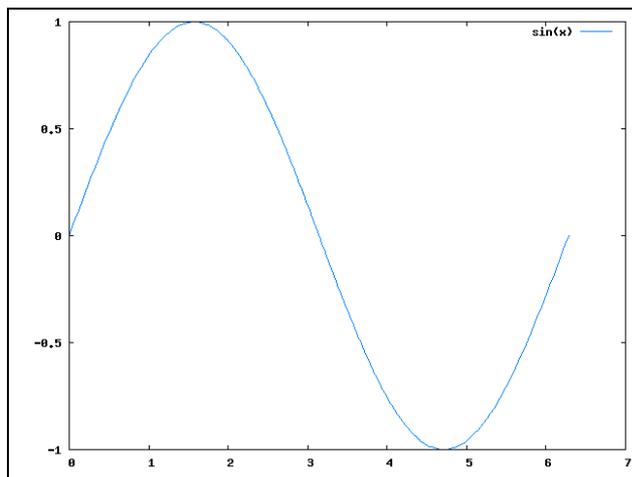




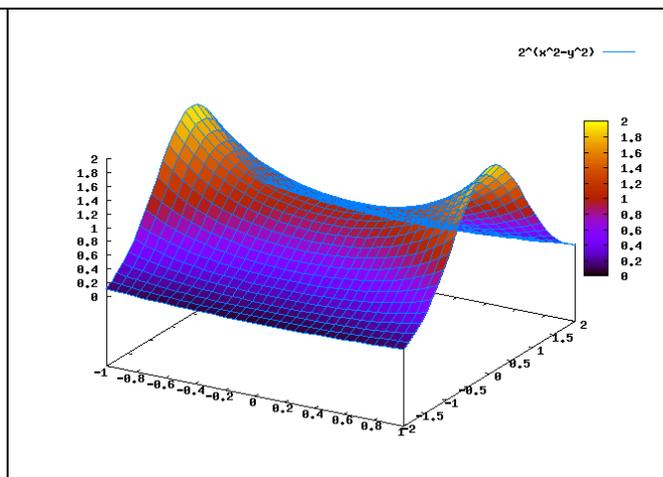
# MAXIMA

## Minimanual to study maths functions

Maxima is a computer algebra system that manipulate symbolic and numerical expressions, including differentiation, integration, systems of linear equations, and vectors, matrices, and so on. Maxima produces high precision results by using exact fractions and arbitrarily long floating point representations, and can plot functions and data in 2D and 3D.

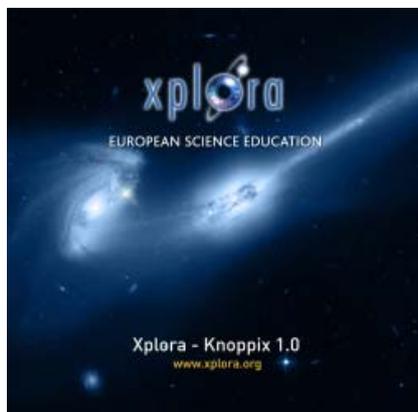


Picture n.1 :  $y = \sin(x)$



Picture n.2 :  $f(x,y) = 2^{(x^2-y^2)}$

The Maxima source code can be compiled on many systems, including Windows, Linux, and MacOS X. The source code for all systems and precompiled binaries for Windows and Linux are available at the <http://maxima.sourceforge.net/index.shtml><sup>1</sup>



You can find Maxima also into the Xplora DVD, a live bootable DVD – Xplora Knoppix, based on the Linux Debian distribution and completely contained on a self booting DVD. This Knoppix version allow you to work in Linux OS without installing it on your PC. The DVD contains software applications focused on science education. As it is open source software it can be given away freely and be copied as well.

(you can read more <http://www.xplora.org/shared/data/pdf/xploraknoppix.pdf> ) The last version Maxima 5.9.3 was released March 19, 2006. The SourceForge file manager has source and binary files, including a self-contained.

Windows installer. Maxima is constantly updating, to fix bugs and improve the code and the documentation. Most discussion and help you can find on the Maxima mailing list. <http://maxima.sourceforge.net/maximalist.html>

<sup>1</sup> SourceForge.net <http://sourceforge.net/> is the world's largest Open Source software development web site, hosting more than 100,000 projects and over 1,000,000 registered users with a centralized resource for managing projects, issues, communications, and code. SourceForge.net has the largest repository of Open Source code and applications available on the Internet, and provides a wide variety of services to projects we host, and to the Open Source community.

**Running Maxima** the prompt like is (%i1) where the "i" represent "input".

If you input a simple calculation

```
(%i1) ((2*5)^3 - 100)/3 ;
```

you will get the result as

```
(%o1) 300
```

where the "o" represent "output".

