be-activated formula, the graphs of the two formulae are also exchanged.

- i** If you want to copy a graph to another formula, just
- save the graph to disk
 - switch to another formula
 - execute `FORMULA / LOAD 2D GRAPH` and load the graph.



8.2. The Graph

As soon as you press [AUTOSCALING] or [PLOT], GSymbolix calculates the graph(s). While evaluating the formula, it displays a “please wait” message in the menu bar.

When the complete graph has been calculated and drawn in memory, the 2D Display Window will open (if not already) and the graph appears (see 6ff.4). You can close, resize, or move this window without losing the graph.

The window consists of two sections: the *information bar* and the *content region*.

The content region shows the system of coordinates. You can select with the cursor...

...either a *single coordinate* by clicking the mouse button once
...or an *area* (selection box) by clicking in the upper left corner of it, holding down the mouse button, moving the cursor to the lower right corner of the box and then releasing the button.

When holding down the mouse button in the yellow information bar, GSymbolix opens a pop-up menu (see 6ff.6) with a couple of commands for analyzing the current graph.

8.3. The graph menu

Finding roots

This item does exactly the same thing as the solver but is easier to use. The leftmost coordinate of the selection or the selected x-coordinate is taken as a starting value. Approximations and the status of the routine appear in the information bar and help you interpret the result:



- **approximating** means that GSymbolix comes closer and closer to a root
- **extremum?** says that GSymbolix has found a local extremum. Try another starting value.
- **bad guess** means that a SANE error occurred (i.e. SANE returned a NaN)
- **found signals** that GSymbolix has found a valid root or intersection.

As with the SOLVER, this command searches for roots when the current formula is a simple algebraic expression. If it's an equation, it scans for the closest intersection.

- **You can interrupt the routine at any time by pressing the mouse button.**

Zoom

Magnifies the current selection. When zooming with autoscaling enabled (see 6ff.8), GSymbolix will stretch or shrink the graph vertically so that it fits into the window.

- **The flags in the 2D Control Window (font, colors, etc.) have global character and affect the zoom function as well. Since this window is a modeless window, you can adjust the attributes (i.e. check boxes and font button) before zooming and GSymbolix will then use the new settings.**

Show Slope

Calculates the slope at the selected x-coordinate (see 6ff.7).

Show Coordinates

Displays the relative position of the cursor in the system of coordinates.



Show range

Shows the display ranges.

Save Graph

Saves the graph. You can load a previously saved graph with the FILE/LOAD 2D GRAPH command.

Because of the object oriented structure of graph files, pictures will be smaller than 4kB though they can be printed at full 300dpi and more.

i The GSymbolix package contains a System Extension for exporting graphs in the Apple Preferred file format.

Copy to clipboard

Stores the graph in the clipboard file. Afterwards, the graph can be pasted into most desktop programs that support object oriented pictures.

Print graph

Prints the current graph. Before the graph is sent to the printer, GSymbolix asks you for a title and font.

Mark the check box "double vertical size" to print the title and the graph in proper proportions.

Purge graph

Erases the graph from memory.

Chapter 9: 3D Graphs

Don't touch them - they aren't real...



3D Plot Module

The 3D Graph module has been part of the GSymbolix 1.6 update. Unlike 2D graphs, only one single 3D graph can be kept in memory. You'll lose a 3D graph as soon as you clear it or calculate a new one.

9.1. Plotting a Graph

The 3D module offers dozens of features. *Don't worry* if you don't understand everything at first glance! Here's how you get your 3D graph a hundred per cent:

1. First enter or load a formula.

The expression must be in the form of $f(x, y) = z$ (for example, "COS(.1*(x*x+y*y))"). As you can see, two independent variables are involved which makes it a bit more complicated.

2. Now tell GSymbolix which variables have to be linked to the x and y axes.

First select the "x" variable in the list of variables (i.e. the variable that will run along the x axis). Then open the 3D CONTROL WINDOW from the WINDOWS menu (☷ 7) and push [X Var:] (☷ 7.2) in the lower left corner.

Linking a variable to the y axis works analogously: select the variable and click on [Y Var:].

To the right of both buttons you should see the descriptors (the names) of the two variables.

- ❗ Deleting variables disconnects the independent variables. You'll have to reassign them.



3. Enter display ranges

Enter the intervals of the independent variables. The x axis will run from X MIN to X MAX, the y axis from Y MIN to Y MAX, and the z axis - you guessed it - from Z MIN to Z MAX. However, the last range, the z interval, may be ignored provided that you let GSymbolix calculate the upper and lower limits (see "autoscaling" below).

4. Specify the resolution

A very important factor is the graph's resolution. GSymbolix allows you to change the resolution in discrete steps from 4x4 to 124x124 blocks (or "tiles"- the graph looks like a distorted chess-board.)

The lowest resolution is 4x4 blocks. This means that GSymbolix calculates 16 blocks which produces only a rough sketch of the graph. However, it takes only a few seconds to calculate the image.

The highest resolution is 124x124 blocks, a rather theoretical limit. The program needs about 30 minutes, and the graph may be as large as 2MB (depending on its attributes). Fortunately this precision is so high that the graph is nothing but a black blotch, but if you want to magnify the graph several times (in another application), you might want to make use of it. But note: the enormous memory requirements will probably prevent you from doing so. (GSymbolix detects out-of-memory errors and cancels the command immediately.)

A fair compromise between speed, memory requirements, and precision are resolutions from 14x14 to 25x25. It depends on the formula mainly: some functions require less accuracy, others more. *Try it!*

5. Set the attributes

All other buttons, check boxes (see 7.1) and scroll bars in this window control the appearance (or "attributes") of a graph:



- **HIDDEN LINES** enables GSymbolix's hidden line algorithm. If it's turned on, the program suppresses lines in the graph that are hidden behind others and aren't visible to you.
- **COLOR FILL** means that the blocks will be filled with a color. The actual color depends on the average "height" of such a block (similar to contour lines in a map).

i This option works only if **HIDDEN LINES** is also checked.

- **DRAW LINES** shows (checked) or hides (not checked) the black lattice-work. Of course, this control is only meaningful if **COLOR FILL** and **HIDDEN LINES** are active.
- If **DRAW AXES** is marked, GSymbolix draws x , y and z axes. *Note that the origin of the three axes is not necessarily at (0/0/0) but pinned to the most negative coordinate of the graph.*
- **DRAW GRID**: GSymbolix calculates appropriate units for the x and y axes and draws thin, gray lines.
- The **COLORS** button opens a window where you can select one of four possible color sets.
- The **PHI/THETA SCROLL BARS** rotate the graph in discrete steps. The sample graph will be updated accordingly.
- The **PERSPECTIVE** scroll bar lets you change the perspective distortions.

Because of the way GSymbolix calculates graphs and stores data in memory, you don't have to recalculate a graph each time you want to change an attribute. Instead of pressing **PLOT** or **AUTOSCALING** (explained below), just push **APPLY** and the program skips the time-consuming process of evaluating the formula many times.

7. Calculate the graph

And last but not least it's time to calculate the graph. You do this by pushing either **PLOT** or **AUTOSCALING**. The first button starts the 3D routines, assuming that you have entered all of the three intervals (x , y , and z). The latter command, however, ignores the z range and scales the graph automatically



by scanning for the most positive/negative z coordinates.

GSymbolix now starts evaluating the formula many times. Because this may take a very long time, the program informs you of what it is doing right now. The messages are:

`Blocks remaining to be calculated: xxxx`

Phase 1: GSymbolix evaluates the formula
xxxx indicates how many times GSymbolix will evaluate the formula.

`Please wait, scaling graph...`

Phase 2: GSymbolix normalizes the graph
Afterwards GSymbolix stretches or shrinks the graph so that it fits into the window.

`Please wait, processing data...`

Phase 3: It records the picture elements
The program draws the picture and stores the picture in a QuickDraw picture data structure.

`Please wait, rendering image in memory...`

Phase 4: GSymbolix updates the off-screen buffer
To accelerate update events, GSymbolix then "plays" back the previously recorded picture and stores a 1:1 copy of the screen image in a separate buffer. This enables GSymbolix to update the 3D window very quickly.

The more attributes you have enabled, the longer it'll take. A graph without hidden line algorithm (and consequently without color fill) requires only a few seconds. More complex graphs may occupy the computer for more than half an hour!



The most important factor is the resolution. You'll have to experiment with this scroll bar: some formulas produce acceptable results only on even resolutions, others need extremely high precision.

- i** The speed depends also on your system. An accelerator card is certainly a good investment. We can't recommend math coprocessors at the moment because they are of very limited use.

9.2. Data Types / Precision

GSymbolix uses different data types in order to calculate graphs as fast as possible.

Formulae are evaluated using SANE's extended data type (10 bytes). Before the result is stored in memory, it is converted to a single value (4 bytes) which saves a lot of memory.

When GSymbolix scales a graph, the single values are retrieved, scaled, and put back as fixed values (4 bytes, fixed point). From now on, all calculations are done without using slow floating-point routines.

The resolution of a fixed point value is $<1.E-5$ which is more than enough for a super hires image. GSymbolix preserves NaNs and infinities during conversions, thus you won't notice any difference in precision.

9.3. The Graph Menu

The 3D module has its own graph menu (see 7.4). It allows you to save, print, copy, and purge graphs.

To open this menu, activate the 3D Graph Display window (menu WINDOWS) and click anywhere in the content region:



Saving Graphs

You can save 3D graphs in any of three possible formats:

- **Save Graph (GSymbolix format)** stores the graph in GSymbolix's own format. Such graphs can be loaded with the FILE / OPEN 3D GRAPH command. Since the program does not save the picture itself but the normalized vectors of the graph, you can apply new attributes.
- **Export (super-hires)** exports a 3D graph in super-hires (bitmap) format. Most paint, DTP and other desktop programs are compatible with this type.
- **Export (object format)** exports a 3D graph in object format. (It's a 1:1 copy of the QuickDraw Picture data structure.) Only very few programs support this type, but because of the many advantages (very small files, unlimited resolution) this will probably soon become a new standard.

(Again: also take a look at the APF Extension that exports graphs in the Apple Preferred picture format)

Copying Graphs to the Clipboard

Select **Copy Graph to Clipboard** and GSymbolix will store a copy of the current graph in the system clipboard. (See 2D graphs for more details.)

Print

Sends a graph to the printer. Because regular printouts have twice the vertical resolution of the screen image, there's a check box that - when marked - stretches the graph vertically. Only double-height printouts show text and graphs in proper proportions.

Chapter 10: Symbolic Operations on Formulae

Eureka!



Symbolic Operations on Formulae

GSymbolix offers several extremely powerful commands that manipulate formulae:

- Expand
- Sort
- Collect
- Simplify
- Eliminate ((..))
- Trigonometric Conversions
- Derive (All)

As soon as you execute one of these functions, the indicator in the formula window's information bar shows you how much memory is used.

i If you want to cancel the commands Expand, Collect, or Derive, press the keys Open Apple-Option-Shift-Control all at once and hold them down until an alert box appears.

