1 Introduction to *Maxima*

Maxima is a symbolic-based mathematical software providing a number of functions for algebraic manipulation, calculus operations, matrix and linear algebra, and other mathematical calculations.

Maxima web page

The *Maxima* web page is located at:

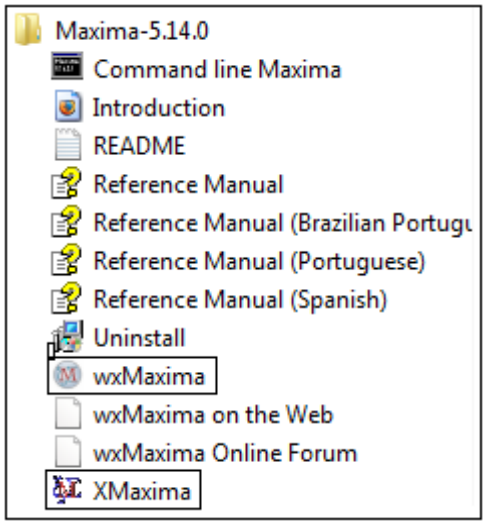
http://maxima.sourceforge.net/

Read the description of *Maxima* shown in this page. The page also includes a number of links including a *Download* link. Download and install *Maxima* in your computer as indicated in the download page.

The *Maxima* web page also includes a *Documentation* link with a number of tutorials on the use of *Maxima*.

xMaxima and *wxMaxima*

The figure below shows the listing of programs and documents available for *Maxima 5.14.0* in a *Windows Vista* installation.



You will notice that there are two possible instances of *Maxima* called *XMaxima* and *wxMaxima*. While both allow the user access to the Maxima commands, the difference is in the graphic user interface (GUI) used to communicate with *Maxima*.

XMaxima

An example of the *XMaxima* interface is shown in Figure 1.1. The top of the GUI is the input window for *Maxima* commands. The lower part is a display of a *Maxima* Primer document providing the user with some information about getting started with *Maxima*. In between the top and lower part of the display you will find buttons labeled *File*, *Back*,

Forward, *Edit*, *Options*, and *Url*: The last button refers to the file specification shown in the field immediately to its right. In this case, the file specification reads:

file:/C:/PROGRA~/MAXIMA~1.0/share/maxima/514~1.0/xmaxima/INTRO~1.HTM

The full reference to this file should be:

file:/C:/Program Files/Maxima-5.14.0/share/maxima/5.14.0/xmaxima/intro.html

The *XMaxima* GUI abbreviates some of the sub-folders in the first file specification producing the reference shown above, which could be a bit confusing. The full file specification shows the location of the file being shown in the bottom window of the *XMaxima* GUI. This *html* file is located in the *Maxima* installation as indicated above.

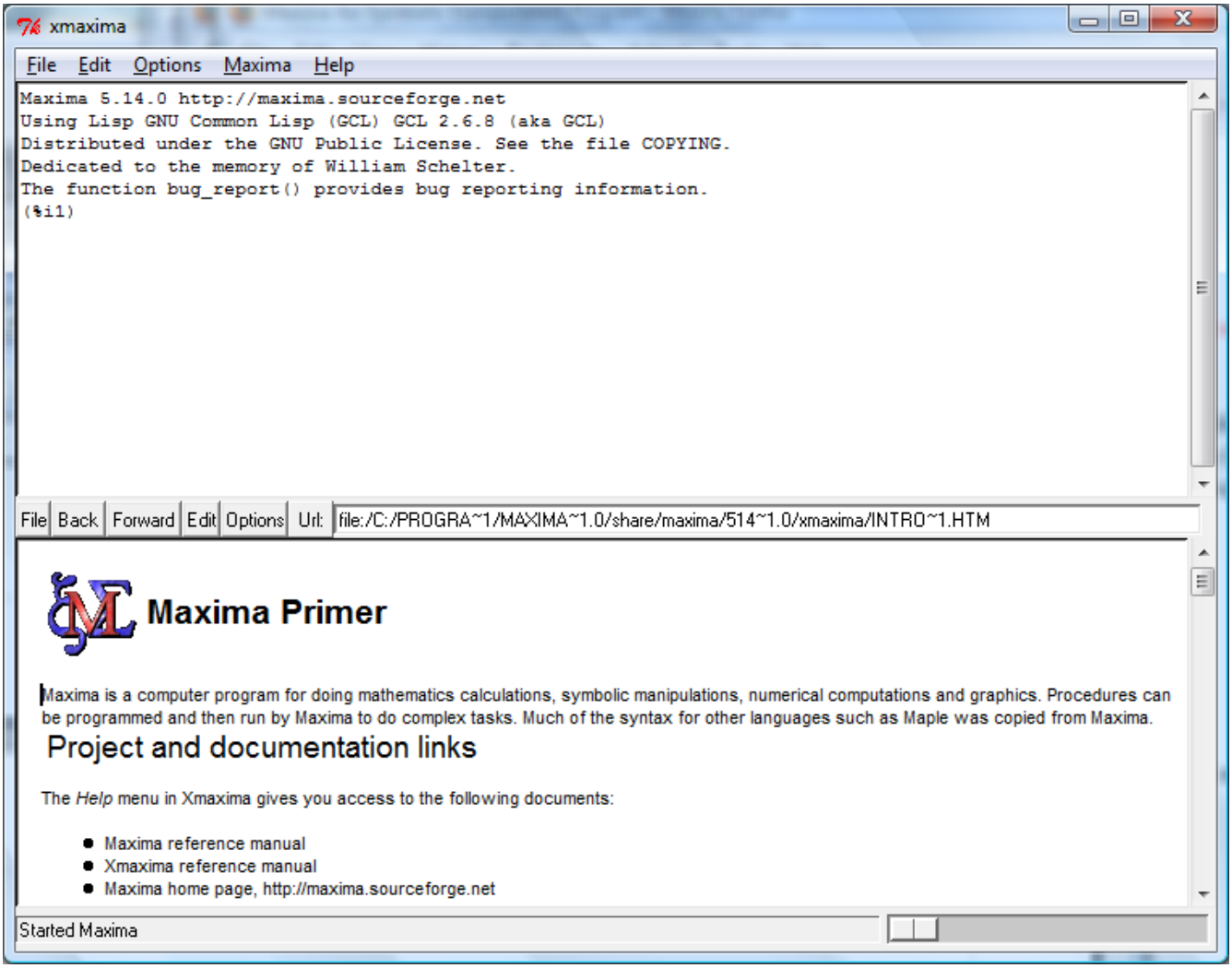
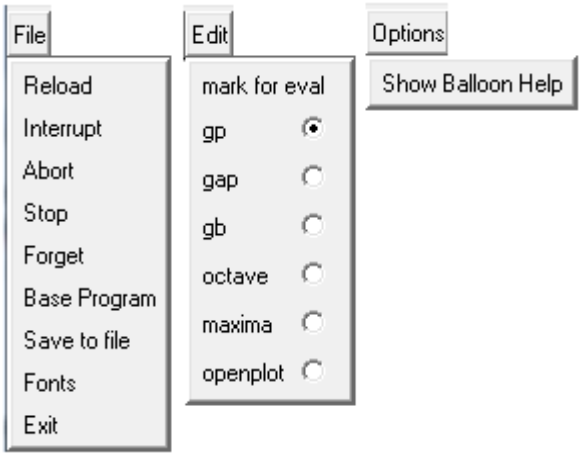


Figure 1.1. *XMaxima* starting GUI

The *Back* and *Forward* buttons allow the user to move about the document, while the other buttons provide the following menu items:



Using the *Maxima Primer*

Scroll down the *Maxima Primer* document to learn about the use of *Maxima*. One of the first applications is presented in the following paragraph (lifted from the document):

To do basic operations, a line is typed, followed by a semicolon, and then entered. This can be done in the window above. Alternately you may edit the blue portions in this buffer, and click on them, to see the result evaluated above and/or inserted in this window, depending on what was specified in the html source for this file. For example clicking below

- `integrate(1/(1+x^3),x)`

You may double click the above formula, and the integral will be substituted into the Maxima evaluation in the other window. There are examples which you may also look at 3d plotting If you wish to have your plots appear in a separate window, go to the preferences button under file, and select separate. You may also go to the netmath page to see some more capabilities.