When we use a "expand",

```
(%i2) expand((x+y)^2);  
(%o2) y2 + 2 x y + x2
```

or if you use "factor"

```
(%i4) factor(4!);  
(%o4) 23 3
```

And

```
(%i1) factor(x^3 - 1);  
(%o1) (x - 1) (x + x + 1)2
```

**NOTE : end each Maxima command with a semicolon ;**

**end the session with the command (%in)**

**quit()); if you don't want a result showed, you can finish your command with \$ instead of ;.**

## Variables and parameters

A variable can be defined by

```
(%i1) x:1;
```

which will put the value "1" into the variable "x".

```
(%o2) 1
```

This variable is accessible everywhere (even in functions) unless it is redefined in a BLOCK statement. Since the variable is accessible "everywhere", it is advisable to use BLOCK in a functions or else temporary variables inside the function might affect your results.

Or you can introduce a parameter "a";

```
(%i1) a:100;
```

```
(%o1) 100
```

```
(%i2) sqrt(a);
```

```
(%o2) 10
```

Where **sqrt** ( $x$ ), is the square root of  $x$ . It is represented internally by  $x^{(1/2)}$ .

## Constants

There are several constants such as

- %e - 2.71828
- %inf - infinity
- %i - square-root of -1, complex number
- %pi - pi (3.14159265)

Their value can be obtained using

```
bfloat(%e);  
and with the precision set by changing the variable  
fpprec:150;
```

in the order below:

```
(%i1) fpprec:150;  
(%o1) 150  
(%i2) bfloat(%e);  
(%o2) 2.7182818284590452353602874713526624977572470936999595749669676277240766\  
303535475945713821785251664274274663919320030599218174135966290435729003342952\  
6b0
```

Maxima knows about complex numbers and numerical constants:

```
(%i1) cos(%pi);  
(%o1) - 1  
(%i2) exp(%i*%pi);  
(%o2) - 1
```

**Introduction to Help** - Maxima can search the info pages. Use the `describe` command to show all the commands and variables containing a string, and optionally their documentation. The question mark `?` is an abbreviation for `describe`:

```
(%in) ? function  
  
0: (maxima.info)Special Functions.  
1: Introduction to Special Functions.  
2: Definitions for Special Functions.  
3: Elliptic Functions.  
4: Introduction to Elliptic Functions and Integrals.  
5: Definitions for Elliptic Functions.  
6: Function Definition.  
7: Introduction to Function Definition.  
8: Function.  
9: Definitions for Function Definition.  
10: Function and Variable Index.  
11: functions :Definitions for Function Definition.  
12: remfunction :Definitions for Command Line.  
13: tr_bound_function_applyp :Definitions for Function Definition.  
14: tr_function_call_default :Definitions for Function Definition.  
15: tr_warn_bad_function_calls :Definitions for Function Definition.  
16: (contrib.info)Function and variable index.  
17: mat_function :Definitions for diag.  
18: (numericalio.info)Function and variable index.
```

19: (descriptive.info)Function and variable index.  
 20: (distrib.info)Function and variable index.  
 21: (simplex.info)Function and variable index.  
 22: (solve\_rec.info)Function and variable index.  
 23: (stringproc.info)Function and variable index.  
 24: (linearalgebra.info)Function and variable index.  
 25: (orthopoly.info)Function and variable index.  
 26: (simplification.info)Function and variable index.  
 Enter space-separated numbers, `all' or `none': 8

#### 40.2 Function =====

To define a function in Maxima you use the := operator. E.g.

```
f(x) := sin(x);
```

defines a function `f'.

### An example regarding function and its zeroes:

```
(%i1) f(x) := x^3-4*x^2+3*x;
(%o1) f(x) := x^3 - 4 x^2 + 3 x

(%i2) f(x)=0;
(%o2) x^3 - 4 x^2 + 3 x = 0

(%i3) solve(f(x)=0,x);
(%o3) [x = 3, x = 1, x = 0]
```

Maxima calculate differential and integral of functions:

```
(%i4) diff (f(x), x);
(%o4) 3 x^2 - 8 x + 3

(%i5) f1(x) := 3* x^2 - 8* x + 3;
(%o5) f1(x) := 3 x^2 - 8 x + 3

(%i6) integrate(f1(x),x);
(%o6) x^3 - 4 x^2 + 3 x
```

Maxima can solve linear and nonlinear sets of equations:

```
(%i1) eq_1: x+ 3*y = 0$
(%i2) eq_2: 3*x + y = 1$
(%i3) solve ([eq_1, eq_2]);
(%o3) [[y = - 1/8, x = -1/8]]

(%i4) float(%o3);
(%o4) [[y = - 0.125, x = -0.125]]
```

Maxima can display a plot of one or more functions of one variable.

```
plot2d ([expr_1, ..., expr_n], x_range, y_range)
plot2d ([expr_1, ..., expr_n], x_range)
```

To get the graphics in picture 1 at page 1

```
(%i1) plot2d(sin(x), [x, 0, 2*%pi]);
```

To get a PNG file:

```
plot2d(sin(x), [x, 0, 2*%pi],
      [gnuplot_preamble,
       "set terminal png;set out 'sin.png'"])$
```