10.1. Expand

Expand rewrites a formula by expanding products and powers.

Examples:

$(a+b)^2$

EXPAND returns $a^2+a^2b+b^2a+b^2b$

b^4

EXPAND returns b^2b^2b

$(a+b)(c-d)$

EXPAND returns $c^2a+c^2b-d^2a-d^2b$



10.2. Sort

Sorts the individual components of a formula and puts them in the following order:

- Numeric constants
- Symbolic constants
- "i"
- Expressions containing constants
- Variables
- Expressions containing variables
- Other expressions
- RAN

Examples:

$\text{SIN}(X)*e^{3*X}$
 $c*a+c*b-d*a-d*b$

SORT returns $3*e*X*\text{SIN}(X)$
SORT returns $a*c+b*c-a*d-b*d$

10.3. Collect

Simplifies a formula by collecting like terms. It also evaluates numeric expressions (see "Symbolic and numeric results" below). It does the opposite of the **EXPAND** command (although it doesn't always return exactly the original expression).

Examples:

$15+2*2.5$

COLLECT returns 20

$a+3*a$

COLLECT returns $4*a$

$\text{SIN}(\text{Pi})*X^{(\text{RAN})}-4*\text{NA}*X^{(\text{RAN})}$

COLLECT returns $(\text{SIN}(\text{Pi})-4*\text{NA})*X^{(\text{RAN})}$

$a*a*a$

COLLECT returns $a^{(3)}$

It's highly recommended to sort a formula before executing **COLLECT**, otherwise it may not produce the expected result.



10.4. Simplify

Simplifies a formula by eliminating unnecessary expressions.

Examples:

$3+0*\text{SIN}(3)+5$	SIMPLIFY returns $3+5$
$5+(\text{SIN}(4))$	SIMPLIFY returns $5+\text{SIN}(4)$
$\text{EXP}(\text{LN}(X))$	SIMPLIFY returns X
$i^{(5)}$	SIMPLIFY returns $i^{(1)}$
$i^{(-1)}$	SIMPLIFY returns $i^{(3)}$

ELIMINATE ((..)) is a subcommand of Simplify: it removes parentheses which do not affect the priority of operations. It does not rewrite expressions.

10.5. Trigonometric Conversions

Rewrites trigonometric expressions. Please use the help function for more information about the individual buttons.

Example:

$\text{SIN}(X)$	TRIG CONVERSION returns $\text{TAN}(\text{SQRT}(1+\text{TAN}(X)^{(2)}))$
-----------------	---

10.6. Derive (All)

GSymbolix 1.5 introduced a superfast algorithm for differentiating elementary functions. You can apply this command to either the complete formula or to marked expressions only.

The command DERIVE differentiates only expressions that have been put in "der()" pseudo-functions. The syntax is...



$$\text{der}(\textit{variable} : \textit{simpex})$$

der is the keyword, *variable* is a variable that has already been defined, and *simpex* is a simple algebraic expression. Contrary to the second menu item (DERIVE ALL), you aren't restricted to a single independent variable in a formula. You can use different variables and even nest der-functions at will.

Example:	Result:
der(x: COS(x))-IP(x)	NEG(SIN(x))-IP(x)
der(x: der(y: x*SQ(y)))	2*y
...and so on.	

DERIVE ALL on the other hand is useful when you want to know the derivative of the entire formula. Before executing this command, select the independent variable in the list of variables.

Example:	Result:
SIN(x)=3*x	COS(x)=3
(x has been highlighted in the list)	

Limitations of the Algorithm

Of course, GSymbolix can differentiate elementary functions only (except for $\text{ABS}'(x)=\text{SIGN}(x)$). When the program encounters other functions (MOD, IP, FP, FAC, CEILING, FLOOR, SIGN, RE, IM), it shows an alert box and cancels the command. If you want (for whatever reasons), you can use block protection brackets to skip invalid expressions (see below).

10.7. Symbolic and Numeric Results

GSymbolix lets you use symbolic constants in your formulae. However, these constants represent strictly speaking nothing



but pure numbers.

As already mentioned above, COLLECT evaluates numeric expressions and replaces them with the result. You can tell GSymbolix

- to treat symbolic constants as numbers and convert them to numeric results when a COLLECT command is being executed,
- or to keep them in the formula with their name.

Both possibilities have their advantages and disadvantages. When you convert symbolic constants to numeric results you do not know what happened to them, and sometimes it's very useful to know where a constant has been moved to. On the other hand, formulae will be shortened much more if you let GSymbolix convert them to numbers.

Examples:

You enter the formula $e^*e^*e^*e^*e$

With numeric results enabled: COLLECT returns 148.4131591

With symbolic results enabled: COLLECT returns $e^{(5)}$.

You can select the appropriate mode by setting one of the two radio buttons in the window "Preferences" ([EVALUATE SYMBOLIC CONSTANTS TO NUMERIC/SYMBOLIC RESULTS]).

Note that GSymbolix will convert symbolic constants to numbers only if they're part of an expression consisting of (numeric or symbolic) constants only. For example, COLLECT doesn't change the formula $3+Pi*SIN(X)$ since the sine function can't be converted to a numeric result. But if you change the expression to $3+Pi*SIN(e)$, COLLECT is able to solve the parentheses and returns 4.291.



10.8. Block Protection Brackets ("[...]")

This special kind of parentheses "locks" expressions in formulae. This means the commands `Expand`, `Collect`, `Simplify`, and `Derive` cannot manipulate expressions in such brackets. Note that they're treated like regular parentheses, thus any violations of the syntax will force `GSymbolix` to abort the current command.

For example, protect expressions like the following:

$$\dots + (a+b)^{(9)} + \dots$$

Now choosing `EXPAND` would occupy your computer for at least an hour. (Don't try it: you'll get an "out of memory" message anyway.)

However, putting the dangerous expression in brackets solves the problem. Just rewrite the formula so that it looks like this:

$$\dots + [(a+b)^{(9)}] + \dots$$

and everything will work fine.

i It's possible that `COLLECT` collects two protected but completely identical expressions. An example:

You enter: $4 * [c-d]^{(9)} + [c-d]^{(9)}$
COLLECT returns: $5 * [c-d]^{(9)}$

Chapter 11: System Extensions

Copy & use them. Your very own GSymbolix.



System Extensions

“System Extensions” is a new concept that enables assembly programmers to improve the “kernel” very easily. While the writing of such additional programs requires some skill in programming in assembly language, extensions are very easy to use.

Note that even if you don’t know how to program in assembly language, Bright Software will help you. If you have good ideas for extensions others might be interested in, there’s a good chance that we’ll do the tedious work for you.

11.1. Installing Extensions

Installing them is as simple as copying files. Launch your favorite GS/OS copy program and put a copy of the files into the EXTENSIONS folder. When GSymbolix is being loaded, the program scans this directory for system files and shows their names in the new menu to the right (“Extensions”).

⚠ Only programs that specifically have been written for GSymbolix may be copied to the EXTENSIONS folder. Putting other programs - even with the same file type - into this directory will almost certainly crash GSymbolix.

11.2. Executing Extensions

If you want to execute an extension, just open the EXTENSIONS menu and select it with the mouse. It’s up to the extension what will happen next. If you are not sure what the program will do, go into the help mode (APPLE / HELP), then



choose an extension. A well programmed extension now displays a help screen.

⇒ Most routines display a modal dialog box only. That's because previous versions of GSymbolix did not support desktop items belonging to extensions. This has changed with GSymbolix 1.7: an extension can now open its own windows just as if GSymbolix had opened it.

Loaded or not loaded?

There are two kinds of extensions: temporary and memory-resident programs. The latter category is loaded when you select it for the first time and will be kept in memory until GSymbolix is shut down. On the other hand, temporary extensions will be released immediately after they have passed control back to GSymbolix.

This really doesn't affect you. The only reason why GSymbolix supports temporary extensions is because they can be launched even on very small systems (since they need memory only while they are running).

You can easily determine whether an extension is kept in memory or not: if its menu item is checked, then this program is still in memory.

Problems with Extensions

If an extension causes problems, you can de-install it by executing `EXTENSIONS / SHUTDOWN`. This call closes all programs and frees their memory blocks.



11.3. Sample Programs

GSymbolix comes with several sample programs:

FormulaStatus

...show some information about the formulae on the desktop.

FastVariable

...scans your formula for yet undefined variables. It frees you from defining variables and is therefore a quite useful utility.

Note: all descriptors (i.e. the "names" of the variables) must consist of exactly one character (x, y, a,...).

MiscInfo...

changes the flags in the "Preferences" window and adds new variables to the list.

Skeleton

...is a simple extension for your own projects. It does nothing - except for not crashing...

ExpandCollect

...selects the commands EXPAND, SORT, COLLECT, SIMPLIFY for you.

Taylor

...is probably the most useful extension. It calculates a Taylor/McLaurin approximation of the current formula. Although it could stand improvement, it works quite well as it is.

First enter the formula. Then select the independent variable, choose EXTENSIONS / TAYLOR and press a number between 1 and 9 (the red border indicates that GSymbolix is waiting for a key to be pressed). Afterwards GSymbolix will do some difficult calculations that may take some time. The Taylor approximation will be stored in a new formula document called "Taylor". The new independent variable consists of a "T" followed by the



name of the old variable. For MacLaurin approximations, just set the old independent variable to zero.

3D Animator 1.1

...is the most exciting extension. It calculates animated sequences of 3D graphs. This routine actually does two things at once: it varies a third independent variable which is linked to the time axis and it also allows you to change the view-point while the animation is shown.

First prepare the 3D module as if you were to draw a single graph: enter the display ranges and set the attributes. Before pressing PLOT or AUTOSCALING, select a third independent variable in the list and start the animator.

A modal dialog box appears. The first two controls are edit lines which determine from where to where the third parameter will run.

Then enter the number of frames you wish to calculate. Of course, the more frames you calculate the longer it will take. 20 to 30 frames are good values. Be careful when entering large numbers: each frame requires 32kB of free memory! So first look at the RAM CDev in the Control Panel NDA and divide the free RAM by 32 kB. The result is max. number of frames GSymbolix can store in memory.

After that go to the "phi" and "theta" edit lines. These two controls enable you to change the view-point during the playback. The values must be in degrees.

Finally set or clear the autoscaling check box. If you have specified a z interval in the 3D Control Window, unmark the box, otherwise check it.

Now press CALC. GSymbolix calculates and draws the individual frames and stores them in memory. The bottom line shows information as to how far the program is. If you want to interrupt the animator, just click the mouse button and GSymbolix won't calculate the remaining pictures.

