mathX

Kai Nickel

COLLABORATORS

	TITLE : mathX	

ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Kai Nickel	July 25, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

mathX

1.1 mathX documentation - table of contents

```
                    m a t h X

            Math program for Amiga Computer

               Version 1.00 (1996-11-10)
                (c) 1995-96 Kai Nickel


Introduction    Description
                Copyright
                Installation
                Registration
                Author

Expression      Expressions
                Expression elements

Usage           Main window
                Settings
                Expression display

Graphs          2D graph
                2D settings
                3D graph
                3D settings
                SIRDS graph
                SIRDS settings
                SIRDS technique

Analysis        Derivative
                Integral
                Discussion
                Tangente
                Taylor
                Simplification

Lin. Algebra    Char. polynom
                Determinant & Inverse
```

1.2 Description

Short description

mathX is a math program for Amiga. It replaces my old program Graph2D.

Feature overview

And that's what mathX can:

* 2D function plots with an optional 2nd parameter
* 3D function plots, SIRDS-"Magic3D"-plots
* Formal derivatives and simplification
* Calculation of integrals
* Creation of tangents
* Power series expansion using Taylor formula
* Discuss functions: search for zeros, extrema, turning points, symmetry, value tables
* Determinant and inverse of a matrix
* Solve linear systems
* Characteristical polynom of a matrix

mathX calculates symbolic using rational numbers if possible. It has a very comfortable graphical user interface featuring drag'n'drop, bubble- and online help. The program has got an ARexx port, installation script and is localized to English and German.

Shareware

mathX is shareware. If you want to use the program for more than a 30 days evaluation period, you have to register for US$15/US$20.

System requirements

To run mathX you need at least AmigaOS 2.0, MUI 3.3 and 2MB RAM.

1.3 Copyright

Copyright

Shareware The program is shareware, so it is freely distributable meaning that non-commercial copies are allowed as long as the programs archive remains complete and unchanged. "Meeting Pearls CD" and "Aminet CD" are explicit allowed to include mathX.
If you want to use the program for more than a 30 days evaluation period, you have to register for US$15/US$20.

MUI This application uses MUI by Stefan Stunz.

Icons MagicWB and some of the icons in the mathX package are copyright by Martin Huttenloher. Some SmallMWB-Icons have been created by Almut Silja Hildebrand for mathX.

SIRDS The SIRDS algorithm used in mathX was developed with the help of Kilian Singer.

1.4 Installation

Installation

To install mathX just start the script "mathX-Install" from the Workbench.

By hand If you do not have "Installer" the script will not work and you have to install mathX by hand:

* Copy the whole "mathX" drawer somewhere to your harddisc
* Move all documents from "mathX/Docs/Language" to "mathX/Docs". ("Language" stands for the language you like.)
* Delete drawers with languages you do not need from "mathX/Docs" and "mathX/Catalogs".
* If you want: replace the standard icons by the ones from "mathX/Icons"

Deinstallation If you want to get rid of mathX just delete its drawer. That's all.

1.5 Registration

Registration

mathX is shareware. If you want to use the program for more than a 30 days evaluation period, you have to register.

By registering you get a personal keyfile which gives you the right to use mathX and future updates (if there are any) permanently. This keyfile also disables the nerv requesters in mathX.

Please consider honestly to register. I spent lots of work

and time in programming mathX. And the registration fee is really low, so I believe that everybody can afford it.

Registration window

To make registration easy, mathX features a registration window, that creates a registration form for you. You can print this form and send it to me via postal mail - or save it as ASCII, to submit it by eMail. If you have neither a printer nor eMail, you can of course also write down that form by hand and send it to me. The resulting form is visible on the right side of the window and contains my address to send the form to.
Don't hesitate to ask me if you have any questions.