Double-click on the *integrate* command shown in the *Maxima Primer* to see *Maxima* in action in the *XMaxima* window. The top window will now show the following operation:

```
Maxima 5.14.0 http://maxima.sourceforge.net
Using Lisp GNU Common Lisp (GCL) GCL 2.6.8 (aka GCL)
Distributed under the GNU Public License. See the file COPYING.
Dedicated to the memory of William Schelter.
The function bug_report() provides bug reporting information.
(%i1)
                                                2 x - 1
                                    2         atan(-------)
                                 log(x  - x + 1)    sqrt(3)     log(x + 1)
(%o1)                          - --------------- + ------------- + ----------
                                        6            sqrt(3)          3
(%i2)
```

Notice that there are two *input* locations labeled *(%i1)*, or input 1, and *(%i2)*, or input 2. Input 1 *(%i1)* is missing any input. This is so, because by double-clicking the *integrate* line

in the *Maxima Primer*, we activated the input without copying it to the top window. The result, however, is available in the top window as output 1 *(%o1)*. Also, notice that *XMaxima* presents the result of the integral as closely as possible as a two-dimensional mathematical expression, i.e.,

```
                                                 2 x - 1
                                      2       atan(-------)
                                   log(x  - x + 1)    sqrt(3)     log(x + 1)
(%o1)                            - --------------- + ------------- + ----------
                                         6             sqrt(3)          3
```

as opposite to a one-dimensional mathematical entry, i.e.,

```
-log(x^2-x+1)/6+ atan((2*x-1)/sqrt(3))/sqrt(3) + log(x+1)/3.
```

The full mathematical operation calculated in this example can be, on paper, written as

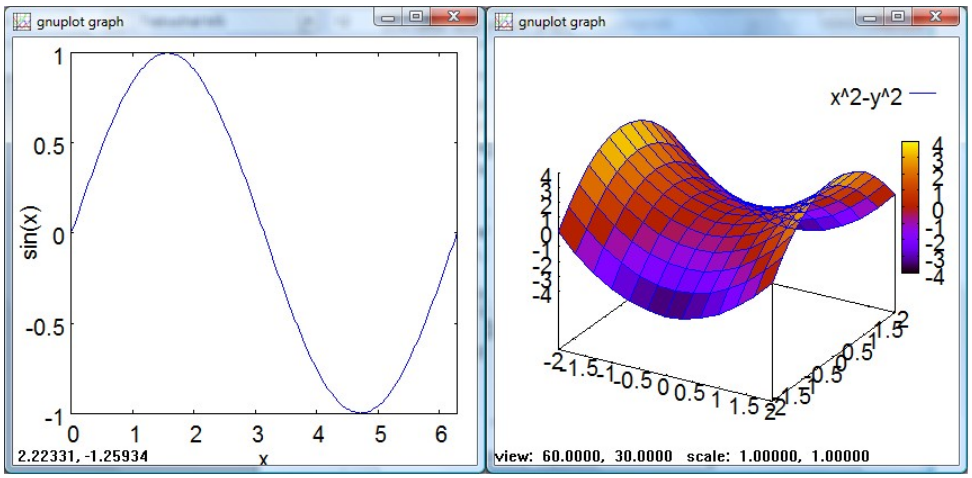
$$\int \frac{dx}{1+x^3} = -\frac{ln(x^2-x+1)}{6} + \frac{tan^{-1}\left(\frac{2*x-1}{\sqrt{3}}\right)}{\sqrt{3}} + \frac{log(x+1)}{3} \quad .$$

The user is invited to continue reading the *Maxima Primer* document and double-click on the different examples listed to learn the basic operation of *Maxima*. Following those exercises, one may notice, for example, that in the *XMaxima* interface, the mathematical constant π (the ratio of the length of a circumference to its diameter) is referred to as `%pi`. Also, infinity (∞) is referred to as *inf*.

The *Maxima Primer* examples include also plots that are produced in their own separate graphics window, e.g., the commands

- `plot2d(sin(x),[x,0,2*%pi])`
- `plot3d(x^2-y^2,[x,-2,2],[y,-2,2],[grid,12,12])`

produce, respectively, the two-dimensional and a three-dimensional graphs shown below.



Click-off the graphical windows before continuing with the other commands in the *Maxima Primer*.

wxMaxima
wxMaxima uses an interface as shown in Figure 1.2, below.

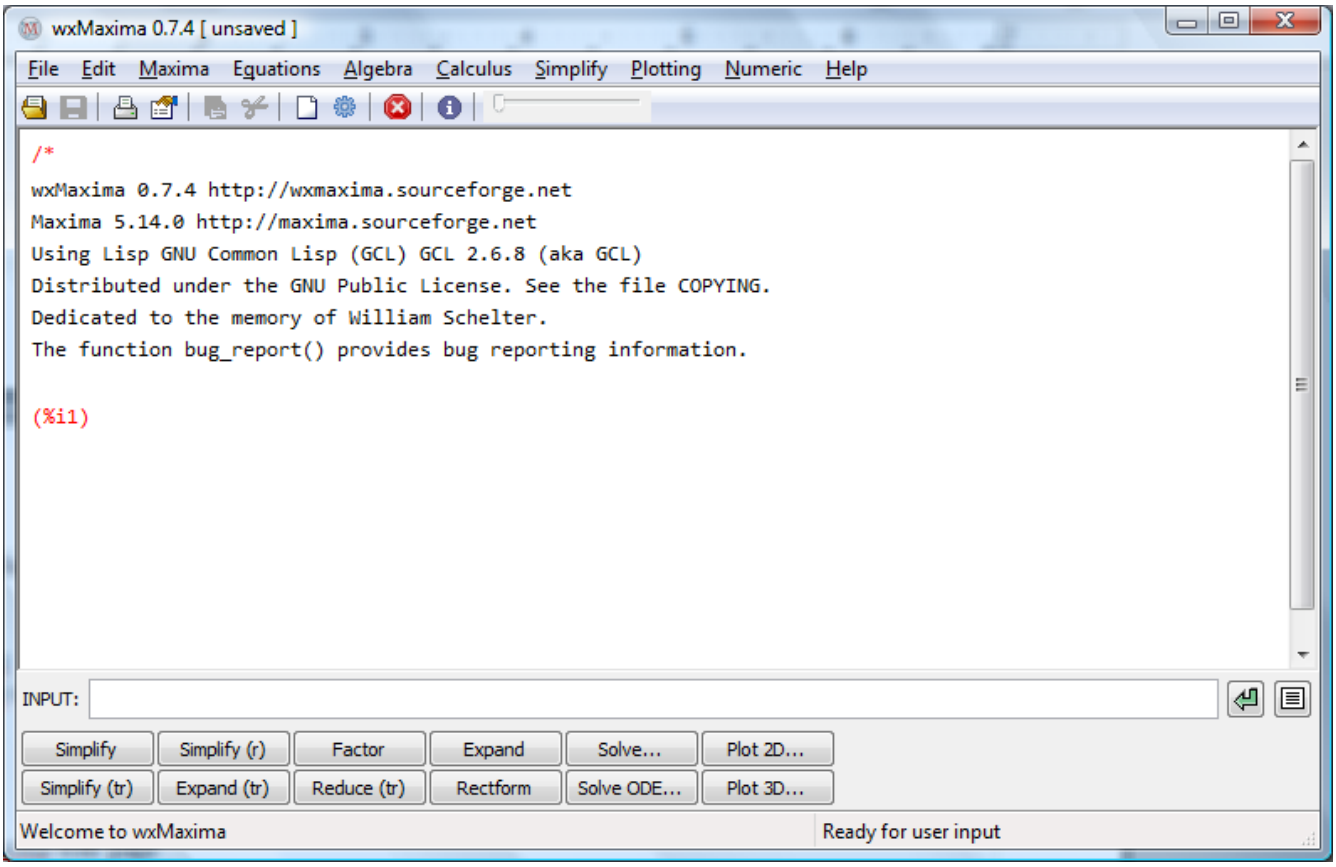


Figure 1.2. The *wxMaxima* GUI.