When the text "status: done" appears (or you have pushed the



(mouse button), you can adjust the speed at which the whole sequences will be played in the "delay" edit line. 0 means single step mode, 1 means as fast as possible, 2 and all integers greater than 2 say that GSymbolix will add pauses between two frames. When the check box "loop" is checked, GSymbolix will play the sequence again and again until you press the mouse button.

Press PLAY to view the animation. Provided that you have activated the single step mode (delay=0), press any key to advance to the next frame.

(**△** The buttons CALC and DONE erase a previously calculated animation from memory.

You can even export sequences with the SAVE button. The animator then stores the frames in the \$C1 (super-hires) format. The program automatically adds an index to the file's name (name.000, name.001 etc.).

11.4. More About Extensions

(The main idea behind this new concept was that anybody can easily improve the overall functionality of GSymbolix. We've included a complete description of the structure of extensions in the PROGRAMMERS folder on the program disk. Also, most of the extensions come with fully documented source code. We hope that we can encourage you to write your own utilities. Please give Bright Software a buzz if you get stuck - we'll be happy to answer your technical questions and provide you with tips and suggestions.

We'll be glad to help you distribute your own programs at no charge (under the condition that they are freeware). Please contact Bright Software if you have further questions.



Chapter 12: **Display Modes**

Show the results in the most favourable light...



The Display Format

The display modes controls the format in which GSymbolix displays numbers. Except for some special places mentioned below, the program stores numbers - regardless of the current mode - always as a mantissa (with up to 19 significant digits) with a signed exponent.

To select a display mode, open the window EDIT / DISPLAY FORMAT. The program is divided into four groups: each group can have its own display mode. Note: the settings in the last group, "other", affect the remaining part of GSymbolix (everything except the stack, formula, and graph windows).

GSymbolix supports three display formats: fixed point numbers (FIX), scientific format (SCI), and engineering mode (ENG). A fourth type, AUTO, automatically selects the most suitable type. Both main modes, FIX and SCI, display x digits to the right of the radix mark, where x is any integer between 0 and 12. In addition, the scientific mode adds an exponent to the number. Since the fixed format can display only a very limited range, GSymbolix automatically switches to scientific mode (or engineering format when [ENG] is checked) when a value is too large or too small.

ENG is a slight modification of the scientific format: the program displays one to three digits to the left of the radix mark and an exponent that is a multiple of three.

AUTO is a combination of the above modes: the program selects the appropriate mode for you. Also, GSymbolix does not show trailing zeros when in auto mode. Hence AUTO shows numbers always in the most convenient format.



Examples

value	places	FIX	SCI	ENG	AUTO
1	4	1.0000	1.0000e0	1.0000e0	1
0.5	3	0.500	5.000e-1	500.000e-3	0.5
12300	2	12300.	1.23e4	12.30e3	12300
1.7×10^7	0	1.e7	1.e7	17.e6	1.e7 or 17.e6
infinity	4	INF	INF	INF	INF
0	1	0.0	0.0e0	0.0e0	0

⚠ Important

In most cases, the display format does not affect the precision of the mantissa, i.e. digits that have been truncated so that a number fits a certain display mode are not eliminated.

However, there are exceptions. As a rule, the precision of those numbers *that can be edited* (in edit lines or the formula window) depends on the selected format. *Only those digits that are visible* will be preserved, all others are truncated. Of course, you can switch to another display format (preferably AUTO) before executing a “mode sensitive” task.

Appendix A: **The Menu Bar**

Mathematics à la carte...



Appendix A: The Menu Bar

This chapter gives you a detailed description of all menu choices. Some of them have already been discussed in previous chapters, others haven't.

A.1. The Apple Menu

- About GSymbolix...

Displays a short animation, some information about the people behind GSymbolix, and the folks at Bright Software.

- Help

Activates the help mode. When in help mode, a colored HELP! blinks in the upper right hand corner of the screen, signaling that GSymbolix is waiting for a selection. To learn more about a certain item, select, highlight, or push it (depending on the kind of the item). GSymbolix will then open a window and explain its function.

After you have read the help text, either press OK to leave the help mode or NEXT to get help on another item.

- System...

Displays information about the version you're using.

A.2. The File Menu (⌘ 8)

- New

Opens a new, blank document. Once you've already opened eight formulae, this menu item is disabled.



- **Open Formula**

Loads an equation file with all variables.

GSymbolix installs a new document and retrieves the chosen formula. Once you've already opened eight formulae, this menu item is disabled.

- **Open 2D Graph**

Loads a 2D graph into memory. Since GSymbolix stores such graphs in the widely used QuickDraw picture format, you can actually open any picture of this type.

The graph is displayed in the 2D Display Window.

- **Open 3D Graph**

Loads a 3D graph into memory. GSymbolix uses an internal format which preserves important characteristics only.

Note that loading a GSymbolix 3D graph document doesn't change the graph in the 3D Display Window! It only loads internal parameters and the graph's coordinates into memory. You have to open the 3D Control Window, set the attributes and then press [APPLY] to calculate the picture. Please refer to the chapter "The 3D Plot Module" for more information.

- **Close**

Closes the current formula and erases its 2D graph (if any).

Note: at least one document must be open. If there's only one formula in memory, this menu item is disabled.

- **Save (As)**

Saves the current equation and the list of variables (under a name you specify).

- **Merge Variables**

Merges the list of variables and the variables stored in an equation file. When there are two variables of the same name, the one from the selected file replaces the variable in the current list.



- **Insert Formula**

Inserts a formula at the insertion point. The list of variables isn't affected by this command.

- **Delete**

Deletes a file from disk.

- **Page Setup**

Lets you specify paper size and other specifications.

- **Print Formula**

Prints the current formula like a document in a word processor. By using the menu FORMULA / SET FONT you can print the formula in any font you like.

⚠ The formula will appear exactly as it is displayed on the desktop. Thus, if the window is not as wide as the screen, the right margin on the printed page will be set accordingly.

- **Quit**

Saves the scrapbook and returns to the calling program (normally the Finder).

A.3. The Edit Menu

- **Undo**

Cancels the last symbolic operation.

- **Cut, Copy, Paste**

Moves the highlighted text from a text window (i.e. from the formula window, the formula table, or the scrapbook) to the clipboard or from the clipboard to the formula or to the scrapbook. These commands work only when the target window is the foremost window.

- **Clear**

Clears any highlighted text in the foremost window.



- **Select All**
Highlights the entire formula.
- **Selection To Stack**
Converts any highlighted text (from the formula, the scrapbook or the formula table) to a number and stores it in level 1 of the calculator's stack.
- **Stack To Formula**
Inserts the value in level 1 of the calculator at the insertion point in the formula window.
If the complex mode is turned on (refer to appendix B), GSymbolix copies the whole complex number to the formula.
- **Display Format**
Opens the Display Format window. Please turn to chapter 12 for more information.

A.4. The Rules Menu (☞ 9)

- **Derive (All)**
Calculates the symbolic derivatives of the marked expressions (DERIVE) or of the entire formula (DERIVE ALL). The latter command assumes that you have highlighted the independent variable in the list.
- **Expand**
Rewrites a formula by expanding products and powers.
- **Sort**
Sorts a formula.
- **Collect**
Simplifies a formula by collecting like terms.
- **Simplify**
Simplifies a formula by replacing certain expressions with



shorter, equivalent expressions.

- Eliminate ((..))


Deletes parentheses that are not needed.

- Trigonometric Conversions

Rewrites trigonometric expressions.

- Bell

Turns the bell on or off to signal the end of a symbolic operation. You can install your own bell with the Sound CDev in the Control Panel NDA (assign the sound wave to the event "Task Complete").

 GSymbolix disables the bell when it runs under system 5.x.

- Sort Before Simplify

When this menu item is checked GSymbolix will always execute SORT (see above) before simplifying a formula.

A.5. The Formula Menu (10)

- Check Syntax of Formula

Validates the formula. If there are errors in the formula, Check Syntax shows exactly where and why the errors occurred.

- Evaluate

Opens a window with several options for evaluating the formula.

- Solve

Shows a dialog box which lets you find roots of a simple expression or intersections of a formula.

- Integrate

Calculates integrals and areas of a formula.



- Plot 2D/3D Graph of Formula

Plots the graph of one or two formulas with one (2D) or two (3D) independent variables.

- Add Variable

This option can be used to quickly define new variables in formulas. Highlight the variable's name in the formula (or in another text window) and select Add Variable (OA-+). A new variable will be created with a value of 0. If the same variable already exists, Add Variable initializes it to zero.

- Preferences

Shows the PREFERENCES window.

- Set Font

Changes the font in the formula window. The system font is Shaston 8, but you can use any other font as well.

A.6. The Window Menu (☞ 11)

- Formula

Shows or hides the formula window.

- Functions

Shows or hides the window of functions.

- Constants/Variables

Shows or hides the window of constants and variables.

- Stack

Shows or hides the calculator.

- Stack Commands

Shows the stack commands.

- 2D/3D Display Window

These windows display the graphs that have been calculated



(with the 2D or 3D plot module.

- Elements
Shows the library of elements.
- Mathematics
Shows the math formula table.
- Geometry
Shows the geometry formula table.
- Scrapbook
Shows the notepad.
- Close Front Window
Closes the active window.



Appendix B: **Preferences**

*Everything
you always wanted to know
about these check boxes...*



The Window “Preferences”

The window “Preferences” (Fig. 13) contains many switches and check boxes that let you customize GSymbolix to fit your individual needs. Following is a list with explanations of all controls:

- Check Syntax Rewrites Formula

Cleared: FORMULA/CHECK SYNTAX parses the formula only, but does not rewrite it.

Checked: GSymbolix scans for errors in the formula and rewrites the formula if none were found. How the formula is displayed depends on the check box discussed next.

- Insert spaces when rewriting a formula

Cleared: When an item from the RULES menu or FORMULA/CHECK SYNTAX is executed, GSymbolix clears the formula window and writes the formula again without separating items with a space.

Checked: GSymbolix separates two items by inserting a space between the two. The result is a much easier to read formula.

- Update screen during symbolic operations

Cleared: When you select an item from the RULES menu, GSymbolix shows the final result only.

Checked: GSymbolix shows intermediate steps. Note that a symbolic operation takes much longer when this flag is set.

- Show alert windows

Cleared: GSymbolix doesn't show the alert windows which tell you that you are about to delete something.

Checked: Inform you that you are about to do something dangerous (such as delete a variable).



- Allow complex numbers

Cleared: GSymbolix allows real values only.

Checked: the program accepts complex values and resizes the stack window to make room for a second, imaginary stack. Formulae are evaluated to complex results.

- Auto-route keystrokes

Cleared: keystrokes that can't be processed by a control in the active window are ignored.

Checked: GSymbolix activates either the Formula or the Stack window (depending on the state of switch 3.1) and tries to handle unknown key commands in the formula or stack module.