The registration window wants to get your name and address. You are asked to enter the following things in order to get your keyfile:

Name - Your first and your family name
Street - Your street and number
City - The city you live in, including the ZIP-code
Country - Your country
eMail - Your eMail address, if you have one
Comment - Optional, if you want to tell me something

Then you can choose how to get your keyfile and how to pay:

Delivery

Choose, if you want me to send the keyfile via postal mail on a disc with printed label and the newest mathX version, or just by eMail. eMail is a bit cheaper.

Payment

If you live outside Germany, you can only pay in cash included to your letter. This is because all other kinds of payment from foreign countries are much to expensive for such a small ammount of money.
If you live inside Germany, you can also pay with Euro-Cheque or bank draft ('Ueberweisung').

Currency

I accept payment in Deutsche Mark (DEM) or in US Dollar (US$). Please do not send other currencies, because changing means lot of work and costs to me.

The final price depending on delivery and currency is indicated below.

Thank you for supporting the shareware idea!

Graph2D users

Registered Graph2D users are automatically registered to mathX, too. Please consider mathX as a Graph2D update! Just move your "Graph2D.key" from the Graph2D drawer to the mathX drawer and rename it to "mathX.key".

1.6 Author

Author

mathX was written by Kai Nickel.

Mail: Kai Nickel
Herzogstrasse 29
67435 Neustadt
Germany

eMail: un7x@rz.uni-karlsruhe.de
kai@rpsbbs.rlp.org
WWW: http://www.uni-karlsruhe.de/~un7x

Updates? New versions of mathX will be put to Aminet/misc/math and to my WWW-homepage, too. You may also call the german mathX-Support-BBS called RPSBBS:

Modem: 49-6323-93066 (HST/V32b V42b)
ISDN : 49-6323-93065 (X75)
Login: graph2d
Path : /Lokal/Support/Amiga/Graph2D
(my username there is "Kai")

1.7 Expressions

Expression

MathX has been made to evaluate, display, discuss, ... mathematical expressions. In mathX the use of the word "expression" is equal to the words "term" or sometimes "function".

Mathematical expressions are represented by a string that follows a certain syntax. There is nothing extraordinary in this syntax concerning mathX, so it should be no problem to create a valid expression just like you are used to.

To be more systematic one could say that an expression always consists of some operands connected with operators.
Look at the chapter Expression elements if you are interested in what things exactly you can use in your expressions.

Examples To give you a short overview about how an expression looks in mathX, here are some examples of correct ones (Please do not enter the spaces!):

```
2*x^3 - 42.5*x^2 + x - 1/2
sqrt(a^2 + b^2)
-sin(pi/2) + cos(x)
sum(arg1, arg2) := arg1 + arg2
double(x) := sum(x, x)
g := 9.81
s(x) := (1/2)*m*g^2
```

```
            [[1, lambda], [x^2, 2]]
            f(x,y):=sqrt(1/x^2+y^2)

Syntax      Expression syntax in Backus-Naur-Form

            Expression = [ Identifier [ Arglist ] ":=" ] Prio0 .
            Prio0      = Prio1 [ ( "=" | "!=" ) Prio1 ] .
            Prio1      = [ "-" ] Prio2 { ( "+" | "-" ) Prio2 } .
            Prio2      = Prio3 { ( "/" | "*" ) Prio3 } .
            Prio3      = [ "~" ] Prio4 { "^" Prio4 } .
            Prio4      = Number | "(" Expression ")" |
                         Identifier [ "(" ExpList ")" ] | Matrix .
            ExpList    = Expression { "," Expression } .
            Matrix     = "[" ExpList "]" .
            Identifier = Letter { Letter | Figure } .
            Number     = Figure { Figure } [ "." Figure { Figure } ] .
            Figure     = "0" | .. | "9" .
            Letter     = "A" | .. | "Z" | "a" | .. | "z" | "'".


            Explanation: [ x ]  x appears 0 or 1 time
                         { x }  x appears 0, 1, 2, ... times
                         "x"    the string x appears explicit
                         x | y  x or y appears
```