This interface is more sophisticated than that of *XMaxima* for the following reasons:

- *wxMaxima* produces true two-dimensional mathematical output
- *wxMaxima* provides most *Maxima* commands in menus (e.g., *Equations, Algebra, etc.*)
- Some commands can be activated by using the buttons shown at the bottom of the interface, e.g., *Simplify, Factor,* etc.
- *wxMaxima* provides dialogues to enter parameters of selected commands.
- *wxMaxima* maintains a command line history buffer where previously used commands can be accessed, repeated, or edited.
- *wxMaxima* allows mixing text with mathematical expressions to produce printable documents.
- The current version of *wxMaxima* supports simple animations (to see the current version use the menu item *Help* > *About*).

A web page for *wxMaxima* is available here:

http://wxmaxima.sourceforge.net/wiki/index.php/Main_Page

For hints on the efficient use of *wxMaxima* visit:

http://wxmaxima.sourceforge.net/wiki/index.php/Howto

NOTE: Because of the additional features available in *wxMaxima*, we will use this GUI exclusively to present the examples contained in this and subsequent chapters. We will not be using *XMaxima* anymore in this or subsequent chapters.

***wxMaxima* menus**

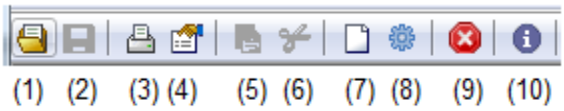
Take some time to explore the different menus in the *wxMaxima* GUI:

- The *File* menu contains items typically found in windows-based applications such as *Open, Read file, Save, Save As..., Export to HTML, Select File, Print,* and *Exit*. Some items in the *File* menu, such as *Load package*, *Batch file*, and *Monitor File*, are proper of *wxMaxima*.
- The *Edit* menu contains typical commands such as *Copy, Cut*, and *Paste*, as well as others that are proper for *wxMaxima*.
- The *Maxima* menu contains items that allow the user to control the operation of *Maxima*.
- The *Equations*, *Algebra*, *Calculus*, *Simplify*, *Plotting*, and *Numeric* menus provide mathematical functions that are entered using dialogues.
- The *Help* menu contains several items of interest such as:
  - *Maxima help*: opens the *Maxima Manual* window with description and examples of *Maxima* commands.
  - *Describe*: produces a dialogue where the user can enter the name of a specific command. Try, for example, *plot3d*, and press OK. The dialogue will access the section of the *Maxima Manual* corresponding to the requested command.
  - *Example*: enters a series of examples of applications of the requested command into the *wxMaxima* interface. Try, for example, *integrate*, and press OK.
  - *Apropos*: use this dialogue to enter a keyword to search for a command that is similar to the keyword. For example, if you were seeking information on integration, you could enter the word *integra*, to get a listing of commands that may be related to *integra*. Then, you can use *Describe* or *Example* with one of the commands listed.
  - *Show tip*: shows tips on the use of *Maxima*.
  - *Build info*: provides information on the current version of *Maxima*.
  - *Bug report*: provides a web site where users can report errors in the operation of *Maxima*, or unexpected results of some operations. These "bugs" are reported to the programming team and solutions to them (if available) get incorporated in the new versions of the software.
  - *About*: provides the current version of *wxMaxima*. Notice that the versions of *Maxima* and *vxMaxima* are not necessarily the same. My installation, at the

moment of typing this book, showed *Maxima* version 5.14.0 and *wxMaxima* version 0.7.4. Remember that *Maxima* is the computer program that performs the mathematical calculations, while *wxMaxima* is the graphics user interface (GUI).

wxMaxima tool bar

The *wxMaxima* GUI provides a tool bar with the following buttons:



(1) Open session
(2) Save session
(3) Print document
(4) Configure wxMaxima
(5) Copy selection
(6) Delete selection
(7) Insert text
(8) Insert input group
(9) Interrupt current computation
(10) Show Maxima help (same as menu item *Help* > *Maxima help*)

Using the *INPUT* line

The *INPUT* line in the *wxMaxima* interface can be used for a variety of purposes such as:

- To perform a calculation, e.g., `sqrt(1+3.5^2)/sin(%pi/6);`
- To define one or more variables, e.g., `a:2; b:2;`
- To define a function, e.g., `f(x):=sqrt(1+x^2);`
- To evaluate a function, e.g., `f(2/3);`
- To produce a plot, e.g., `plot2d(f(x),[x,-2,2]);`
- To enter other type of operations, e.g., a derivative: diff(t^2*sin(t), t);

Here are some observations from the examples shown above:

- To enter the value of a variable use a colon (:)
- To define a function use a colon followed by the equal sign (:=)
- *Maxima* expressions end with a semi-colon. If you forget to enter the semi-colon in the *INPUT* line, *wxMaxima* will enter it for you.

This is additional information useful when entering expressions:

- Variable or function names must start with a letter, and may include letters, numbers, and undersign, e.g.,

```
vx:2; x2:3; y_2:5; Initial_Velocity:-2.5;
```

- The following are reserved words in *Maxima* and cannot be used as variable names:

```
integrate            next          from              diff
in                   at            limit             sum
for                  and           elseif            then
else                 do            or                if
unless               product       while             thru
step
```

Some pre-defined functions: Some of the common pre-defined functions in *Maxima* include:

sqrt	square root	*sin*	sine	*cos*	cosine
tan	tangent	*cot*	cotangent	*sec*	secant
csc	cosecant	*asin*	inverse sine	*acos*	inverse cosine
atan	inverse tangent	*acot*	inverse cotangent	*asec*	inverse secant
acsc	inverse cosecant	*exp*	exponential	*log*	natural logarithm
sinh	hyperbolic sine	*cosh*	hyperbolic cosine	*tanh*	hyperbolic tangent
asinh	hyperbolic asin	*acosh*	hyperbolic acos	*atanh*	hyperbolic atan
floor	integer below	*ceiling*	integer above	*fix*	integer part
float	conver to floating point			*abs*	absolute value

Maxima does not have a logarithm-base-10 function. Instead, use:

$$log_{10}(x)=\frac{log(x)}{log(10)}$$

Here are some examples you can try:

```
sin(2.5*%e);float(sin(2.5*%e));
floor(%pi);ceiling(%pi);
log(5);float(log(5));
k:float(log(3)/log(10));
float(10^k);abs(-2);fix(3.3);fix(-3.2);
```

Notice that *Maxima* will tend to give symbolic results (i.e., results including fractions, square roots, unevaluated trigonometric, exponential, or logarithmic functions) rather than floating-point (or numerical) results. Use function *float*, as in the examples above, to get floating-point solutions.

Automatic parentheses. Whenever you enter an opening parenthesis in the *INPUT* line, a closing parenthesis is added automatically. If you are not used to this feature, you may end up entering more closing parentheses than needed. This situation will result in an error that is easy to spot.

The percentage (%) operator. The percentage (%) symbol represents the most recent result. Try these examples:

```
exp(-2.5)*sin(3*%pi/11);float(%);exp(-3);float(%);log(5);float(%);
```

To access the second-to-last commans use `%2`, the third-to-last, use `%3`, and so on.

Mathematical constants. Some of the common mathematical constants available in *Maxima* are:

`%e`	base of the common logarithms (=`exp(1)`)
`%i`	imaginary unit (=`sqrt(-1)`)
`inf`	real positive infinity
`minf`	real negative infinity
`infinite`	complex infinity
`% phi`	the golden ratio (ϕ)
`% pi`	ratio of length of circumference to its diameter (π)
`%gamma`	Euler's constant (γ)
`false, true`	boolean values (or logical values)

Here are some examples to try (in some examples we use function *is* to check whether comparisons of numbers are true or false):

```
float(%phi);float(%pi);float(%e);%gamma;
is(3>2);is(3<2);is(x<3);
integrate(exp(-x^2/2),x,-inf,inf);integrate(exp(-x^2/2),x,minf,inf);
```

Some examples of complex numbers. The unit imaginary number *i* is entered as `%i` in *Maxima*. Here are some examples of complex number calculations:

```
z1:3+5*%i; z2:-2+6*%i;z1+z2;z1-z2;expand(z1*z2);expand(z1^2);
```

The following functions apply to complex numbers:

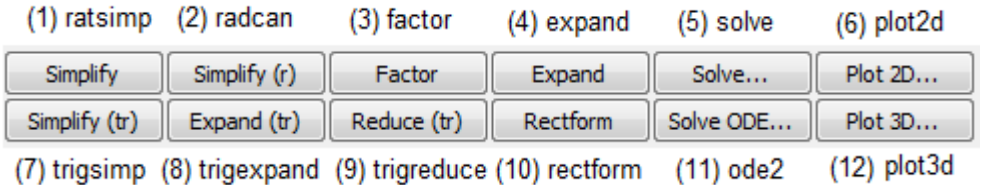
cabs	(*c*omplex *abs*olute value) calculates the modulus
carg	(*c*omplex *arg*ument) calculates the argument
rectform	generate rectangular (Cartesian) form
polarform	generate polar form
realpart	extract the real part
imagpart	extract the imaginary part
conjugate	calculates the complex conjugate

The following examples illustrate some of these functions:

```
cabs(z1);arg(z1);
z2;-z2;conjugate(z2);expand(z2*conjugate(z2));
rectform(z1/z2);rectform(sqrt(z1));polarform(z1);polarform(z2);
```

Using the button panel

The bottom of the *xwMaxima* GUI contains 12 buttons that can be used for common operations. The collection of buttons is shown in the figure below, with the *Maxima* commands associated with them.