- Don't clear edit lines in stack window

Cleared: ENTER pushes the values in the edit lines (see 2.1, 2.2) onto the stack *and* then clears them.

Checked: ENTER pushes the values in the edit lines onto the stack but does not change them.

- Evaluate constants to symbolic/numeric results

To symbolic results: constants are kept in the formula.

To numeric results: constants are converted to numbers.

Please refer to Chapter 10 ("Symbolic and Numeric Results") for further information.

- Rounding direction

This group of radio buttons gives you full control over the standard rounding direction:

To-nearest: The value closest to the infinitely precise result

Upward: The value closest to and no less than the infinitely precise result

Downward: The value closest to and no greater than the infinitely precise result

Toward-zero: The value closest to and no greater than the infinitely precise result



i Note that these buttons affect the SANE environment and thus GSymbolix's internal calculations, too. It is therefore highly recommended that you don't change the default settings unless you know what you're doing.

- Halts enabled

These switches tell GSymbolix under which error conditions an execution should be interrupted. If a certain halt is enabled, and such an error occurs, GSymbolix prints the condition that caused the halt and then continues as if nothing has happened.

Possible sources are:

invalid operation: occurs when an operand is invalid for the operation to be performed

underflow: occurs when a floating-point result is both tiny (i.e. its magnitude is smaller than $1.7E-4932$) and, therefore, inexact

overflow: occurs when the result of an operation is larger than $1.1E4932$

divide-by-zero: occurs when an operation on finite operands produces an exact infinite result

inexact: occurs when the rounded result is not identical to the mathematical result

⚠ Enabling halts on inexact results is not recommended because some internal conversions may cause this error even though nothing is wrong.



Appendix C: **Scientific Functions**

*Everything
you always wanted to know
about these buttons...*



Scientific Functions

GSymbolix knows the following functions:

Button:	Function:	Syntax:
+	addition	$Y + X$
-	subtraction	$Y - X$
*	multiplication	$Y * X$
/	division	Y / X
SIN	sine	SIN(X)
COS	cosine	COS(X)
TAN	tangent	TAN(X)
ASIN	arc sine	ASIN(X)
ACOS	arc cosine	ACOS(X)
ATAN	arc tangent	ATAN(X)
ASINH	arc hyperbolic sine	ASINH(X)
ACOSH	arc hyperb. cosine	ACOSH(X)
ATANH	arc hyperb. tangent	ATANH(X)
SEC	secans	SEC(X)
CSEC	cosecans	CSEC(X)
COT	cotangent	COT(X)
LN	natural logarithm	LN(X)
BLOG	binary logarithm	BLOG(X)
LOG	common logarithm	LOG(X)
EXP	natural antilogarithm	EXP(X)
ALOG	common antilogarithm	ALOG(X)
Y^X	power function	Y^X
SQ	square	SQ(X)
SQRT	square root	SQRT(X)
XROOT	xth root of a number	$Y \text{ XROOT}(X)$
INV	inverse	INV(X)
NEG	negative	NEG(X)
ABS	absolute	ABS(X)
SIGN	sign of argument	SIGN(X)



FAC	factorial	FAC(X)
IP	integer part	IP(X)
FP	fractional part	FP(X)
CEILING	smallest integer greater than or equal to	CEILING(X)
FLOOR	greatest integer less than or equal to	FLOOR(X)
MOD	modulo (remainder)	Y MOD(X)
der	derivative	der(X:X)
RE	real part of argument	RE(X+Y*i)
IM	imaginary part of argument	IM(X+Y*i)

- i** When you calculate with the stack, the 'Y' refers to the value in level 2, and 'X' to the value in level 1. The result is always located in level 1.

The functions IM, RE, and der() are not available on the calculator.

If you need more information, take a look at the math formula tables which contain a complete list of all functions, along with examples on how to use them.

Appendix D: **Installation**

How you copy a bunch of files...



Installation

8.1. Installing GSymbolix on a Hard Disk

Moving GSymbolix to a hard disk is as easy as copying a few files.

First boot your favorite copy program. Note that it must be a GS/OS application, because others can't copy "extended" file. The easiest way to accomplish this task is probably to use the Finder™.

1. Create a new subdirectory on your hard disk. It's a good idea to create a second one for your data files at the same time.

2. Copy the files...

- GSymbolix and Intro.Data
 - GSymbolix.Main
 - GSHelp.1, GSHelp.Chem, GSHelp.Geo, and GSHelp.Math
- ...files to your hard disk. That's it.

i If you want to skip the intro, just launch GSymbolix.Main instead of GSymbolix.

8.2. Installing GSymbolix's Icons

GSymbolix supports "rBundles", a new feature of GS/OS system 6.0. To install the icons, just launch the file GSymbolix.Main (instead of GSymbolix). Done!



8.3. Files Created by GSymbolix

GSymbolix creates three file types: formula, stack, and super-hires picture files.

Formula Files (filetype \$45, auxiliary type \$8002)

Whenever you save a formula with the SAVE (AS) command, GSymbolix creates a formula file. Such files start with a formula and end with a list of all variables that were defined when you saved the file. **DO NOT MODIFY THE FORMULA FILES!** They contain pointers which can cause a system crash if they've been changed.

Stack Files (binary files)

..are a copy of the entire stack (i.e. of all 21 levels). They are created as soon as you push [WINDOWS/STACK/SAVE].

Graph Files

GSymbolix allows you to save graphs in many different formats:

- as QuickDraw picture files
- as bitmaps (\$C1 files)
- as Apple Preferred pictures
- as GSymbolix 3D graph files (3D only, filetype \$45/\$8001)

QuickDraw picture files are object oriented and should be used when you intend to print the document (the print quality depends on your printer and printer drivers only). Also, GSymbolix is able to load such files and display them in a window.

Bitmap or super-hires picture files are compatible with almost any paint program. They have a resolution of 640x200 pixels. Printouts are limited to 80x40 dpi.

Apple Preferred files are supported through a System Extension that comes with your package. This utility exports 2D



and 3D graphs and features a “vertical stretch” option for improved print quality in some programs (GraphicWriter™ for example). This file is also a 1:1 copy of the screen image (i.e. a bitmap) but is stored in a compressed format.

GSymbolix 3D graph documents are a replica of the program’s internal data structures. Store 3D graphs in this format when you want to re-load it later and change its attributes (view-point, colors etc.). These files are relatively small (max. 64kB) and allow you to save 3D graphs in any of the formats listed above after retrieving such a file.

And last but not least: the easiest way to export graphs is still the clipboard. Both graph modules copy graphs to the system clipboard so that you can simply paste them into other desktop applications.

We found, however, that two large programs did not support standard QuickDraw pictures. The first one, GraphicWriter™ III 1.1, does not recognize the system clipboard at all (Use the Apple Preferred extension with vertical stretch for best results) though the technical staff told us that this will change in one of the next releases. The second application is AppleWorks GS™: this program crashes when you try to paste 3D graphs with hidden line suppression into the paint module. So make sure this attribute is not enabled when you want to move graphs to AppleWorks GS via the clipboard.



Index

*Even
simple solutions to complex math
need a manual
with an index...*

Your Own Index

keyword

page

keyword

page

Bright Software, P.O. Box 18, 4153 Reinach 2, Switzerland
voice +41 61 711 23 83 fax +41 61 711 52 63
InterNet gudat@avalon.physik.unizh.ch



Index

- 2D Control Window..... 8-1
- 2D Display Window.....8-4
- 3D Animator..... 11-4

- Accuracy of Integration..... 7-4**
- Active Formula.....6-10
- Area, Calculating..... 7-5
- Argument.....5-3ff
- Apple Menu.....A-1
- Apple Preferred Pictures..... D-2
- Apply
 - 3D Attributes..... 9-3
 - Display Modes..... 12-1
- Approximation
 - 2D Graph.....8-5
 - Solver.....7-3
- Attributes
 - 2D Graph.....8-5
 - 3D Graph.....9-2
 - Apply Button.....9-3
- AUTO Display Mode..... 12-1
- Autoscaling
 - 2D Graph..... 8-1
 - 3D Graph.....9-3

- Block Protection []..... 10-6**

- Calculator..... 5-2**
- Calculations (Stack).....5-3
- Case, sensitive..... 6-4
- Chain Calculations (Stack).....5-9
- Clear All (Stack).....5-6
- CLOSE (Formula)..... 6-9
- Collect.....10-2

- Color
 - Fill (3D Graph)..... 9-3
 - of 2D Graphs.....8-2
 - of 3D Graphs..... 9-3
- Complex Formulae..... 6-2
- Complex Numbers
 - Stack..... 5-10
- Controls (Desktop).....4-1
 - Check Box.....4-1
 - Button.....4-2
 - List.....4-1
 - Pop-Up Menu..... 4-2
 - Radio Buttons..... 4-1
 - Window.....4-1
- Conventions in this Manual...2-3
- Coordinates, Show..... 8-5
- Coprocessor..... 2-2
- Copy (Stack).....5-7
- Copy to Clipboard
 - 2D Graph.....8-6
 - 3D Graph.....9-6
- Cues, Visual..... 2-3
- Current Formula..... 6-10

- Data Types (3D Graph).....9-5**
- Defining Variables.....6-5
- DER().....10-4
- Derive (All).....10-3
- Display Ranges (3D Graph)... 9-2
- Down (Stack).....5-7
- Draw Axes (3D Graph)..... 9-3
- Draw Grid (3D Graph).....9-3
- Draw Lines (3D Graph).....9-3
- Drop (Stack)..... 5-6
- DUPLICATE (Stack)..... 5-6



- DUPLICATE2 (Stack)..... 5-6
 Dyadic Operators..... 5-4
- Edit Menu**..... A-3
 Editing a Formula..... 6-8
 Eliminate ((...))..... 10-3
 ENGIneering mode..... 12-1
 Entering a Formula..... 6-4
 Error..... 6-6
 - Bad DER()..... 6-7
 - Brackets..... 6-7
 - Buffer Overflow..... 6-7
 - Nested Files..... 6-7
 - Sense of Expression..... 6-6
 - Syntax Error..... 6-6
- Evaluating a Formula..... 7-1
 Executing Extensions..... 11-1
 Expand..... 10-1
 Extended Data Type..... 2-1
 Extension, Apple Preferred.. D-2
 Extensions, Sample..... 11-3
- Figures**..... 2-5
 File Menu..... A-1
 Files..... D-2
 Finding Roots
 - 2D Graph..... 8-4
 - Solver..... 7-2
- FIX Display Mode..... 12-1
 Floating-Point Coprocessor... 2-2
 Font
 - 2D Graph..... 8-3
 - Formula..... 6-9
- Foreword..... 0-1
Formula..... 6-1
 Formula Menu..... A-5
 Functions (Formula)..... 6-4
 Functions, Scientific..... C-1
- Graph**
 - 2D/Formula links..... 6-9, 8-3
 - Menu, 2D..... 8-4
 - Menu, 3D..... 9-5
- GSymbolix..... 0-1
- Hardware, Required**..... 2-1
 Hidden Lines..... 9-3
- i**..... 6-2
 Icons..... D-1
 IMaginary() (Formula)..... 6-3
 Independent Variables
 - 2D Graph..... 8-1
 - 3D Graph..... 9-1
- Infinity (INF)..... 2-3
 Integration, Numeric..... 7-4
 Interpretation (Solver)..... 7-3
 Intersection
 - 2D Graph..... 8-4
 - Solver..... 7-2
- Intervals (3D Graph)..... 9-3
 Installing
 - Extensions..... 11-1
 - Icons..... D-1
 - Program on Hard Disk... D-1
- Labels (2D Graph)**..... 8-2
 Last (Stack)..... 5-6
 Launching GSymbolix..... 3-1, D-1
 Levels (Stack)..... 5-2
 Limitations of DER()..... 10-4
 Linking Variables to Axes..... 9-1
 Load (Stack)..... 5-7
 Lock (2D Graph)..... 8-2
- Memory (3D Graph)**..... 9-2
 Memory-Resident Extensions
 - 11-2