1.8 Expression elements

```
                              Expression elements

Number      At the moment only plain, signed numbers and rational numbers
            written as fractions are allowed. No exponential notation yet.

            Examples: 1234.5678   1/2    -42   0    -4/5

Variable    Every string that does not fit into another category is
            considered to be a variable.

            Examples: x   myfirstvar   var1

Constant    pi                   the well known 3.14159265...
            e                    the also well known 2.17...

            You can define and use your own constants with the ":=" operator.

Matrix      A (m,n)-matrix

            /                       \
            |  A11  A12  ...  A1n   |
            |                       |
            |  A21  A22  ...  A2n   |
            |   .                .  |
            |   .                .  |
            |   .                .  |
            |  Am1  Am2  ...  Amn   |
            \                       /
```

is given by:

```
[[A11, A12, ..., A1n], [A21, A22, ..., A2n], ...,
[Am1, Am2, ..., Amn]]
```

Function

sqrt(x)		sqare root function
exp(x)		exponential function e^x
log(x)		logarithmus, base e
abs(x)		absolute value (always positive)
sin(x)	sinh(x)	trigonometrical functions
cos(x)	cosh(x)	
tan(x)	atan(x)	
asin(x)	acos(x)	
trace(A)		Trace of a aquare matrix
charpol(A)		Characteristic polynom of a matrix
solve(A, x)		Solve the Linear system
rang(A)		Rang of a matrix
det(A)		Determinant of a matrix
inv(A)		Inverse matrix of A
diff(f(x),x)		Derivative of f(x) to x
integral(f(x),a,b)		Numerical integral of f(x) in [a,b]
tangente(f(x),a)		Tangente or
normale(f(x),a)		Normale of f(x) at the point a
taylor(f(x),x0,d)		Taylor polynom with degree d approximating f(x) at x0

You can define and use your own functions with the ":="
operator.

Operator

+ - * / ^	Standard operators. +, - can also handle matrices.
&*	Matrix multiplication
= !=	Comparative operators, return 1 if true and 0 if false
:=	Assignment operator (definition) The expression on the right side will be connected with the identifier on the left side. Now you can refer to the expression by calling its name anywhere in mathX. Arguments can be given to define functions. A definition is valid as soon as you interpret the expression containing the ":=" operator. It stays valid until you quit mathX or until you assign the same identifier to another expression.

```
Examples: g      := 9.81
          f(x)   := sin((x/2)^2) * 2
```

```
poly     := -x^3 + (1/6)*x^2 - 5
```

1.9 Main window

Main window

With the buttons and menus of the main window you can open any other window of mathX and start all actions. If you close te window you will close mathX.

The main window consists of the following things:

Image buttons

2D graph	Open a new 2D graph window
3D graph	Open a new 3D graph window
SIRDS graph	Open a new SIRDS graph window
Expression display	Open a new Expression display window
Settings	Open the program settings window
ARexx	Execute an ARexx script

Text buttons

These buttons are there to easily access the mathematical options of mathX. Because there exists an according menu item for each button, you can configure mathX to hide the buttons in the settings window.
For a description of the buttons please look at the explanation of the according menu items below.

Expressions

This list contains all expressions you want it to contain. It is a kind of expression clipboard for you. From this list you can drag'n'drop expression from or to any other expression list or expression gadget in mathX. The expressions in the list can be saved and loaded. For details and for description of the list buttons please look at the explanation of the according menu items below.
The expression gadget above the list allows you to edit the selected expression of the list. The text you enter here gets interpreted as soon as you press return. If the interpreter detects a syntax error in your expression, a requester pops up that informs you about that error. Only correct expressions are added to the list and can be dragged'n'dropped.