Buttons (1) through (4), and (7) through (10) operate on an expression typed in the *INPUT* line before pressing the corresponding button. Buttons (5), (6), (11), and (12) trigger dialogues to performed the associated operations. The operation of the buttons, with appropriate examples, is shown next.

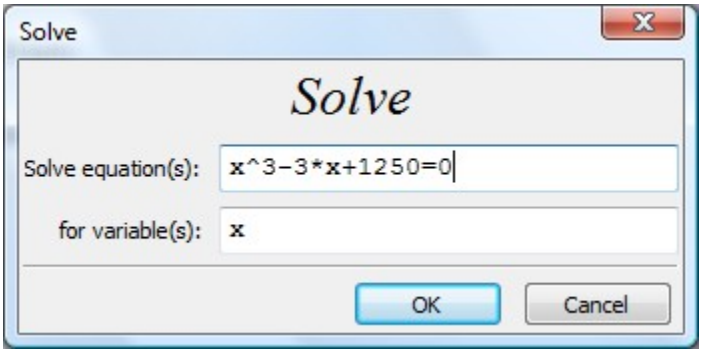
(1) Simplify: simplifies algebraic operations, e.g., `(x+2)*(x-2); [Simplify]`

(2) Simplify(r): simplifies expressions containing logs, exponentials, and radicals, e.g., `(%e^x-1)/(%e^(x/2)+1); [Simplify(r)]`

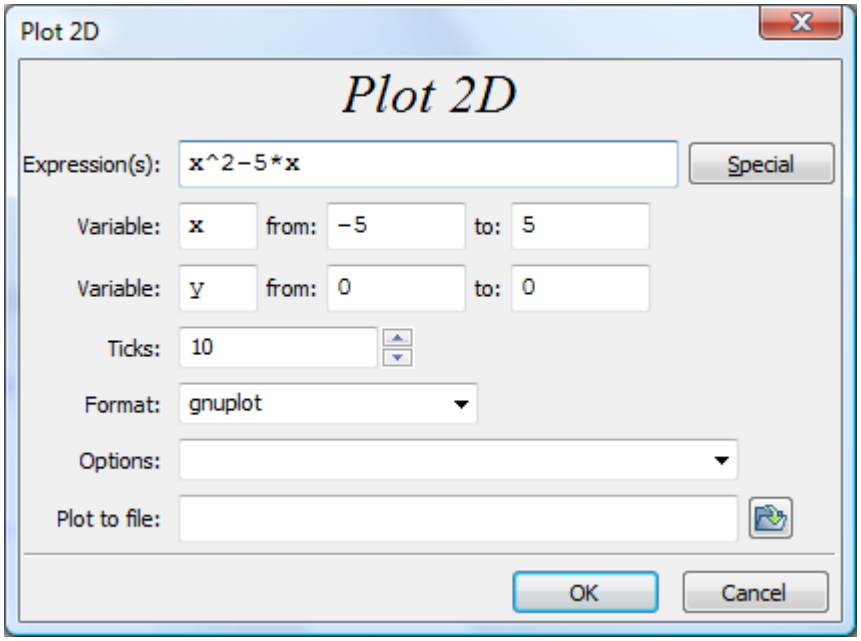
(3) Factor: factors an algebraic expression, e.g., `x^2+y^2-2*x*y; [simplify(r)]`

(4) Expand: expands an algebraic expression, e.g., `(x+1)*(x-1)*(x^2+1); [Expand]`

(5) Solve...: solves an equation, e.g.,



(6) Plot 2D...: produces an *x*-*y* (two dimensional) plot, e.g.,



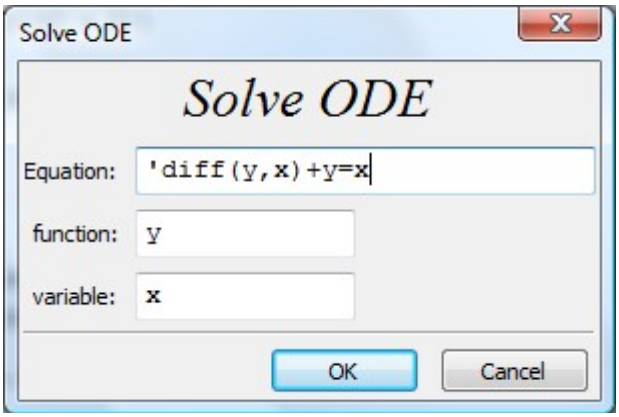
(7) Simplify(tr): trigonometric simplification in terms of *sin* and *cos*,e.g., `tan(x) [Simplify (tr)]`

(8) Expand(tr): expands a trigonometric expression, e.g., `sin(x+y) [Expand(tr)]`

(9) Reduce(tr): convert powers of trigonometric functions to those of multiples of the angle, e.g., `x+3*cos(x)^2-sin(x)^2; [Reduce(tr)]`

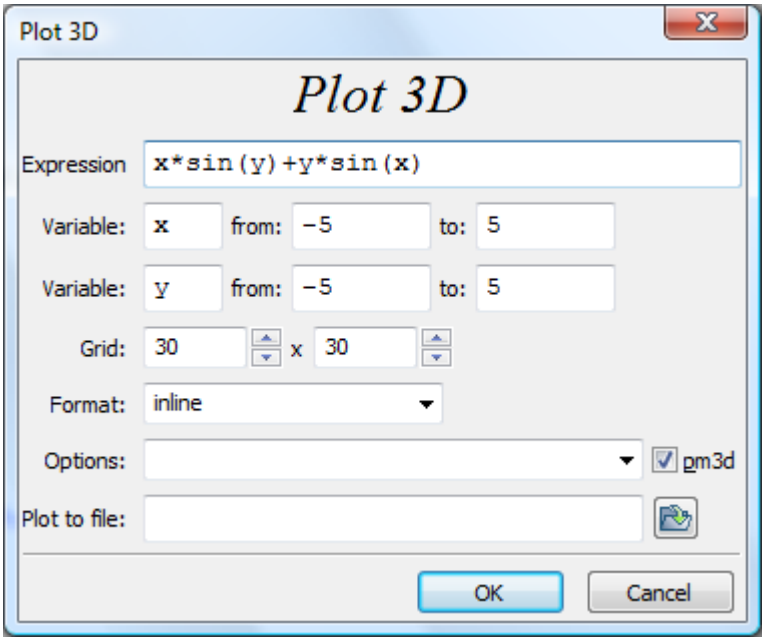
(10)Rectform: produces the rectangular form of a complex number, e.g., `1/(2+3*%i);` [Rectform]

(11)Solve ODE...: solves a 1st order or 2nd order ordinary differential equation, e.g.,