- Messages (3D Graph)..... 9-4
Manual, about this..... 1-1
Menu (Bar)..... 3-1, 4-2
 Apple Menu..... A-1
 Edit Menu..... 4-3, A-3
 Extensions Menu..... 4-3
 File Menu..... 4-2, A-1
 Formula Menu..... 4-3, A-5
 Rules Menu..... 4-3, A-4
 Windows Menu..... 4-3, A-6
- Name, of Variable**..... 5-8, 6-5
Nesting Formula..... 6-10
NEW (formula)..... 6-9
Not-A-Number (NaN)..... 2-2
Numeric Integration..... 7-4
Numeric Results..... 10-4
- Objects (Formula)**..... 6-2
One-Argument Functions..... 5-4
OPEN (Formula)..... 6-9
Over (Stack)..... 5-6
- Parameter**..... *see Variable*
Perspective (3D Graph)..... 9-3
Pick (Stack)..... 5-7
Plotting a 3D Graph..... 9-1
Point & Click (Formula)..... 6-4
Pop-Up Menu
 Formula..... 6-9
Power Function
 in complex expressions..... 6-3
Precision (3D Graph)..... 9-5
Preferences..... B-1
Printing
 2D Graph..... 8-3, 8-6
 3D Graph..... 9-6
- Problems
 with Extensions..... 11-2
 with Display Modes..... 12-2
 Exporting Graphs..... D-3
 Printing 2D Graphs..... 8-3
Programming Extensions..... 11-5
Protecting Expressions..... 10-6
Purge
 2D Graph..... 8-6
 Extensions..... 11-2
- Radio Buttons**..... 3-2
RAN (Random)..... 5-5, 6-4
Range, Show (2D Graph)..... 8-6
REal() (Formula)..... 6-3
REal<>IMaginary (Stack)..... 5-7
Recursion Formulae..... 7-2
Registration Card..... third page
Resolution (3D Graph).... 9-2, 9-5
Results, Symbolic/Numeric.. 10-4
Rules Menu..... A-4
- S.A.N.E.**..... 2-1
Save
 2D Graph..... 8-6
 3D Graph..... 9-6
 Stack..... 5-7
SCientific Display Format..... 12-1
Scientific Functions..... C-1
Selection (2D Display)..... 8-4
Simplify..... 10-3
Slope, Show..... 8-5
Solver..... 7-2
Sort..... 10-2
Stack..... 5-1ff
 Commands..... 5-5ff
 Concept..... 5-3
 Imaginary..... 5-11
 Levels..... 5-2
 Window..... 5-2



Status of Extensions..... 11-2
 STOrE (Stack)..... 5-7
 Structure (Formula)..... 6-2
 Superimposing (2D Graph).... 8-2
 Swap (Stack)..... 5-6
 Switching
 2D Graph/ Formula..... 8-3
 Formula/ 2D Graph..... 6-9
 Symbolic Constants
 Formula..... 6-4
 Stack..... 5-8
 Symbolic Results..... 10-4
 System Extensions..... 11-1
 System Software..... 2-1

Taylor/McLaurin..... 11-4
 Temporary Extensions..... 11-2
 Trigonometric Conversions.. 10-3
 Two-Argument Functions..... 5-4

Up (Stack)..... 5-7

Variables
 Formula..... 6-4
 Stack..... 5-8
 View-Point..... 9-3

Window (Control)..... 4-4
 2D Control..... 4-5
 2D/3D Display..... 4-6
 3D Control..... 4-6
 Constants/Variables. 3-2, 4-4
 Elements..... 4-5
 Evaluate..... 4-5
 Formula..... 3-1, 4-4
 Functions..... 3-2, 4-4
 Geometry..... 4-5
 Integrate..... 4-5
 Mathematics..... 4-5

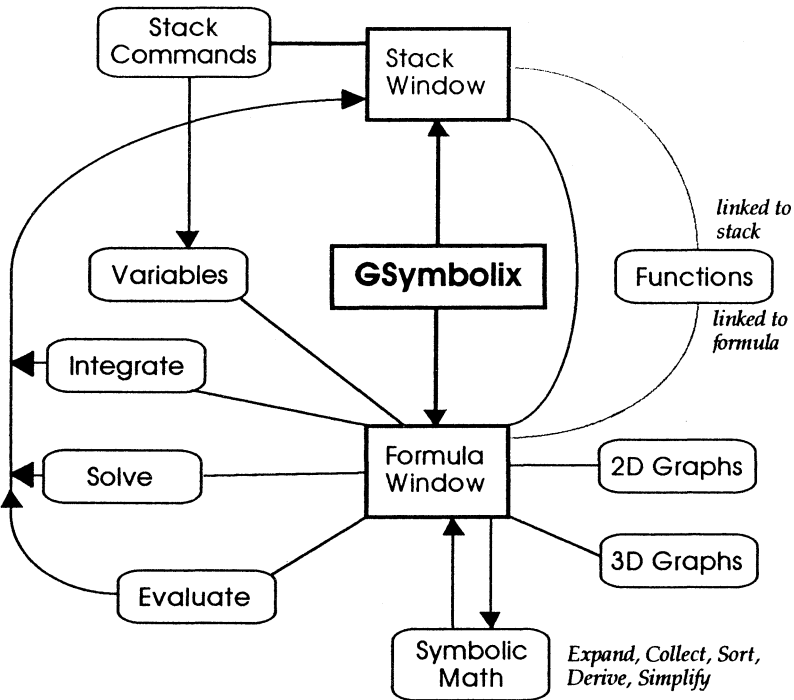
Window, continued
 Preferences..... 4-5, B-1
 Solver..... 4-5
 Stack..... 4-4
 (see also Stack Window)
 Trigonom. Conversions... 4-5
 Windows Menu..... A-6

X-Range
 2D Graph..... 8-1

Zoom (2D Graph)..... 8-5

Figure 0: Tool Dependencies

Pathways and Dependencies in GSymbolix



(Small pathways, auxiliary functions and extensions not shown. Extensions can access any tool and/or add new main modules.)

(

(

(

(

Figure 1: The Formula Window

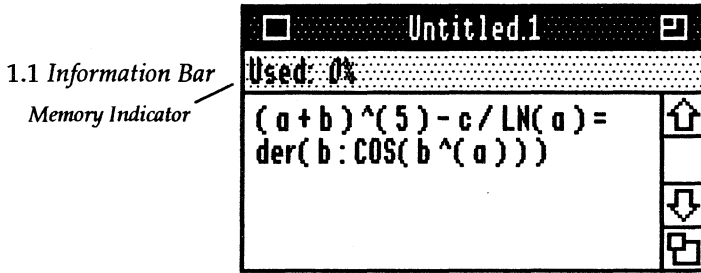
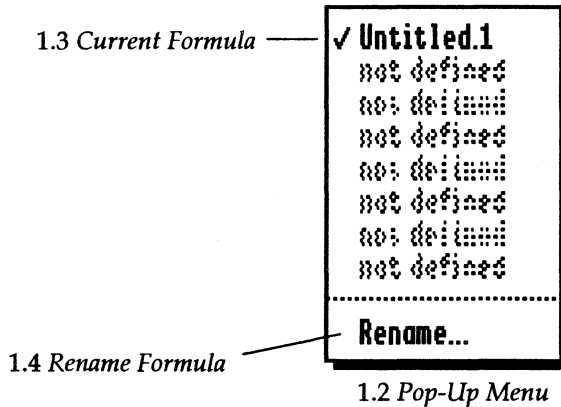