Menu

Project

New 2D graph...	Open a new 2D-graph window
New 3D graph...	Open a new 3D-graph window
New SIRDS graph...	Open a new SIRDS-graph window
About...	Open the program information window
About MUI...	Open a MUI information window
Help...	Open this online documentation. You can press the HELP key on you keyboard to do just the same thing.
Iconify	Close all mathX windows and put an icon on the workbench that allows you to awake mathX again.
Quit...	End mathX

Expression	
New	Create new expression and add it to the list.
Cut	Cut out the selected expression and put it into an internal buffer.
Paste	Re-insert the expression from the buffer.
Clear list	Delete all expression from the list.
Load...	Load new expressions to the list.
Append...	Load expressions and add them to the list.
Save As...	Save the expressions of the expression list.
Display...	Open a new expression display window.
Analysis	
Derivative...	Open a new derivative window
Integral...	Open a new integral window
Simplification...	Open a new simplification window
Tangente...	Open a new tangente window
Taylor...	Open a new taylor window
Discussion...	Open a new discussion window
LinAlg	
Determinant...	Open a new determinante window
Lin. system...	Open a new linear system window
Char. Polynom...	Open a new char. polynom window
Settings	
mathX...	Open the program settings window
MUI...	Open the MUI preferences for mathX
ARexx	
Execute script...	Execute a selectable ARexx script
Term to MathScript	Execute this ARexx script.

1.10 Settings

Settings

You can set the following presets for mathX:

Textbuttons	Show or hide the text buttons in the main window
IFF save size	Pixel width and height of every graph when you save it as IFF image
Term display fonts	Fonts used in the Expression display. You need to specify a small font used for exponents

and a big one for all the rest.

1.11 Expression Display

Expression Display

An expression will be displayed here in a more readable, a more 'natural' way. What means that e.g. fractions are shown with horizontal lines, exponents are at high position, the square root function uses the symbol, ...

The font used for the expression display can be choosen in the program settings.

drag'n'drop If you drop an expression from somewhere else into the window, the expression will be displayed.

Menu Menu items:

Save as IFF... Save the expression as IFF image

1.12 2D Graph

2D Graph

In a 2D graph window you can look at the graphs of functions. They are presented in a standard two-dimensional coordinate system. Expressions you want to display here must contain 1 real variable and return a real value.
An optional second argument can be handeled, too.
You can open as many 2D graph windows as you want if you have enough free memory.

You can start actions with the following things:

Image buttons

Graph settings	Open the 2D graph settings window.
Zoom in	Zoom into the graph
Zoom out	Zoom out of the graph
Zoom mouse	Zoom into a mouse-selectable region of the graph: Click on this button, then press left mouse button somewhere over the graph, hold down and release it anywhere else.
Copy	Create a new graph window that looks the same
Save IFF	Save graph as IFF image. The pixel size of the image calculated for saving can be configured in the program settings.
Print	Print graph

If you don't like these image buttons, you can switch them

off in the 2D graph settings. This will save some
space on your screen and you can still start the actions
via menu.

Menu 2D graph

Redraw	Redraw the graph (you won't need that.)
Zoom in	Like the according image button
Zoom out	Like the according image button
Zoom mouse	Like the according image button
Copy...	Like the according image button
Scroll	Scroll around in the visible area
Save IFF...	Like the according image button
Print...	Like the according image button
Close	Close the window.

Settings

Range...	Open the 2D graph settings window
Scales...	Open the 2D graph settings window
Design...	Open the 2D graph settings window
Functions...	Open the 2D graph settings window
Misc...	Open the 2D graph settings window

Keyboard	Cursor keys	Scroll around
	+	Zoom in
	-	Zoom out

Mouse Press left mouse button and move pointer over the graph.
You will see the coordinates of every pixel on your way.

Drag'n'Drop Drop a math. expression from somwhere else onto the graph
and the function will be added to the graph and drawn.

1.13 2D Settings

2D Settings

Here you can configure the look and contents of a 2D graph.
You can 'Load...' and 'Save...' the graph settings.
Select 'Save as default' from the menu and you make the actual
settings to be used for every new graph you will open in future.