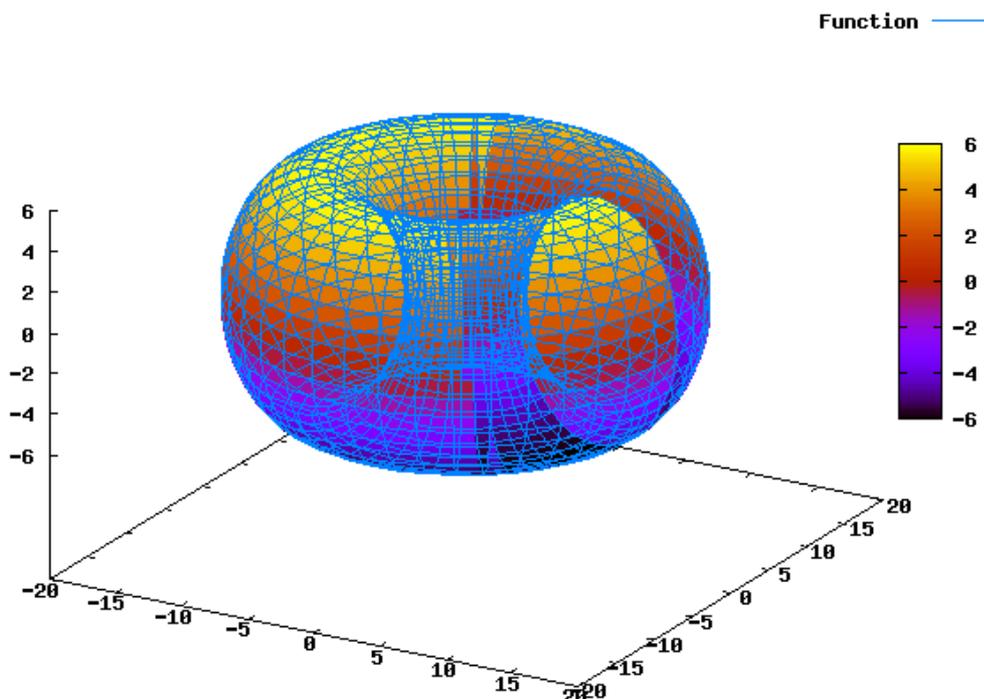
To get an EPS file:

```
plot2d(sin(x), [x, 0, 2*%pi],
      [gnuplot_preamble,
       "set terminal postscript eps;set out 'sin.eps'"])$
```

With Gimp, you can convert then the png to jpeg and other bitmat formats, but there are lots of programs to make file conversions.

To get a 3D graph: a torus

```
(%i1) expr_1: cos(y)*(10.0+6*cos(x));
(%i2) expr_2: sin(y)*(10.0+6*cos(x));
(%i3) expr_3: -6*sin(x);
(%i4) plot3d ([expr_1, expr_2, expr_3],[x, 0, 2*%pi],[y, 0,2*%pi],[grid, 40, 40]);
```



To get a PNG file that allowed you to insert it in a document like a image the command line is :

```
(%i5) plot3d ([expr_1, expr_2, expr_3], [x, 0, 2*%pi], [y, 0, 2*%pi],
  ['grid, 40, 40],[gnuplot_preamble,
  "set terminal png; set out 'torus.png'"])$
```