Figure 1: The Formula Window



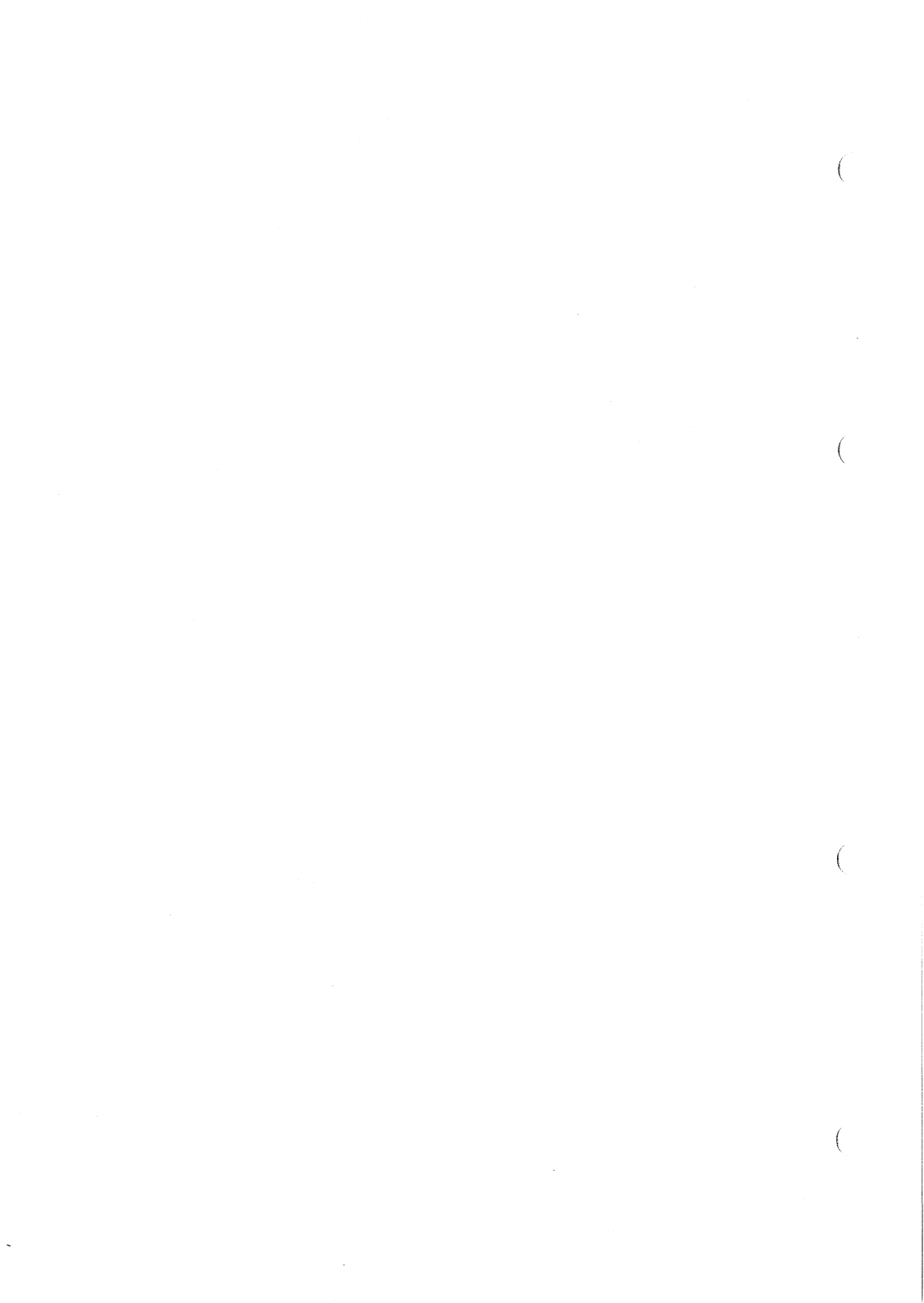


Figure 2: The Stack Window

Figure 2a: *The Stack Window (in real mode)*

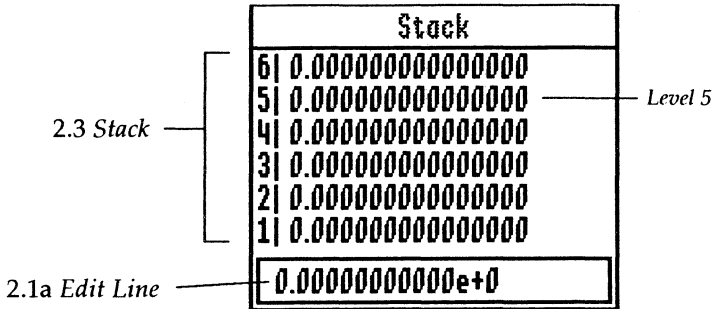
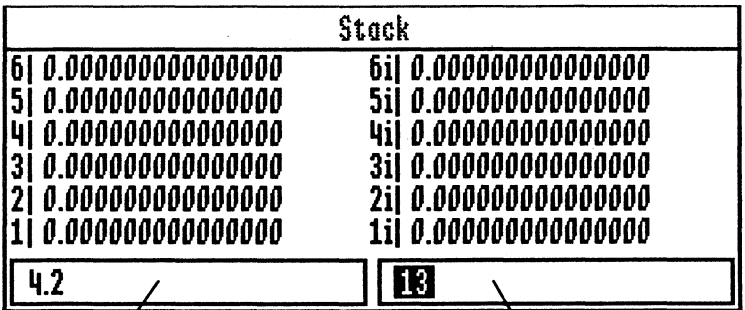


Figure 2b: *The Stack Window (in complex mode)*



2.1b Edit Line (for real parts)

2.2 Edit Line (for imaginary parts)

Stack Commands			
DUP	STO	Save	Up
DUP2	RCL	Load	Down
Over	Swap	Re<>Im	Drop
Last			Clear

2.4 Stack Commands

(

(

(

(

Figure 3: The Functions Window

Figure 3: The Functions Window

Functions			
SIN	COS	TAN	
SINH	COSH	TANH	
ASIN	ACOS	ATAN	
ASINH	ACOSH	ATANH	
SEC	COT	CSEC	
LN	BLOG	LOG	
EXP	y^x	ALOG	
XROOT	SQ	SQRT	
MOD	FAC	INV	
NEG	ABS	RAN	
IP	FP	SIGN	
CEIL	FLOOR	der	
+	-	*	/
=	↗	(...)	[...]
	Formula		
	Stack		

elementary/scientific functions
non-elementary functions
 "derive()" pseudo-function
 3.1 Module Link Switch

(

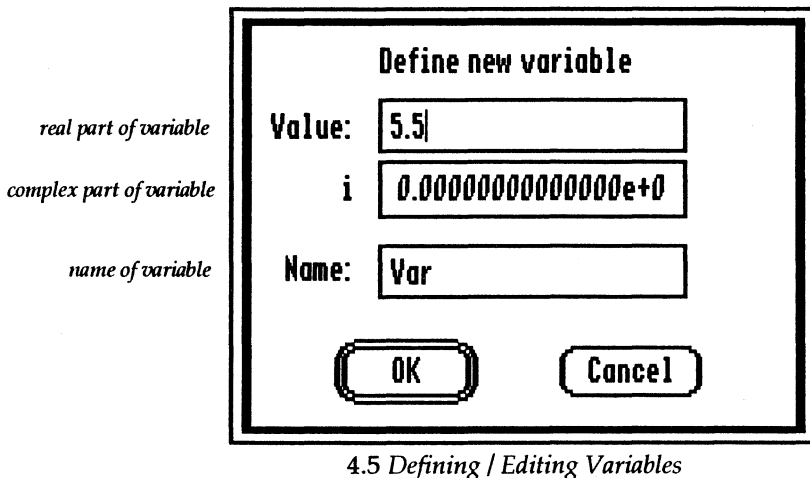
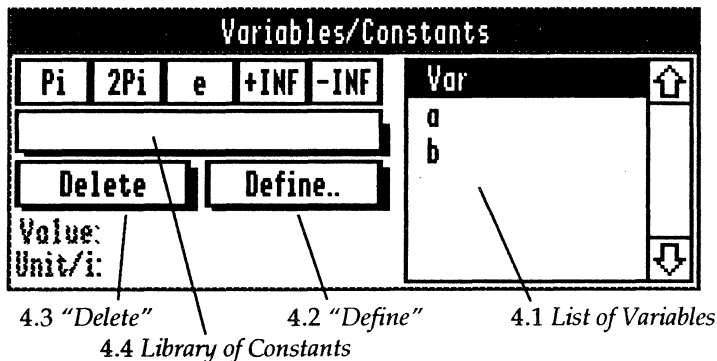
(