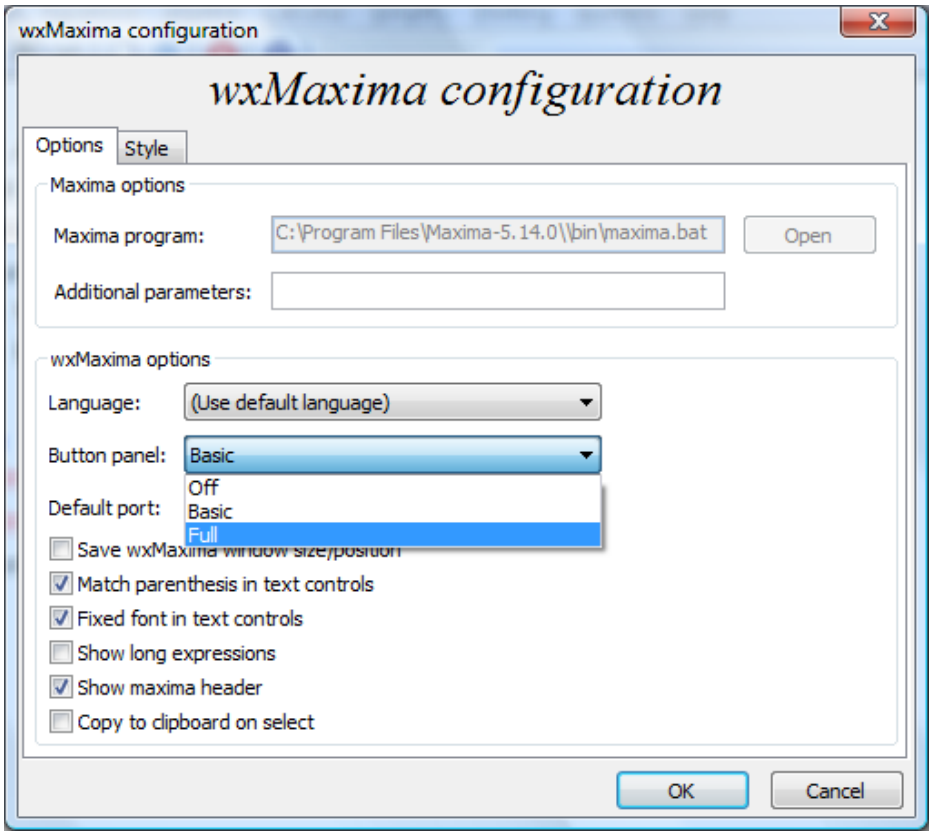
Note: Derivatives are written using `'diff(y,x,n)` to represent $\frac{d^n y}{dx^n}$.

(12) Plot3D: produces a three-dimensional plot, e.g.,

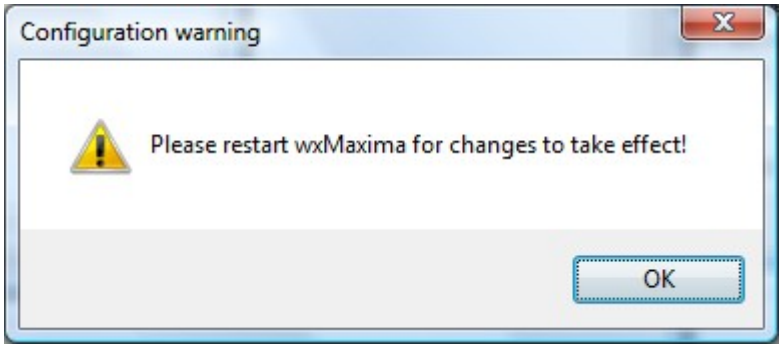


Using the full button panel

The button panel described above is referred to as the *Basic* button panel. It is possible to activate a *Full* button panel by using the menu option *Edit* > *Configure*. This activates a *wxMaxima configuration* window as shown next:

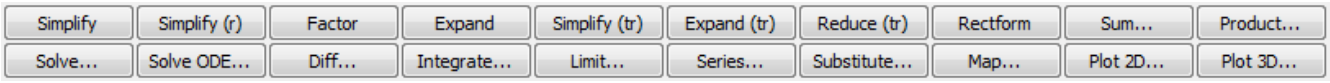


Select the option *Full* in the *Button panel* drop-down menu to activate the *Full* button panel, and press [OK]. *wxMaxima* will respond with the following message:

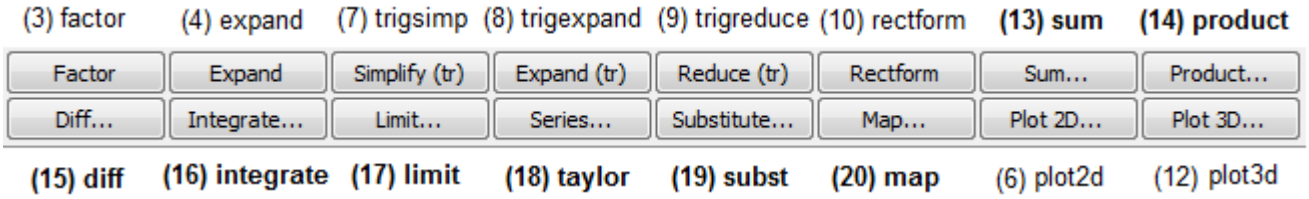


Press [OK] in this message form, and then [OK] again in the *wxMaxima configuration* window, and click off *wxMaxima*. The *Full* button panel will not be active until you re-start *wxMaxima*.

When you re-start *wxMaxima*, the bottom part of the interface will show the *Full* button panel:

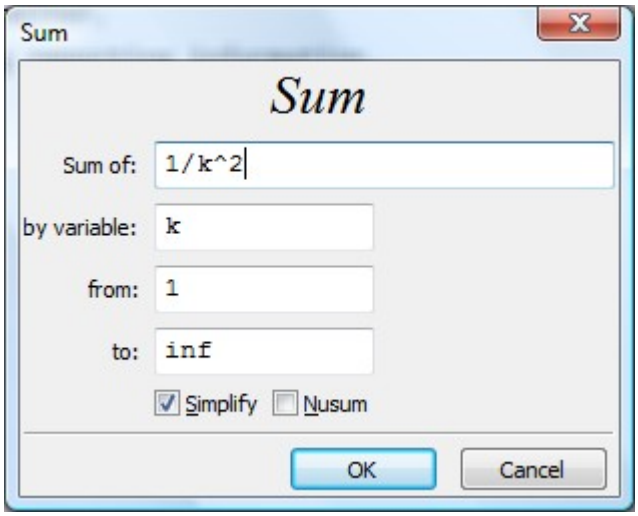


which now includes 20 buttons, instead of the 12 buttons of the *Basic* button panel. The new buttons are shown in the following figure, labeled (13) to (20), with labels shown in boldface letters (no all the buttons are shown):

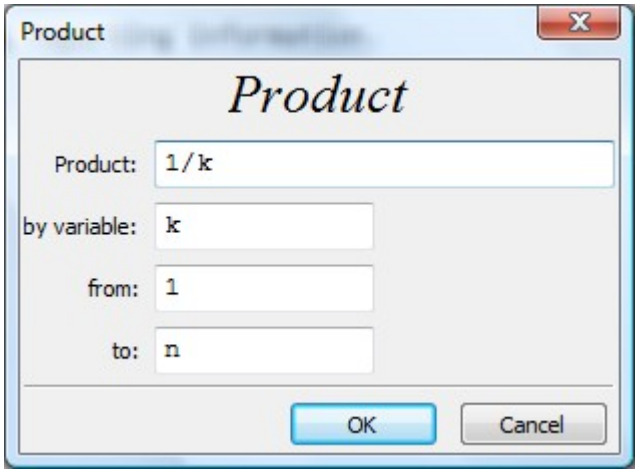


The operation of buttons (13) through (20) is described below:

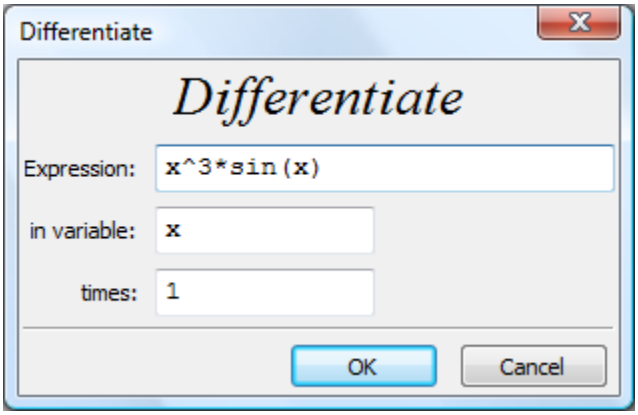
(13) *sum*: allows setting up and calculating a summation, e.g.,