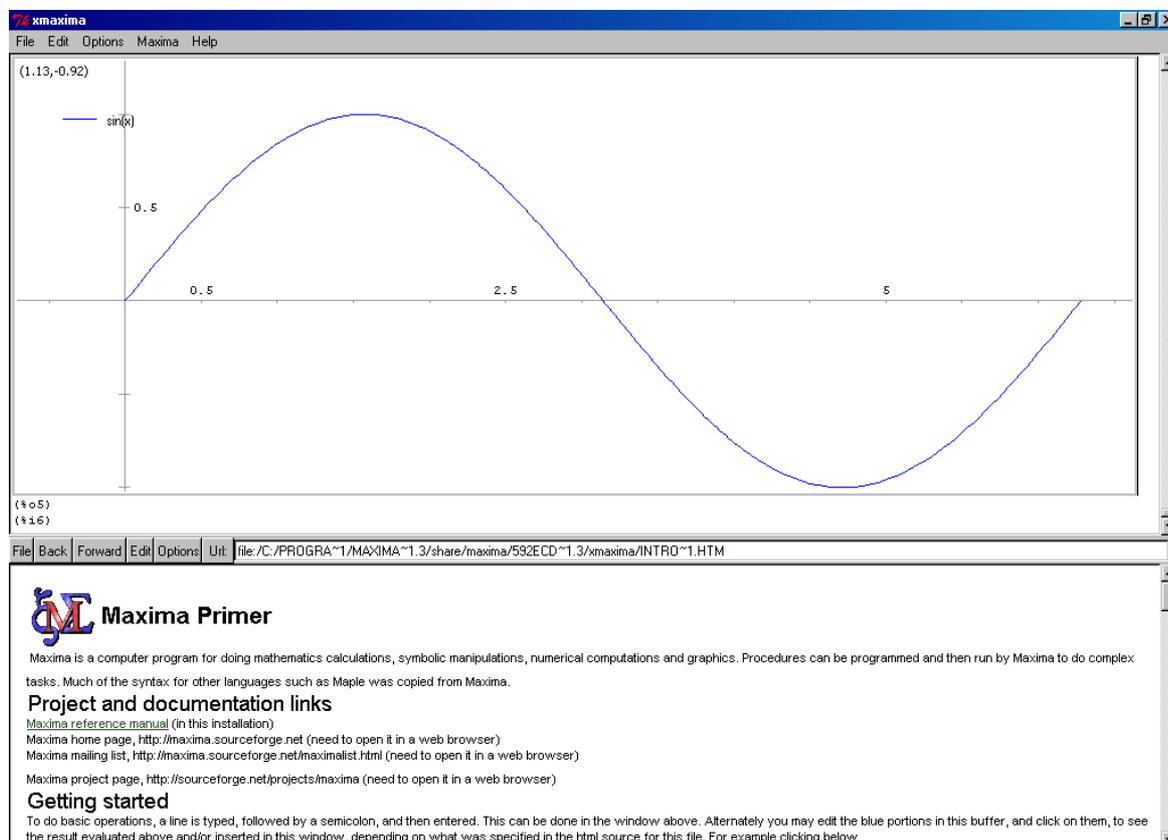
Maxima can generate several plot\_formats: OPENMATH, GEOMVIEW, GNUPLOT and IZIC. "OPENMATH" is the default plot\_format and "GNUPLOT" ( a most advanced, plotting package) is very in general.

```
(%i4) set_plot_option([plot_format,openmath]);
(%o4) [[x, - 1.75555970201398E+305, 1.75555970201398E+305],
[y, - 1.75555970201398E+305, 1.75555970201398E+305], [t, - 3, 3],
[grid, 30, 30], [view_direction, 1, 1, 1], [colour_z, false],
[transform_xy, false], [run_viewer, true], [plot_format, openmath],
[gnuplot_term, default], [gnuplot_out_file, false], [nticks, 10],
[adapt_depth, 10], [gnuplot_pm3d, true], [gnuplot_preamble, ],
[gnuplot_curve_titles, [default]], [gnuplot_curve_styles,
[with lines 3, with lines 1, with lines 2, with lines 5, with lines 4,
with lines 6, with lines 7]], [gnuplot_default_term_command, ],
[gnuplot_dumb_term_command, set term dumb 79 22], [gnuplot_ps_term_command, se\
t size 1.5, 1.5;set term postscript eps enhanced color solid 24],
[logx, false], [logy, false]]
```

and after that, every plot command will be handled by Openmath using the Xmaxima graphic console, by default:

```
(%i5) plot2d (sin(x), [x, 0, 2*%pi]);
```

In that way you get a embedded graphic function sin(x).



NOTE: at page n. 6 Option: `gnuplot_preamble`

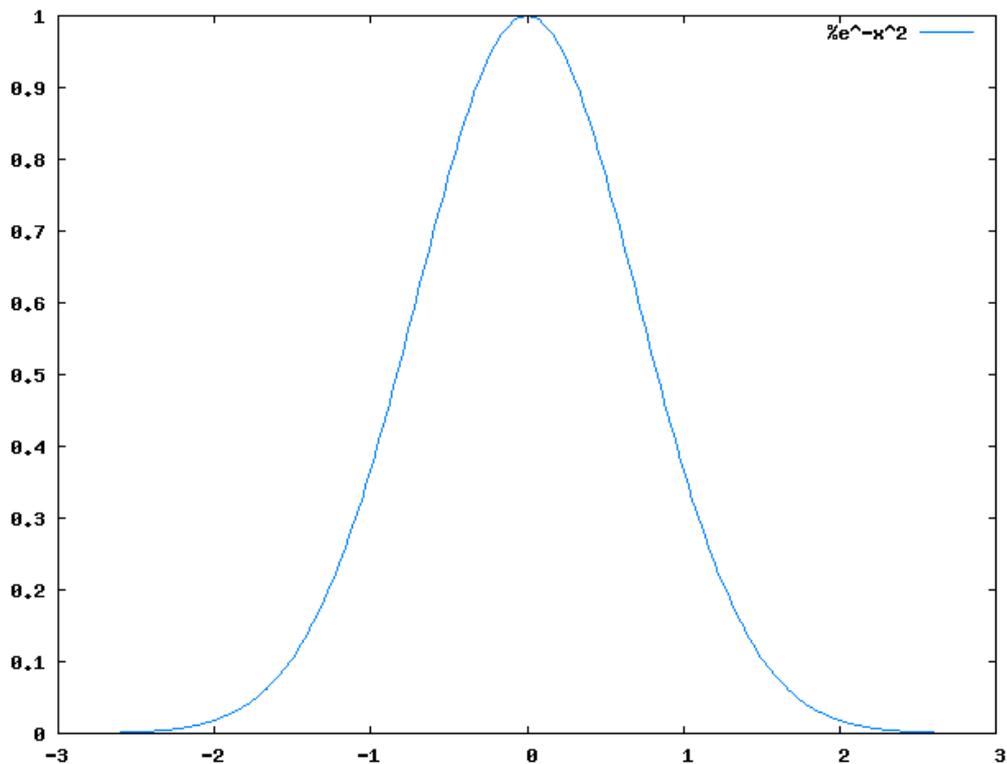
Inserts gnuplot commands before the plot is drawn. Multiple commands should be separated with a semi-colon. The example shown produces a PNG file. The default value for `gnuplot_preamble` is the empty string "".