The window consists of five pages:

Range For each the x- and y-axis you can set the following things:

from / to - Visible interval (depends on unit, too)

Unit - Basical unit of the axis. All other scaling parameter
of the axis will (internally) be multiplied with the unit.
So the unit is a kind of scaling factor. Usually you will set
unit to "1", but if you for example want to generate a graph

for some trigonometric functions, you may set unit to "pi". Then a visible interval of e.g. [-2; 4] would mean [-2*pi; 4*pi]. The unit must of course be > 0.

Type - Switch between linear ("normal") or logarithmical scaling.

Only for the y-axis you can set the following attribute:

Autorange - The y-axis "from" and "to" settings will be ignored. mathX will calculate the y-range automatically, so that all functions are completely visible in the graph.

Scales For each the x- and y-axis you can set the following things:

Marks - Divide the axis by little lines. You can define after how many units a mark should be drawn.

Submarks - Number of smaller "submarks" between every two regular marks.

Numbers - Draw a number at every nth mark. mathX automatically leaves out numbers that would overlap one another if they have to be placed too narrow.

Grid - Draw a grid over the graph with a selectable distance again measured in units.

Grid design - Design of the grid lines

Design System design - Position of the axes: "Cross" means the normal intersecting axis positions. "Box" shows the axes always on the top/bottom and left/right of the graph.

For each the x- and y-axis you can set the following things:

Show axis - Well, you don't have to draw the axis.

Title - A small text that will be placed next to the axis.

Axis design - Design of the axis lines

Arrow - Draws a small arrow at the end of the axis (some people like that very much)

Functions Here you can edit all functions drawn in the graph:

Design - Linestyle of the selected function

Accuracy - The distance in pixels between each two calculated function values. The lower the value the slower but more precise the display will be.

Connect - Connect all calculated function values with a line.

2nd argument - If your function has got two variables you can select a range for the second variable here. This makes mathX draw the function several times with different values for that 2nd argument.

Misc

Title - Window title text. Who cares...

Buttons - Show or hide image buttons

Background - Select the color of the graph background

Axis font - Font of the numbers and axis title

Mouse font - Font of the mouse-coordinate-display. A small font looks nice here.

1.14 3D Graph

3D Graph

In a 3D graph window you can look at the drawn surface of a function with 2 different variables.
You can open as many 3D graph windows as you want if you have enough free memory.

You can start actions with the following things:

Image buttons

Graph settings	Open the 3D graph settings window
Copy	Create a new graph window that looks the same
Save IFF	Save graph as IFF image. The pixel size of the image calculated for saving can be configured in the program settings.
Print	Print graph

If you don't like these image buttons, you can switch them off in the 3D graph settings. This will save some space on your screen and you can still start the actions via menu.

Menu

3D graph

Redraw	Redraw the graph (you won't need that...)
Copy...	Like the according image button
Save IFF...	Like the according image button
Print...	Like the according image button
Close	Close the window.

Edit

Function...	Open the 3D graph settings window
Parameter...	Open the 3D graph settings window
Color...	Open the 3D graph settings window
Window...	Open the 3D graph settings window

Drag'n'Drop Drop a math. expression from somwhere else onto the graph and the function will be shown in the graph.

Sliders With the three sliders you can quickly adjust the projection parameters of the function display. Take a look at the 3D graph settings window to see what the parameters are.

1.15 3D Settings

3D Settings

Here you can configure the look and contents of a 3D graph. You can 'Load...' and 'Save...' the graph settings. Select 'Save as default' from the menu and you make the actual settings to be used for every new graph you will open in future.

The window consists of four pages:

Function Function - The expression that will be displayed

For the x- and y-axis you can set the visible interval.

Parameter Type - Type of the projection. Rcoomended standard is "Perspective", the other two possibilities are not very spectacular - try it to see what they do.

Precision - Number of function values that will be calculated for the image. If you want to draw a big image for a big window, you should use high precision. But this will lead to longer calculation time.