(

(

Figure 4: The Variables/Constants Window

Figure 4: The Variables/Constants Window

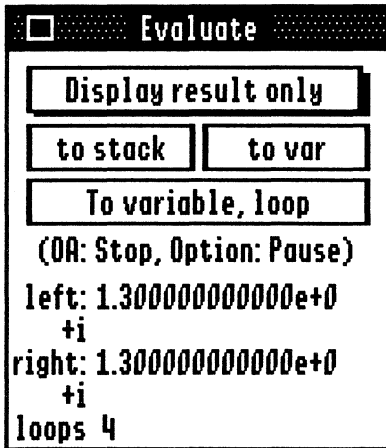


4.5 Defining / Editing Variables



Figure 5: The Evaluate Window

Figure 5: *The Evaluate Window*



real part] *result of left side*
imaginary part] *of equation*

real part] *result of right side*
imaginary part] *of equation*

counter for recursion formulae

(

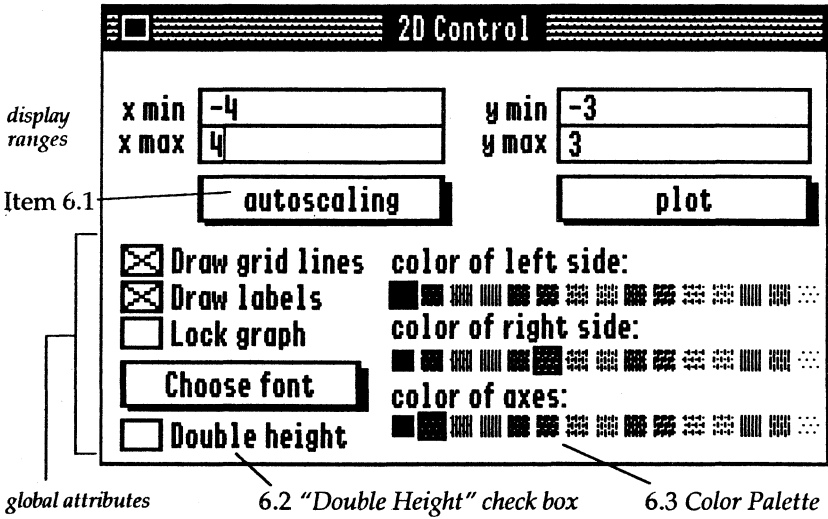
(

(

(

Figure 6: The 2D Control Window

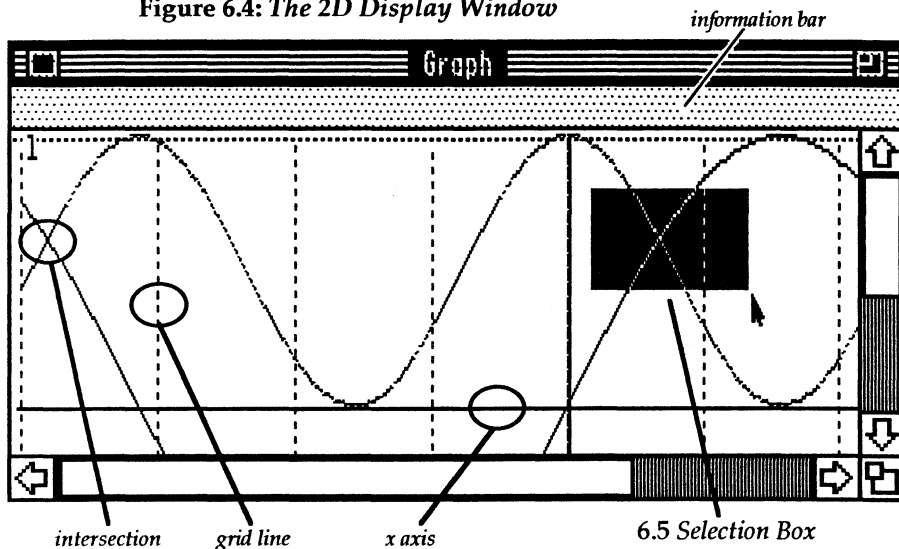
Figure 6: The 2D Control Window



continued on next page

Figure 6 (cont.): The 2D Display Window

Figure 6.4: The 2D Display Window

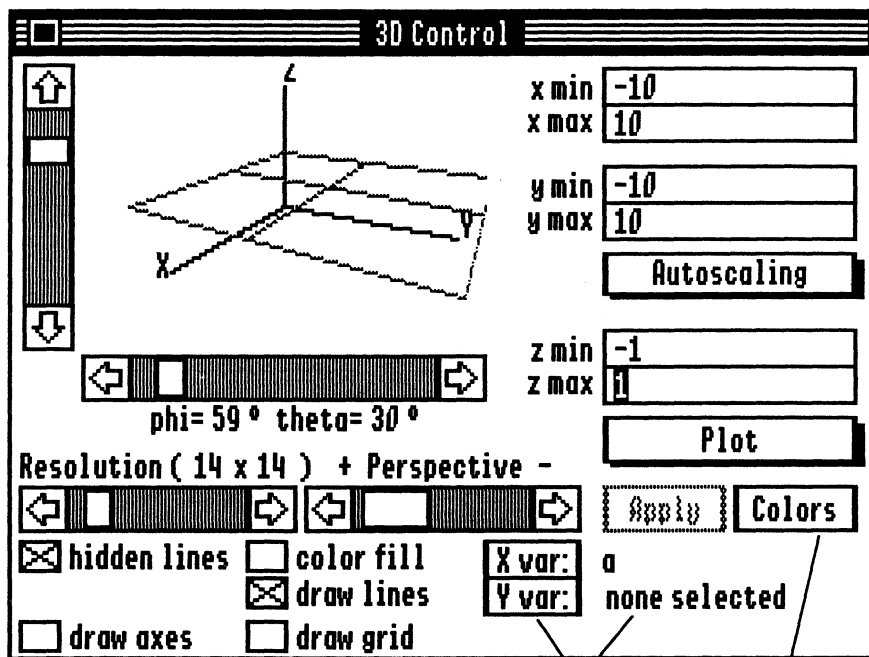


- | | |
|--------------------------------|----------------------------|
| Find next root | |
| Zoom with autoscaling | 6.8 "Zoom" Menu Item |
| Zoom manually | |
| Show slope | 6.7 "Show Slope" Menu Item |
| Show coordinates | |
| Show range | |
| Save graph | |
| Copy graph to clipboard | |
| Print graph | |
| Purge graph | |

6.6 2D Pop-Up Menu

Figure 7: The 3D Control Window

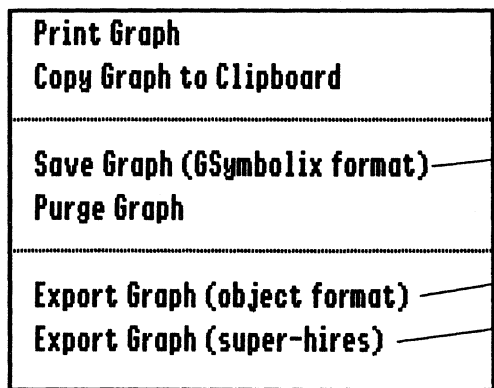
Figure 7: The 3D Control Window



7.1 Global Flags

7.2 Variable-to-Axis Links

7.3 3D Color Palettes



Click into content region of 3D Display Window to open menu (works only if graph has been calculated before)

Save in GSymbolix's own vector format (press Apply in window above to view graph after retrieving it)

PICT format

SHR (\$C1) format

7.4 3D Pop-Up Menu

(

(

(

(

Figure 8: The File Menu

File	
New	⌘N <i>opens an empty document</i>
Open Formula...	⌘O <i>loads a formula from disk</i>
Open 2D Graph...	<i>loads a PICT file and displays it in the 2D Display Window</i>
Open 3D Graph...	<i>loads a 3D graph file. Press Apply (Fig. 7) to view the graph.</i>
Close	⌘K <i>purges the current formula & graph</i>
Save As...	<i>saves the current formula</i>
Save	⌘S
<hr/>	
Merge Variables	
Insert Formula	
Delete File	
<hr/>	
Page Setup	
Print Formula	
<hr/>	
Quit	

Figure 8: The File Menu



Figure 9: The Rules Menu

Rules	
Derive	⌘D
Derive All	
<hr/>	
Expand	⌘E
Sort	⌘R
Collect	⌘L
<hr/>	
Eliminate ((..))	⌘C
Simplify	⌘Y
Trig Conversions	⌘T
<hr/>	
Sound	
Sort Before Simplify	

Figure 9: *The Rules Menu*

(

(

(

(

Figure 10: The Formula Menu

Formula		
Check Syntax of Formula	⌘!	10.1 "Check Syntax"
Evaluate	⌘9	
Solve	⌘=	
Integrate	⌘I	
<hr/>		
Plot 2D Graph	⌘G	10.2 2D Control Window
Plot 3D Graph		10.3 3D Control Window
<hr/>		
Add Variable	⌘+	
Preferences	⌘P	<i>opens the window "Preferences"</i>
Set Font	⌘F	

Figure 10: The Formula Menu

(

(

(

(

Figure 11: The Windows Menu

Windows		
Hide Formula	⌘1	<i>hides / shows main windows</i>
Hide Functions	⌘2	
Hide Constants/Variables	⌘3	
Show Stack	⌘4	
Stack Commands		<i>shows buttons for stack maintenance</i>
2D Graph Display		<i>shows current 2D graph</i>
3D Graph Display		<i>shows current 3D graph</i>
<hr/>		
Mathematics	⌘5	<i>shows formula tables and notepad</i>
Geometry	⌘6	
Elements	⌘7	
Scrapbook	⌘8	
<hr/>		
Close Front Window	⌘W	