Option: `grid`

Sets the number of grid points to use in the x- and y-directions for three-dimensional plotting.

## Composition of Functions:

```
(%i1) f(x) := -x^2$
(%i2) g(y) := %e^y$
(%i3) h(x) := g(f(x))$
(%i4) plot2d(h(x), [x, -3, 3], [x, -5, 5]);
```



## Maxima trigonometric functions defined

The trigonometric functions and simplification routines are in Maxima, below you can read the list of some of them:

**acos** ( $x$ ) – Arc cosine

**acot** ( $x$ ) - Arc Cotangent

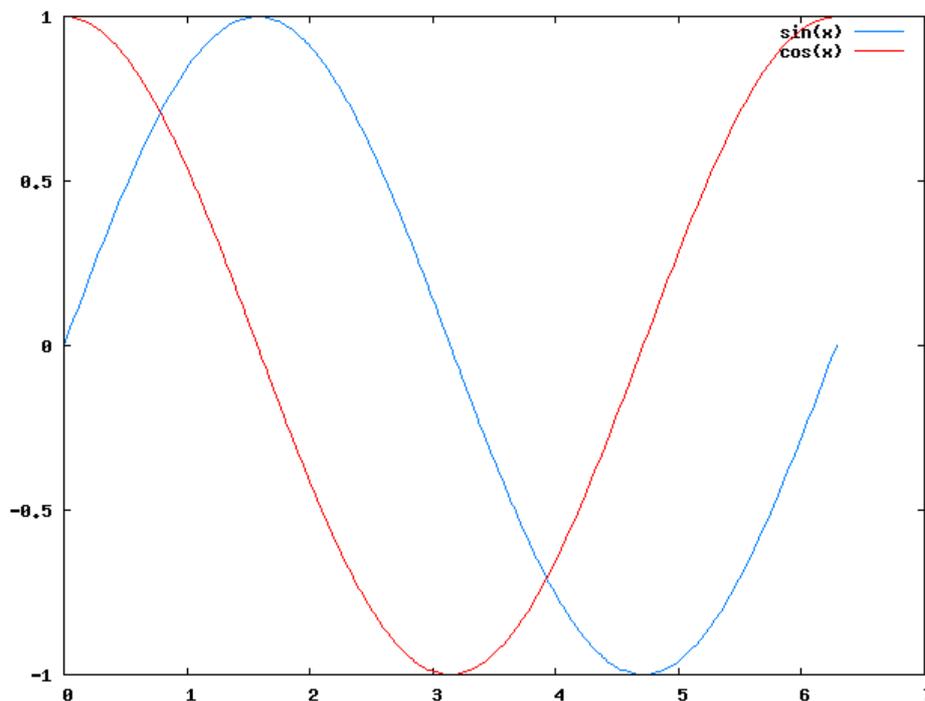
**acoth** ( $x$ ) - Hyperbolic Arc Cotangent.

**acsc** ( $x$ ) - Arc Cosecant.

**asec** ( $x$ ) - Arc Secant  
**asin** ( $x$ ) - Arc Sine.  
**atan** ( $x$ ) - Arc Tangent.  
**cos** ( $x$ ) - Cosine.  
**cot** ( $x$ ) - Cotangent.  
**csc** ( $x$ ) - Cosecant.  
**sec** ( $x$ ) - Secant.  
**sin** ( $x$ ) - Sine  
**tan** ( $x$ ) - Tangent.

Introducing the command line

`plot2d ([sin(x),cos(x)], [x, 0,2*pi],[y, -1,1]);` you get both the graphics.