Phi and Theta - Angle of the observer position

Magnification - Size of the graph, a kind of "zoom"

Color Specify colors for the background, the surface fill and the lines of the graph.

Window Window title - Window title text

Buttons - Show or hide image buttons

1.16 SIRDS Graph

SIRDS Graph

In a SIRDS graph window you can look at the graphs of a function in a special way. The expressions you want

to display here must contain max. 2 different real variables and return a real value.
You can open as many SIRDS graph windows as you want if you have enough free memory.

You can start actions with the following things:

Image buttons

Graph settings	Open the SIRDS graph settings window.
Copy	Create a new graph window that looks the same
Save IFF	Save graph as IFF image. The pixel size of the image calculated for saving can be configured in the program settings.
Print	Print graph

If you don't like these image buttons, you can switch them off in the SIRDS graph settings. This will save some space on your screen and you can still start the actions via menu.

Menu

SIRDS graph

Redraw	Redraw the graph (you won't need that...)
Copy...	See according image button
Save IFF...	See according image button
Print...	See according image button
Close	Close the window.

Edit

Range...	Open the SIRDS graph settings window
Function...	Open the SIRDS graph settings window
Window...	Open the SIRDS graph settings window

Drag'n'Drop

Drop a math. expression from somwhere else onto the graph and the function will be shown in the graph.

1.17 SIRDS Settings

SIRDS Setting

Here you can configure the look and contents of a SIRDS graph.

You can 'Load...' and 'Save...' the graph settings.
Select 'Save as default' from the menu and you make the actual settings to be used for every new graph you will open in future.

The window consists of three pages:

Range

For each the x-, y- and z-axis you can set the visible interval. The x-axis runs horizontally from left to right, the y-axis vertically from to bottom to the top and the z-axis magically comes out of the monitor into your direction.

Function

Function - The expression that will be displayed

Precision - Number of function values that will be calculated for the image. If you want to draw a big image for a big window, you should use medium or high precision. But this will cause longer calculation time.

Misc Window title - Window title text

Buttons - Show or hide image buttons

1.18 SIRDS technique

SIRDS technique

mathX is able to use for the three-dimensional presentation of functions a method known as SIRDS (= Single Image Random Dot Stereogram). You do not need special glasses or something like that, you may even print out the pictures without loosing the effect.

Look! The only thing you have to do to achieve the 3D-effect is to look at the pictures in a special way: Instead of looking at the picture itself you have to look "behind" the picture. Of course you will not see the random dots sharp this way - but this is nescessary for the three-dimensionality. It may take sometimes until the effect is going to come, and for strange reasons some people never get the "kick". For exact instructions how to achieve the effect please refer to the numerous publications concerning SIRDS pictures.

1.19 Derivative

Derivative

Here you can derive an expression.

Function You can derive functions going from R -> R that are continuous and (of course) derivable.

Order Order of derivative. 1 means f', 2 means f'', ...

Derivative Derivative. Is already simplified.

Derive Create derivative

Close Close window

1.20 Integral

Integral

In this window you can calculate the numerical integral of

an expression with Romberg's method.

Expression — Function to be integrated. It must be of the type R -> R, continuous and derivable in the selected interval.

from / to — Specify the desired range of the integration

Precision — Determine how exact (and also slow) the calculation will be. But it cannot be guaranteed that the error of the result will always be smaller than the given precision!

Absolute area — If active, the integration is done on "abs(f)", where "f" stands for your expression. This is useful if you are interested only in the geometric area between function and x axis.

Result — Result of integration.