Figure 11: *The Windows Menu*

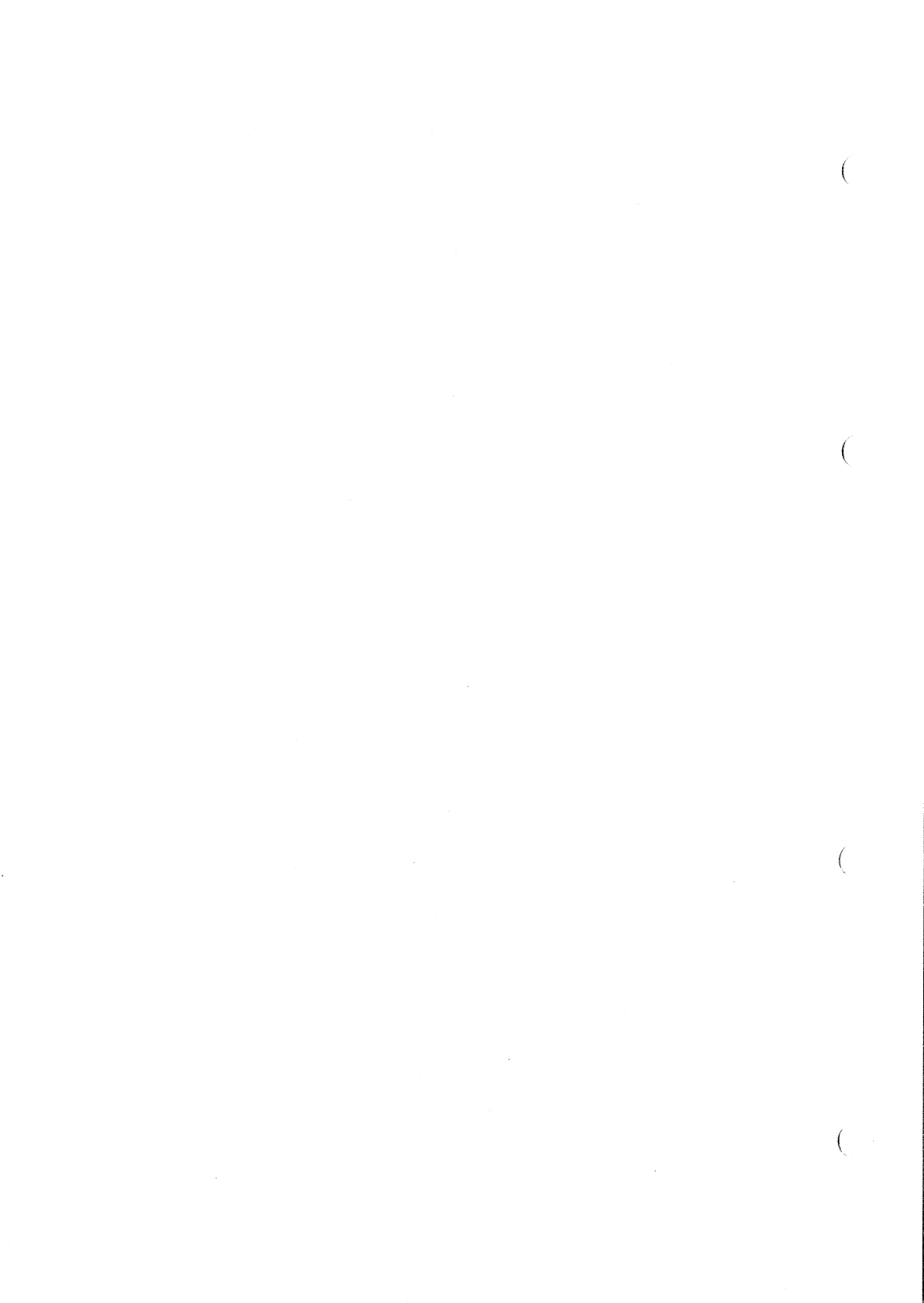


Figure 12: The Extensions Menu

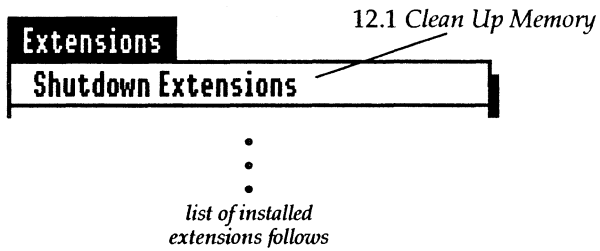


Figure 12: The Extensions Menu

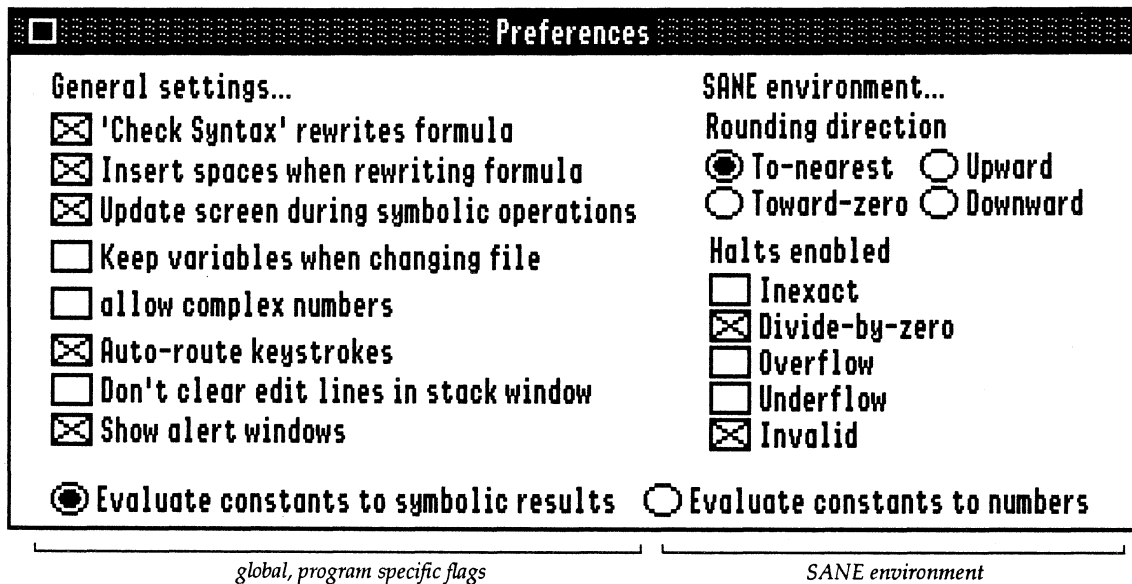
(

(

(

(

Figure 13: The Preferences Window



(

(

(

(

Table 14: Stack Diagrams

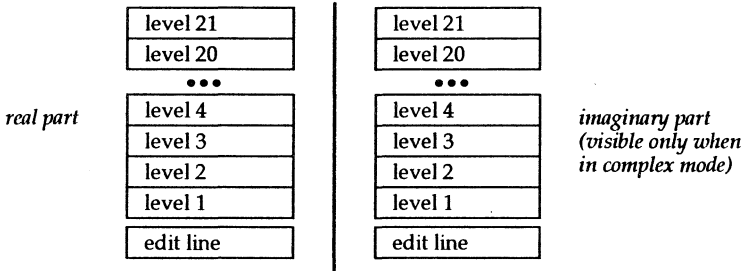
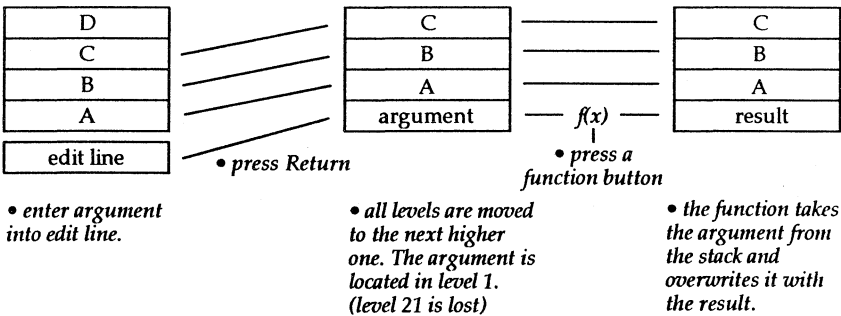


Figure 14: The Stack (compare with fig. 2b)

14.1 One-Argument Functions



14.2 Two-Argument Functions

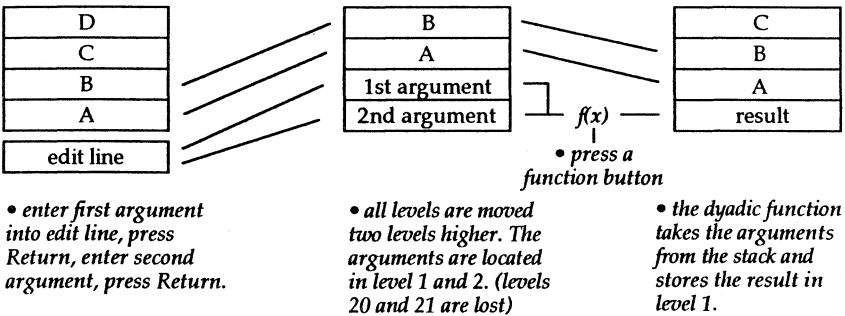
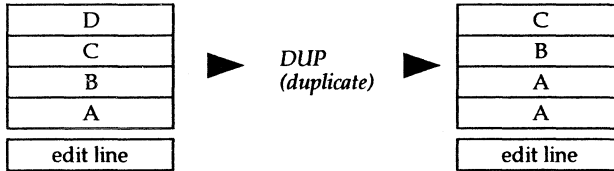
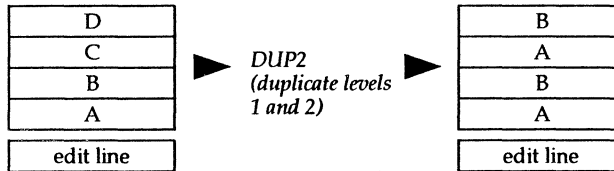


Table 14 (cont.): **Stack Diagrams**

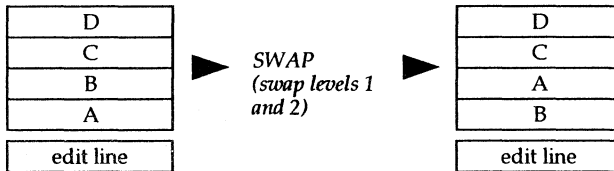
14.3 *DUP*



14.4 *DUP2*



14.5 *Swap*



14.6 *Over*

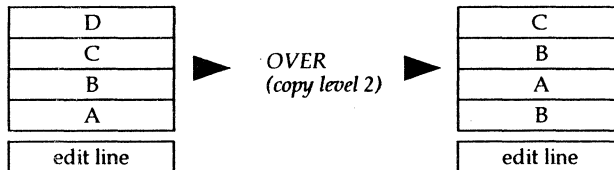
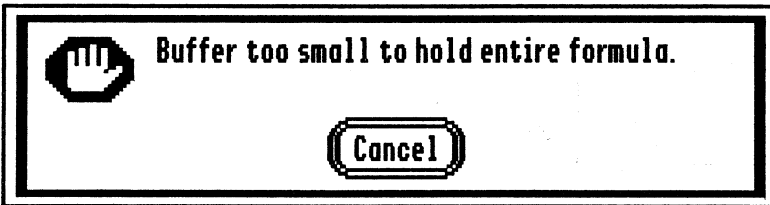
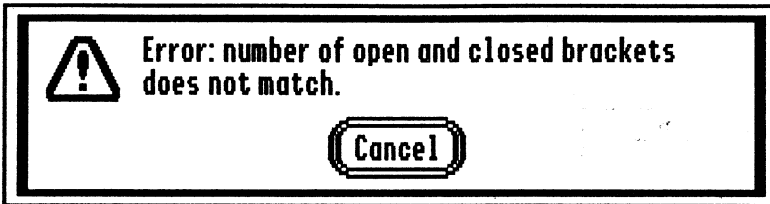
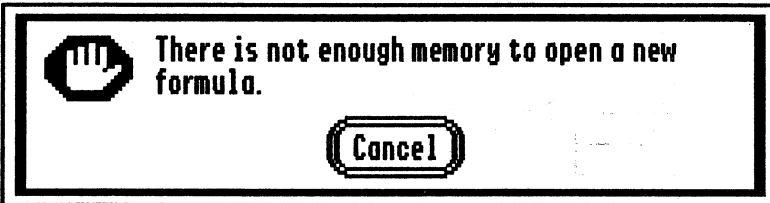


Figure 15: Error Messages

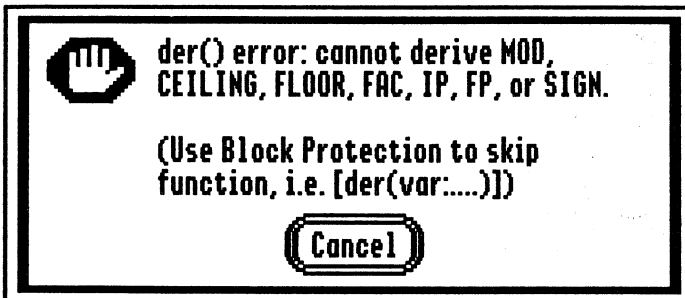
15.1 Parser Error



15.2 Buffer Error



15.3 Insufficient Memory



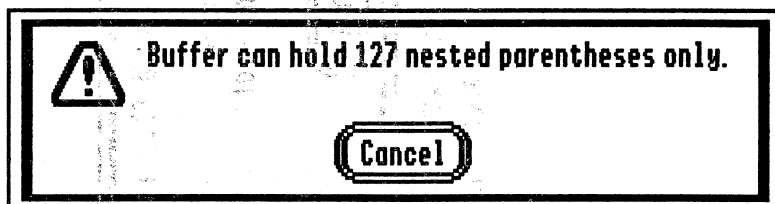
15.4 Derive (All) Error

Figure 15 (cont): **Error Messages**



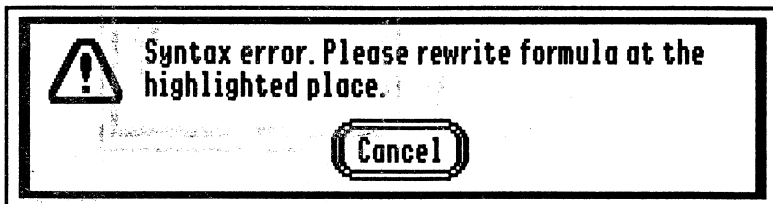
15.5 *Insufficient Memory*

15.6 *EXPAND Error*




15.7 *Parser Error*

15.8 *Parser Error*




15.9 3D Graph Error



The range parameters are not valid. Please make sure that...

- you have entered valid numbers
- you did not specify empty intervals

Cancel

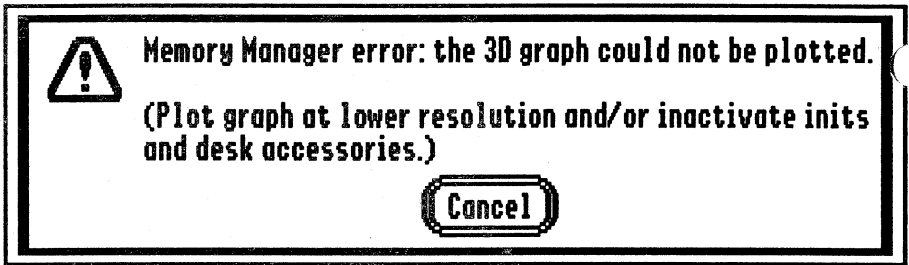


The formula contains one or more `der()`-functions and thus cannot be evaluated.
Press **CANCEL** to abort, or **DERIVE** to differentiate the expressions and execute the command again.

Derive Cancel

15.10 Evaluate Error

Figure 15 (cont.): **Error Messages**




15.11 *3D Graph Memory Error*

1961 (100) ENGINE

1961 (100) ENGINE

Right group of lower parts

1961 (100) ENGINE



1961 (100) ENGINE

1961 (100) ENGINE