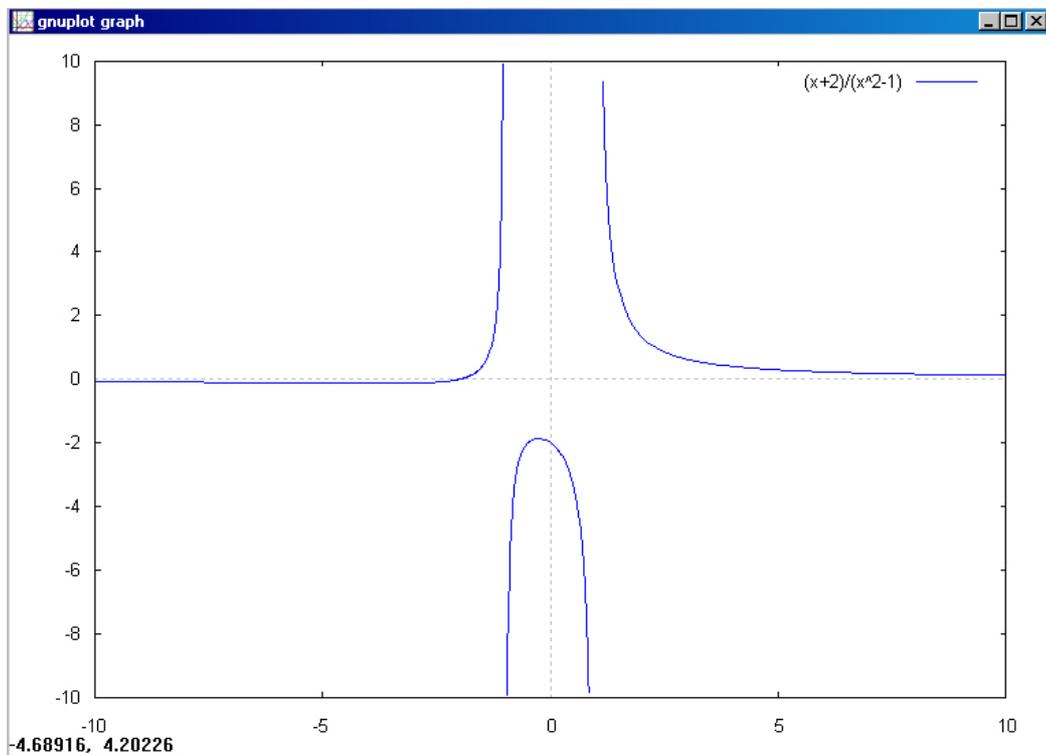
## Read data from file

There are several mechanisms to store data in files and retrieve it.

- **save** writes data to a file as Lisp expressions. **load** retrieves the data written by **save**. Execute `? save` and `? load` for more information.
- **stringout** writes data to a file as Maxima expressions. **load** retrieves the data written by **stringout**. Execute `? stringout` for more information.
- **writefile** creates a console log (a so-called "dribble" file) which records the interactions between the user and Maxima. This file cannot be reloaded into Maxima. Execute `? writefile` for more information.

## The study function $y = \frac{x+2}{x^2-1}$ using MAXIMA

```
(%i1) f(x):= (x+2)/(x^2 - 1)$  
(%i2) plot2d (f(x), [x, -10,10],[y, -10,10],[gnuplot_preamble,"set xzeroaxis;  
set yzeroaxis"])$
```



## MAXIMA computes the function limit

```
(%in) limit (expr, x, val, dir);
```

*expr* as the real variable *x* approaches the value *val* from the direction *dir*.

*dir* may have the value `plus` for a limit from above, `minus` for a limit from below, or may be omitted (implying a two-sided limit is to be computed).

Maxima limit uses the following special symbols: `inf` (positive infinity) and `minf` (negative infinity).

On output it may also use `und` (undefined), `ind` (indefinite but bounded) and `infinity` (complex infinity).

```
(%i3) limit (f(x), x, -1, plus);  
(%o3) minf  
(%i4) limit (f(x), x, -1, minus);  
(%o4) inf  
(%i5) limit (f(x), x, 1, plus);  
(%o5) inf  
(%i6) limit (f(x), x, -1, minus);  
(%o6) inf  
(%i7) limit (f(x), x, inf);  
(%o7) 0
```

## First Derivative

```
(%i8) diff(f(x),x);
```

```
(%o8) 
$$\frac{1}{x^2 - 1} - \frac{2x(x+2)}{(x^2 - 1)^2}$$

```

Expression can be simplified by **ratsimp**:

```
(%i9) Deriv1:ratsimp(diff(f(x),x));
```

```
(%o9) 
$$-\frac{x^2 + 4x + 1}{x^4 - 2x^2 + 1}$$

```

```
(%i10) solve (Deriv1=0,x);
```

```
(%o10) [x = - sqrt(3) - 2, x = sqrt(3) - 2]
```

```
(%i11) float(%o10);
```

```
(%o11) [x = - 3.732050807568877, x = - 0.26794919243112]
```

We use the

Function: `ev (<expr>, <arg_1>, ..., <arg_n>)` to calculate the coordinates of the point .

Evaluates the expression `<expr>` in the environment specified by the arguments `<arg_1>, ..., <arg_n>`. The arguments are (assignments, equations, and functions. `'ev'` returns the result (another expression) of the evaluation.

```
(%i12) ev(f(- 3.732050807568877),float);
```

```
(%o12) - 0.13397459621556
```

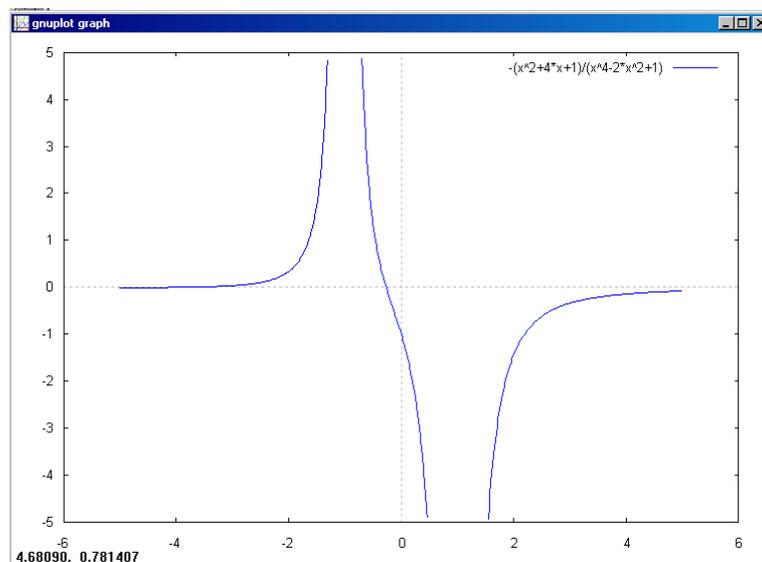
```
(%i13) ev(f(- 0.26794919243112),float);
```

```
(%o13) - 1.866025403784439
```