Integrate — Start calculation

Close — Close window

1.21 Discussion

Discussion

Here you can discuss an expression that contains max. one variable and returns a real value. The result of the discussion gets shown on the right side in a text-view field from where it can be saved or printed

from / to — Discussion interval

Zeros — Find all points with f(x) = 0. In addition to the x-values you will also get the following information about the type of the zero:

-+	f'(x) > 0
+-	f'(x) < 0
++	f'(x) = 0
--	f'(x) = 0

Extrema — Find all points with f'(x) = 0

min	f''(x) > 0
max	f''(x) < 0

Turning points — Find all points with f''(x) = 0

left->right	f'''(x) < 0
right->left	f'''(x) > 0

Symmetry — The function is tested whether it is:

symmetric to origin f(x) = -f(-x)

symmetric to y-axis f(x) = f(-x)
no symmetry

Value table — Create a value table with selectable step width

Start / Stop — Start / interrupt the discussion

1.22 Tangente

Tangente

Create a tangente or normale of an expression with max. 1 variable at a selectable point.

drag'n'drop — Drag'n'Drop to and from the Expression and Result field is possible (like everywhere in mathX).

Type — Create tangente or normale. Look at the picture in the window to figure out what I means when I say tangente and normale...

1.23 Taylor

Taylor

An expression with max. 1 variable can be approximated around a selectable point at with the taylor polynom.

drag'n'drop — Drag'n'Drop to and from the Expression and Result field is possible (like everywhere in mathX).

1.24 Simplification

Simplification

mathX can simplify expressions. These simplifications are neither perfect nor very elegant, but they can be useful to make big expressions a bit more handy.

The main strategy of simplification is to remove operations with neutral elements (e.g. x*0, a+0, 1*sin(r), x^0), and to do calculations with constants (e.g. 2+3). Sometimes it is possible to extract common factors out of sums.

Modes — There are two simplification modes:

symbolic — Keep symbols like 'pi' or 'e'
numeric — Replace all known symbols by their values

Errors — A simplification can find errors in the expression:

undefined	Expression mathematically not defined (like x/0)
type error	An operator is connected to an argument that cannot be handled (e.g. 2&*3, because &* expects matrices)

1.25 Characteristical Polynom

Characteristical Polynom

Calculate the characteristical polynom of a square matrix A. charpol(A) = det(A - lambda*E).

Eigenvalues The zeros of this polynom are the eigenvalues of A. You can start a zero search on the polynom to find the eigenvalues.

Window elements:

Matrix The matrix A. It must be square.

Char. polynom The characteristical polynom. Its variable is 'lambda'.

Calculate Calculate char. polynom

Close Close the window

1.26 Determinant and Inverse

Determinant and Inverse

Calculate the determinant (det A) and inverse (A^{-1}) of a square matrix A.

Window elements:

Matrix Matrix A, must be square

Determinant Determinant det(A), it is already simplified.

Inverse Inverse matrix A^{-1}

Calculate Start calculation. The calculation can take quite a long time depending on the size of the matrix. You can see the progress of the calculation by the progress indicator bar.

Close Close window

Example The determinant of [[1,x],[2,3]] is 3-2*x and the inverse matrix is [[1+x*2/(3-2*x),-x/(3-2*x)],[-2/(3-2*x), 1/(3-2*X)]]

1.27 Linear System

```
                        Linear System

          Solve a linear system given by a square matrix A an an
          appropriate vector c.

Example   Imagine you have a system of equations given like

          -2 * x1    +  (1/2) * x2    -  2 * x3    =   7
           5 * x1    -    9   * x2                 =  -3
          11 * x1    -    2   * x2    +      x4    =   2

          This system can be written as A * x = c, where

               / -2  1/2  -2 \
          A =  |  5   -9   0 | = [[-2,1/2,-2],[5,-9,0],[11,-2,1]]
               \ 11   -2   1 /

          x =  [x1,x2,x3]

          c =  [7,-3,2]

          Let mathX compute and you will see the result vector x.
```