You can read below the command line to get of the graphics of first derivative:

```
(%i14) plot2d (Deriv1, [x, -5,5],[y, -5,5],[gnuplot_preamble,"set xzeroaxis; set yzeroaxis"])$
```

and in the next page its picture.



## Second Derivative

```
(%i15) Deriv2:ratsimp(diff(Deriv1,x));
```

```
(%o15) 
$$\frac{2x^3 + 12x^2 + 6x + 4}{x^6 - 3x^4 + 3x^2 - 1}$$

```

```
(%i16) fpprec:2$
```

```
(%i17) solve (Deriv2=0,x);
```

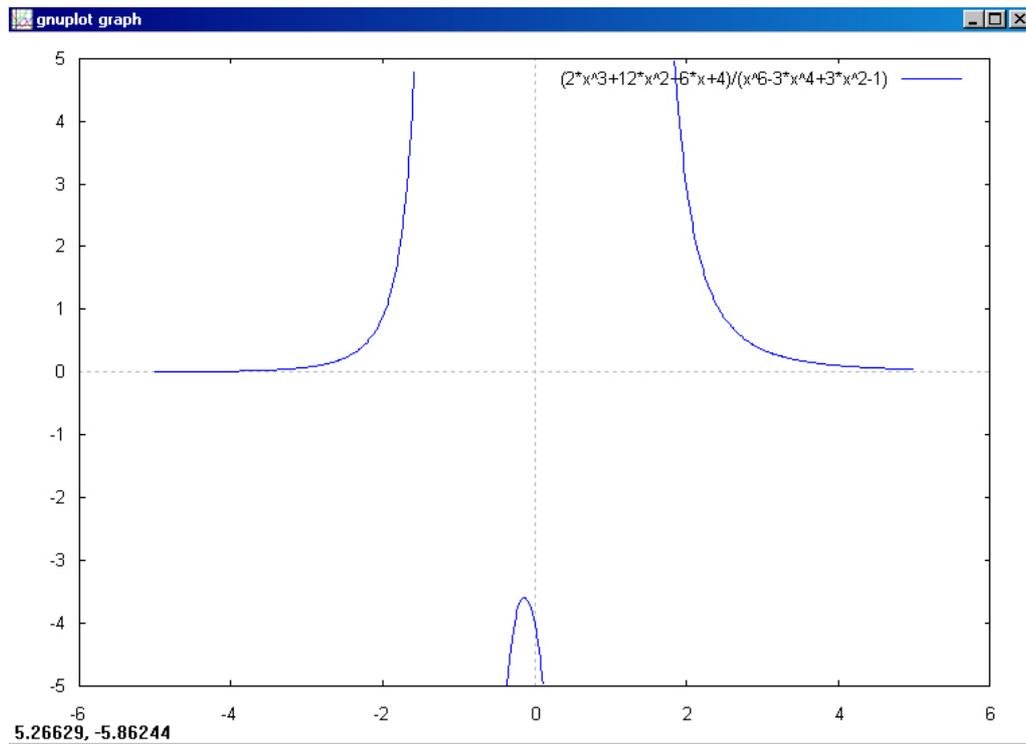
```
.....  
(%i19) float(%o17);
```

```
(%o19) [x = - 2.1 (0.9 %i - 0.5) - 1.4 (- 0.9 %i - 0.5) - 2.0, x = - 1.4 (0.9 %i - 0.5) - 2.1 (- 0.9 %i - 0.5) - 2.0, x = - 5.5]
```

You can read below the command line to get of the graphics of second derivative:

```
(%i20) plot2d (Deriv2, [x, -5,5],[y, -5,5],[gnuplot_preamble,"set xzeroaxis; set yzeroaxis"])$
```

and in the next page its picture.



## wxMaxima release (0.6.5-1)

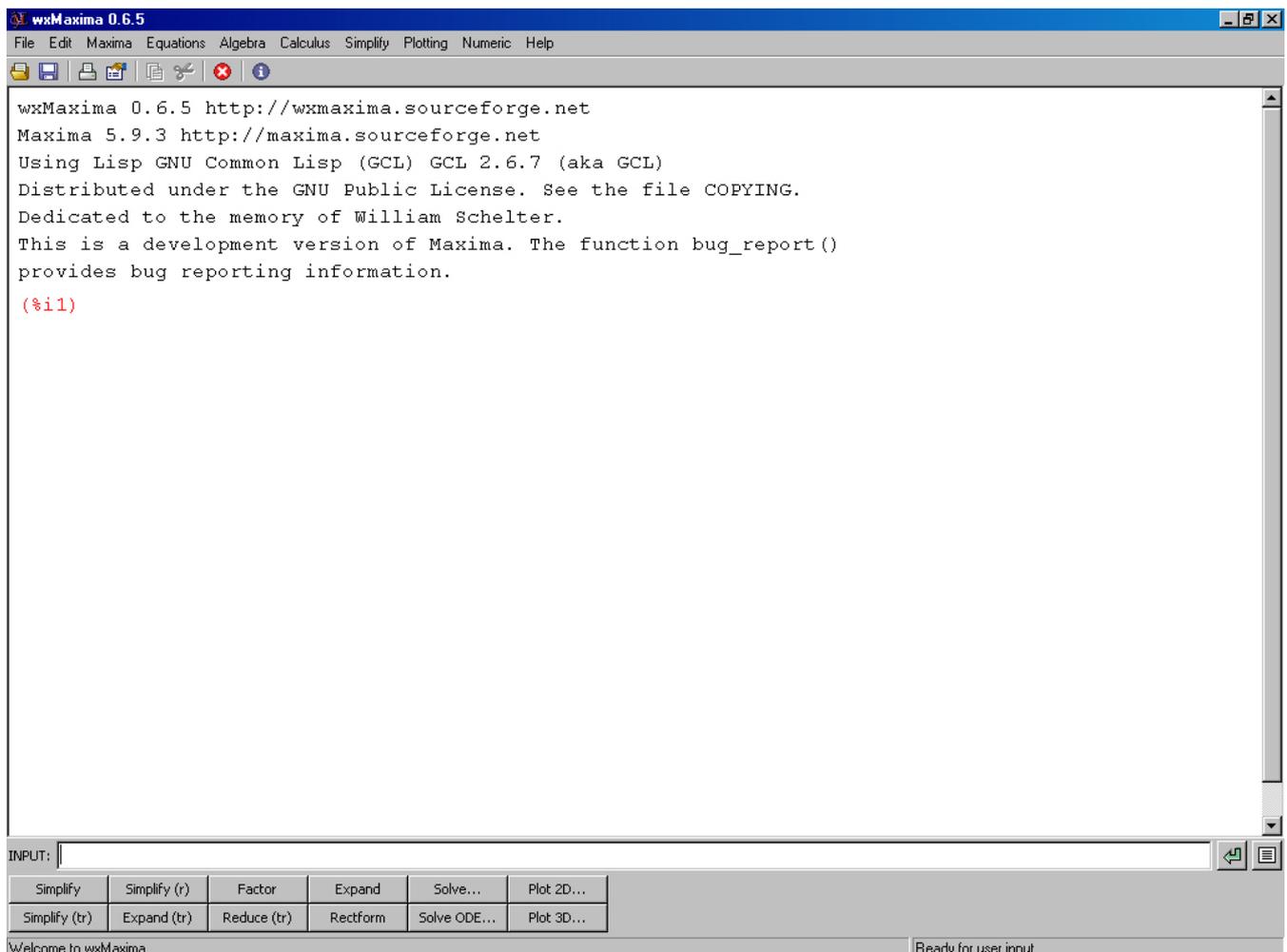
A very interesting related project about Maxima is **wxMaxima** release (0.6.5-1) (author Andrej Vodopivec et al) a wxWidgets <http://www.wxwidgets.org/> graphical user interface for the computer algebra system Maxima.

Wxmaxima (free download at <http://wxmaxima.sourceforge.net/>) provides a wxWindows interface for Maxima, including a menu system, browsable help and formatted output (see picture below in this page). Features as symbolic manipulation of polynomials, rational functions, matrices, integration and graphics are as easy as point and click.

Wxmaxima should be installed in a subdirectory of Maxima installation directory. This way, wxMaxima can find the Maxima binary automatically.

Example: - Maxima installation directory           C:\Program Files\ maxima-5.9.3  
          - wxMaxima installation directory       C:\Program Files\ maxima-5.9.3\wxMaxima.

If you install it in a different directory, you will have to configure it with Maxima->Configure.



# WEB RESOURCES

wxMaxima material to download by Johann Weilharter  
<http://maxima.weilharter.info>

Introduction to Maxima by Richard Rand (PDF)  
<http://maxima.sourceforge.net/docs/intromax/intromax.pdf>

The Computer Algebra Program Maxima by Boris Gaertner  
(Entire tutorial in one file: [gaertner-tutorial-revision.zip](http://maxima.sourceforge.net/docs/tutorial/en/minimal-maxima.pdf))

Micro introduction into Maxima at Harvard UK <http://www.math.harvard.edu/computing/maxima/>

Minimal Maxima by Robert Dodier (PDF) <http://maxima.sourceforge.net/docs/tutorial/en/minimal-maxima.pdf>

<http://www.ugcs.caltech.edu/manuals/math/maxima-5.9.2/index.html> Document generated by *Joshua David Goldstein* on *December, 6 2005*