1.28 ARexx

```
                                ARexx

                The main window contains an ARexx menu that
                allows you to execute ARexx scripts. Some of the scripts
                included in the mathX package are inplemented as own menu
                items, all other scripts can be selected with a file
                requester.

Scripts         Here are the included scripts:

                SendToMathScript - Send the selected expression to the
                program MathScript. MathScript should be running if you
                want to use that script.

                ExportEPS - Export the text of the selected expression as
                EPS-vector-graphic. Also this script only works if Math-
                Script is running.

Commands        Just like other MUI applications, the name of mathX's
                ARexx port is "MATHX.1" and it understands all standard
                MUI commands like 'QUIT'. Here are mathX's own commands:

                GETFUNCTION PLAIN - Returns the text of the expression
                that is actually selected in the control panel. You will
                receive an empty string if there is no selected function.

                GETFUNCTION MS30 - Returns the text of the expression
```

that is actually selected in the control panel. The text gets converted into the formula description code used by MathScript 3.0.

More commands? If you think that mathX needs some new ARexx commands, please contact me! It is no problem to implement a command when there is a need for it - I just didn't want to create lots of commands that probably nobody uses...

1.29 Known Bugs

Known Bugs

* The pattern of a function is not displayed correctly using CyberGraphics. This is probably a bug/restriction of CyberGraphics and not of mathX. Nevertheless you may see a correct pattern if you use low accuracy.

* SIRDS graph sometimes shows strange vertical lines. Change window size to avoid this problem.

1.30 Localization

Localization

As you can see English is not my native language, so prepare to find lots of mistakes in this english documentation and the english catalog...
Please contact me if you want to do a translation into "real" english or into another language that has not been made yet.

The archive contains a standard .cd file that should allow you to create a translation easily. Again: Please contact me so that I can coordinate the work. Thank you!

1.31 MathScript

MathScript

MathScript is a formula-editor by Simon Ihmig. It is under copyright of its Author and it is shareware. MathScript is not part of mathX! So if you want to use MathScript you have to look for it yourself (e.g. in "aminet/misc/math") and register at Simon Ihmig.

Besides many other features MathScripts is able to export mathematical expressions as high quality EPS vector graphic which for example can be imported by most of the popular word processors. The results are excellent and better than the expression display abilities of mathX.

ARexx To exchange expressions between mathX and MathScript, mathX contains some ARexx scripts to communicate with MathScript 3.0.

1.32 Magic User Interface (MUI)

Magic user interface

This application uses

MUI - MagicUserInterface

MUI is a system to generate and maintain graphical user interfaces. With the aid of a preferences program, the user of an application has the ability to customize the outfit according to his personal taste.

MUI is distributed as shareware. To obtain a complete package containing lots of examples and more information about registration please look for a file called "muiXXusr.lha" (XX means the latest version number) on your local bulletin boards or on public domain disks.

If you want to register directly, feel free to send

DM 30.- or US$ 20.-

to

Stefan Stuntz
Eduard-Spranger-Straße 7
80935 München
GERMANY

MUI in mathX mathX needs an installed MUI Version 3.3 or higher. You are allowed to use MUI without registering for it - but when you register you can take advantage of some extended functions in the MUI preferences. It is very recommended to read the MUI documentation carefully - especially of the MUI preferences. Despite that I would like to show you in the following list some advantages of MUI-programs that could be useful using mathX:

Screens If you want mathX to work on an own screen and not on the Workbench then simply configure mathX with the MUI preferences to use any screen you like.

Fonts Windows of MUI applications are resizeable and completely fontsensitive, meaning, that they look fine with every font.

Iconification MUI applications may be iconified at every time with an extra-gadget in the windows title bar.

Keyboard MUI applications can optionally completely be handled with the keyboard. Via Tab-cycling and shortcuts every gadget may be (de-)activated without having to use the mouse. The gadget actually receiving keyboard input is always marked with a border or something like this. Windows can be closed normally by pressing ESC.

Commodity MUI applications are known to the system as commodities and so they can be controlled with the commodity-exchange program.

1.33 Index

Index

```
R     Register

S     Settings window
      Simplification
      SIRDS graph
      SIRDS settings
      SIRDS technique

T     Tangente
      Taylor
```