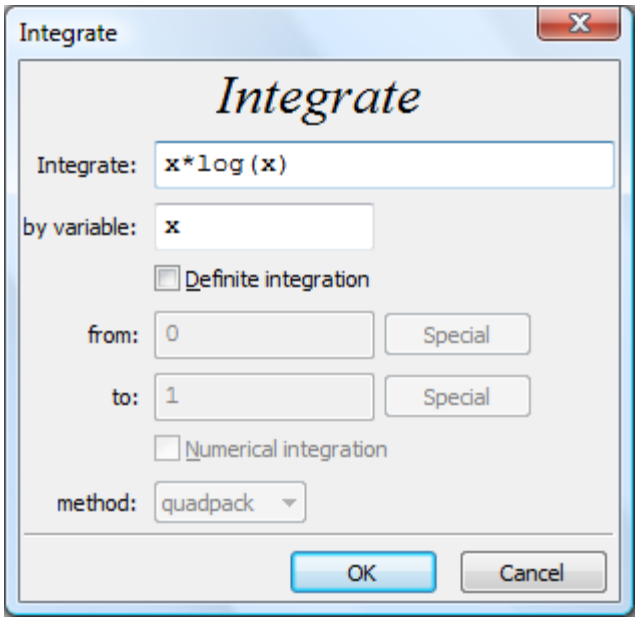
(14) *product*: allows setting up and calculating a product, e.g.,



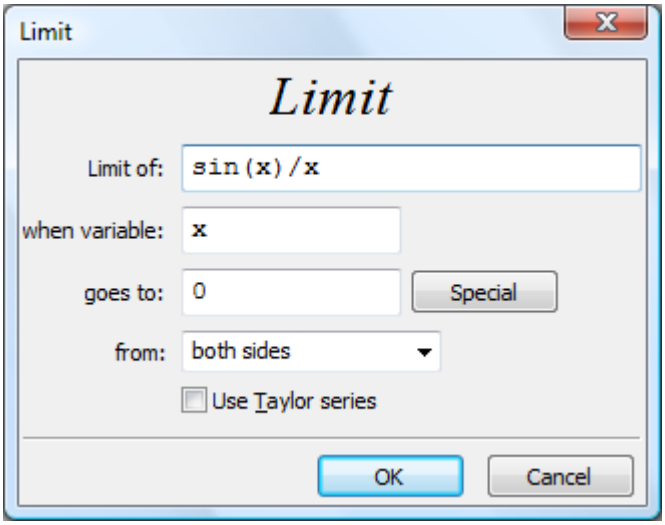
(15) *diff*: calculates a derivative, e.g.,



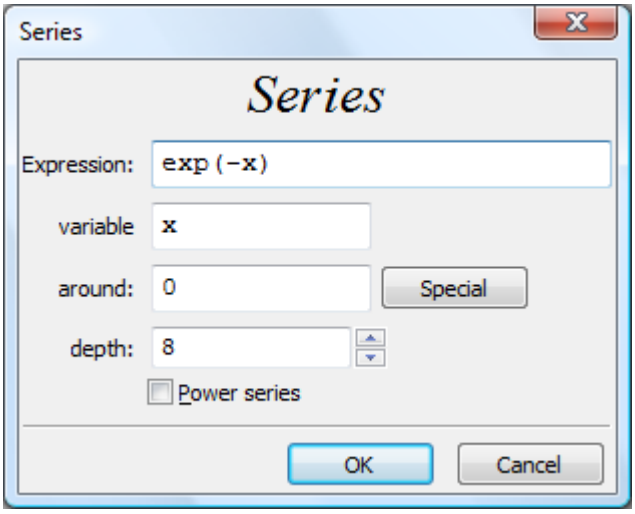
(16) *integrate*: calculates an integral



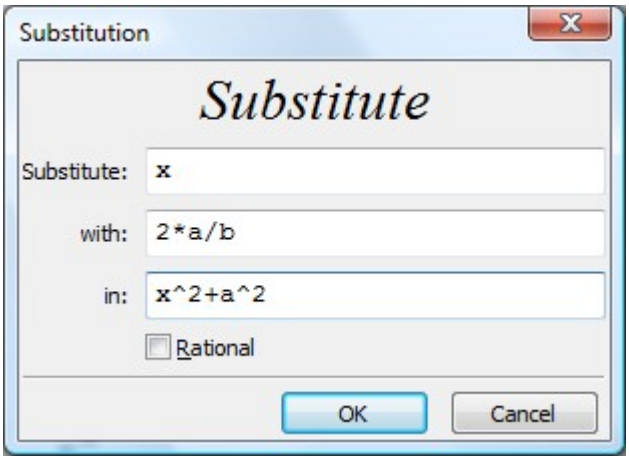
(17) *limit*: calculates a limit



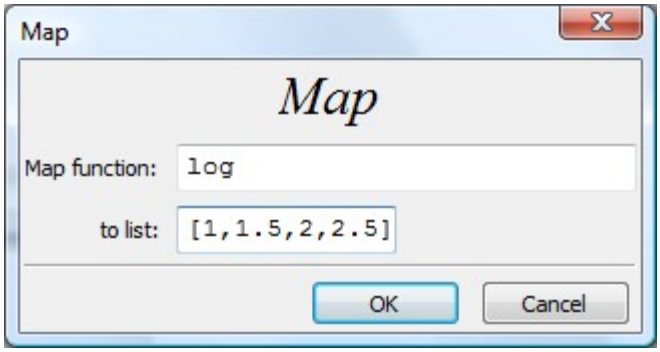
(18) *taylor*: calculates a Taylor series for an expression:



(19) *subst*: substitute an expression into a variable name



(20) *map*: maps a function to a list



Using Greek letters
In order to write Greek letters in *Maxima* you need to have the font *SPIonic* installed in your computer. You can download this font from:

http://www.drshirley.org/fonts/SPIonic.ttf

After installing the font in your computer, you need to select it to show Greek characters in your *wxMaxima* interface. Proceed as follows:

- Select the menu option *Edit>Configure*
- Click on the *Style* tab
- Check-off the *Use greek font* entry, and select *SPIonic*
- Press OK

To enter Greek letters type the English name of the letter in an expression, or precede the name with the percentage symbol (%), e.g.,

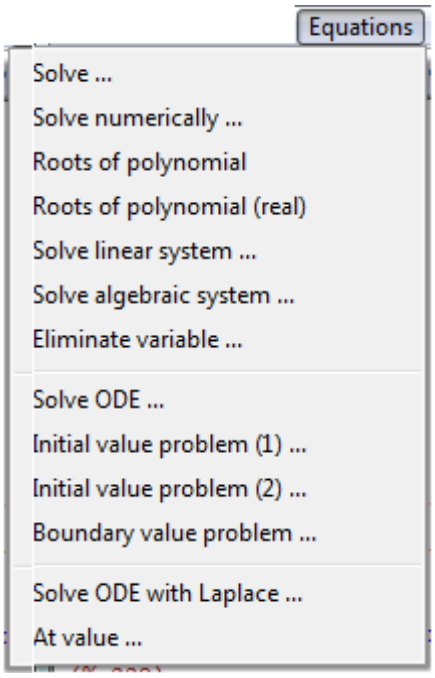
```
factor(beta^2-1);
rectform(1/(%alpha+%beta*%i);
expand((alpa-1)*(beta+gamma));
expand((%alpha-1)*(%beta+%gamma));
```

Notice the difference between typing *gamma* and *%gamma* in the last two examples. Typing *gamma* (without %) produces the upper-case Greek letter Γ which represents the Gamma function from mathematics, whereas, %gamma produces Euler constant γ, defined, as the limit as n , of the quantity

$$\sum_{k=1}^{n} \frac{1}{k} - ln(n) \quad .$$

To illustrate the use of the Gamma function try the following exercises in *wxMaxima*:

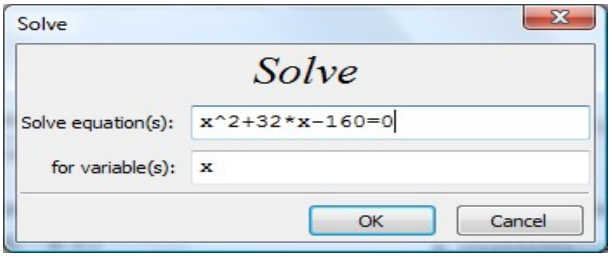
```
gamma(2.5);
plot2d(gamma(x),[x,0.5,3.0]);
```



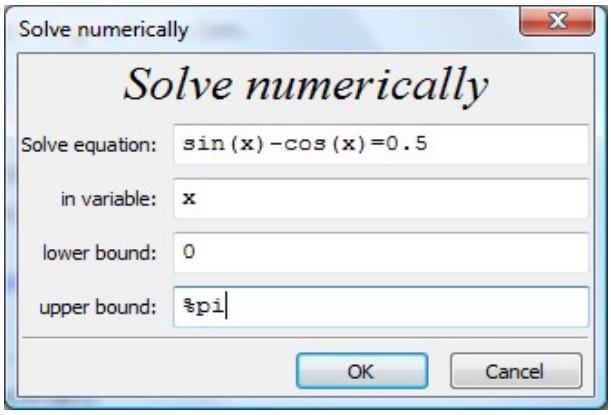
Examples from the *Equations* menu
In this section we present some examples of applications from the *Equations* menu. We use it to illustrate the use of menus such as *Equations*, *Algebra,* Calculus, etc. A listing of the available applications in the *Equations* menu is shown below:

Try the following examples by selecting entries from this menu:

- *Solve* ... same as: `solve([x^2+32*x-160=0], [x]);`



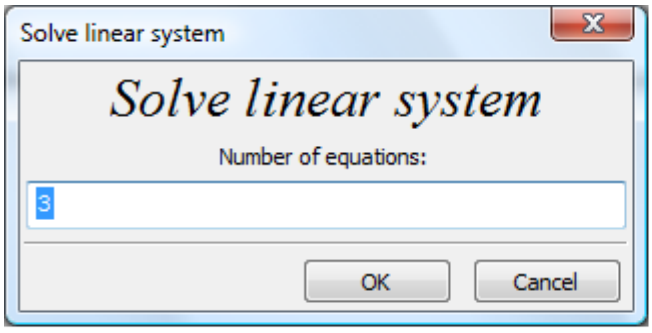
- *Solve numerically ...* equivalent to `find_root(sin(x)-cos(x)=0.5, x, 0, %pi);`

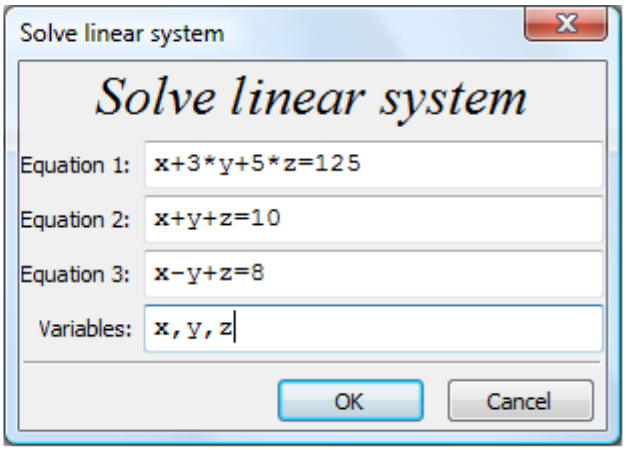


- *Roots of polynomial* -- Try: `x^3+25*x^2-5*x+212=0;` *Equations > Roots of polynomial*. Equivalent to `allroots(%);`

- *Roots of polynomial (real)* - Try: `x^3+25*x^2-5*x+212=0;` *Equations > Roots of polynomial (real)*. Equivalent to `realroots(%);`

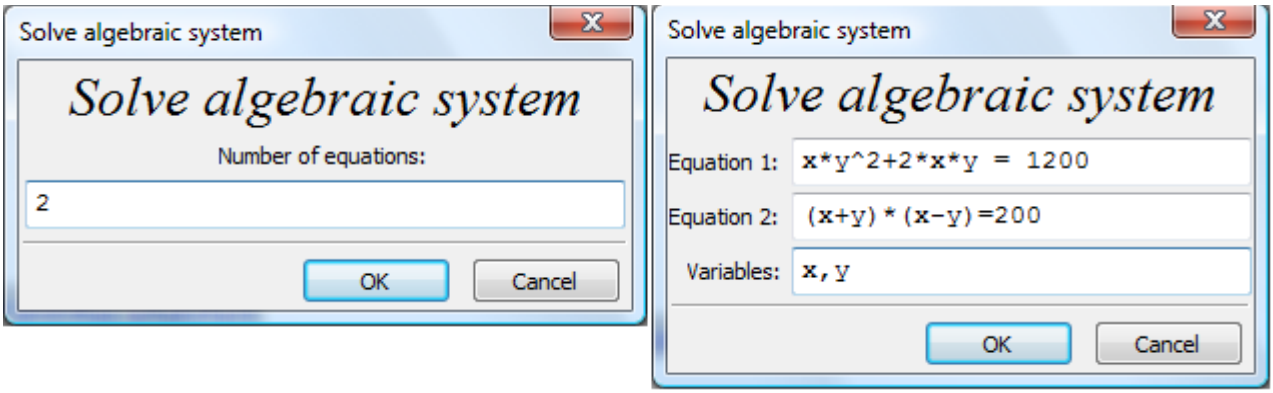
- *Solve linear system ...* equivalent to
  `linsolve([x+3*y+5*z=125, x+y+z=10, x-y+z=8], [x,y,z]);`



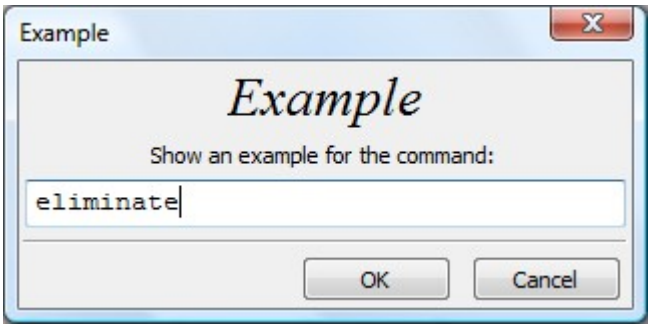


- *Solve algebraic system* ... equivalent to

```
algsys([x*y^2+2*x*y = 1200, (x+y)*(x-y)=200], [x,y]);
```

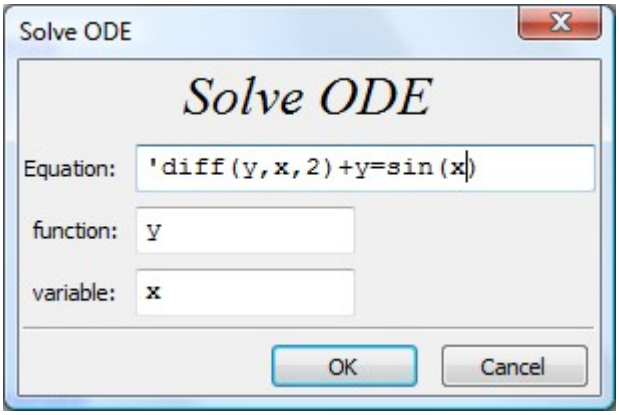


- *Eliminate variable* ... See the example available in the *Maxima Manual* by selecting the menu option *Help* > *Example...*, and type *eliminate*:

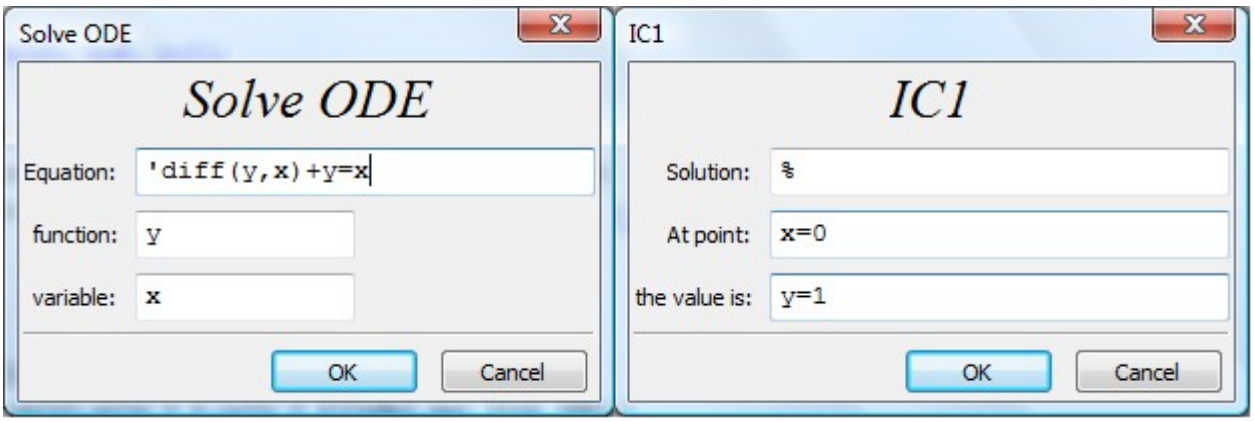


- *Solve ODE*... This is the same as pressing the button [Solve ODE...]. Equivalent to:

```
ode2('diff(y,x,2)+y=sin(x), y, x);
```

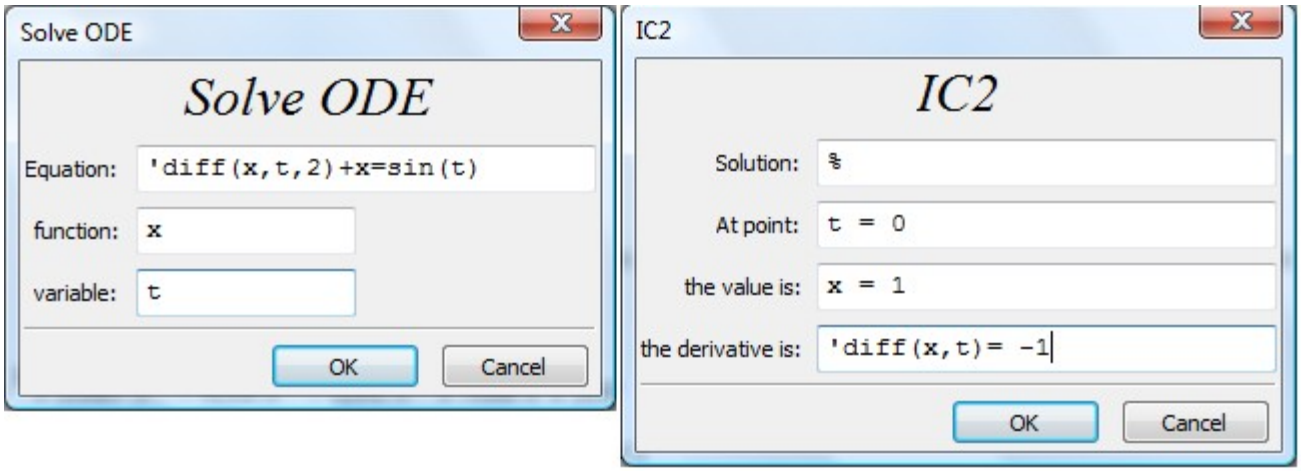


- *Initial value problem (1)* ... Initial value problem for first-order ODE. Uses two steps, first *Solve ODE* ..., then *Initial value problem (1)*. Equivalent to:



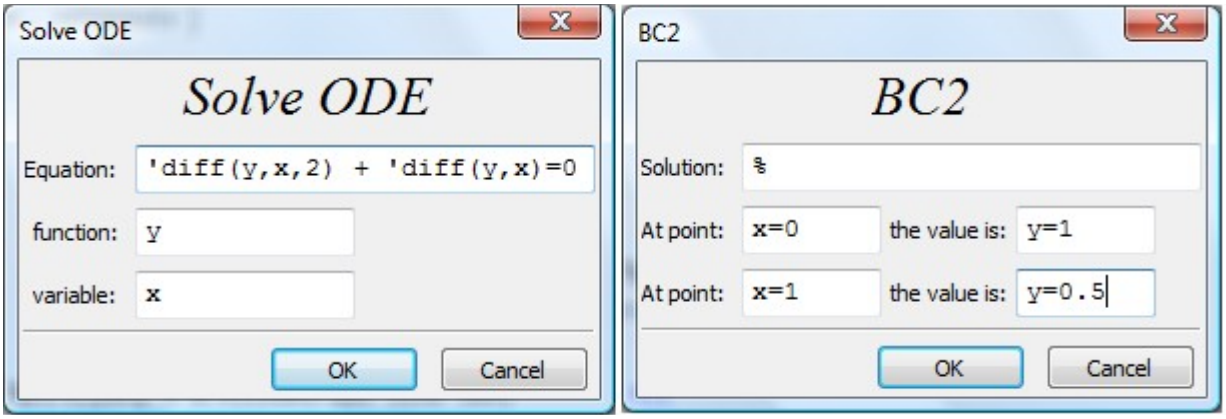
- *Initial value problem (2)* ... Initial value problem for second-order ODE. Uses two steps: first *Solve ODE* ..., then *Initial value problem (2)*. Equivalent to:

```
ode2('diff(y,x)+y=x,y,x); ic1(%,x=0,y=1);
```

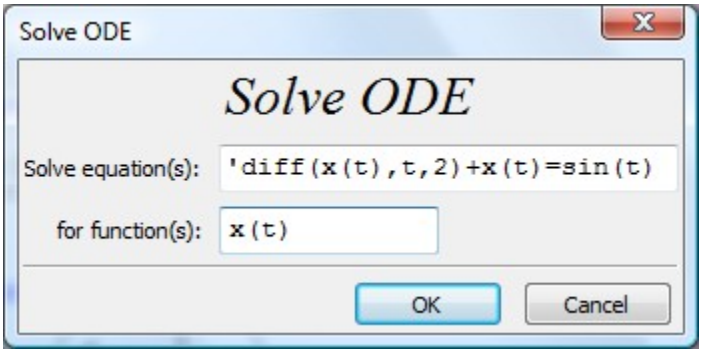


- *Boundary value problem* ... Boundary value problem for second-order ODE. Uses two steps: first *Solve ODE* ..., then *Boundary value problem*. Equivalent to:

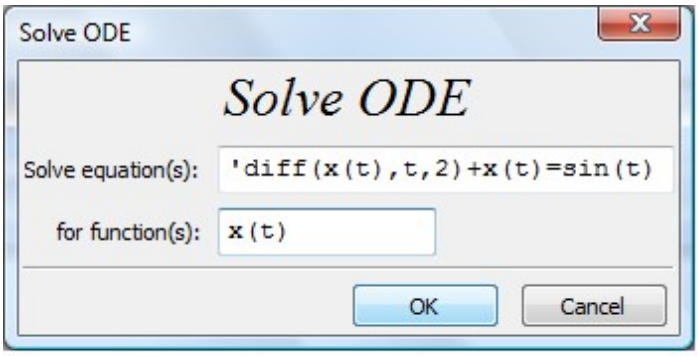
```
ode2('diff(x,t,2)+x=sin(t),x,t); ic2(%,t=0,x=1,'diff(x,t)=-1);
```

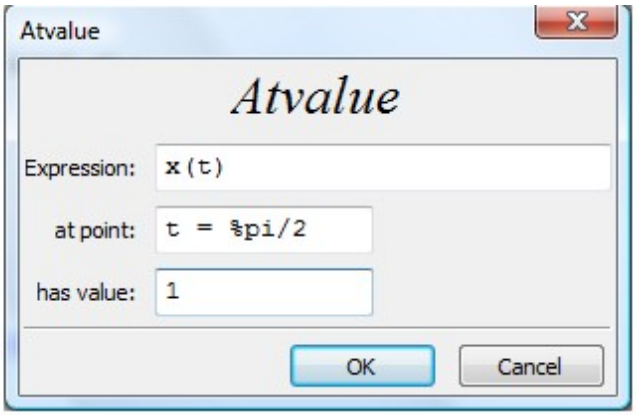


- *Solve ODE with Laplace* – Solve an ordinary differential equation using Laplace transforms. Equivalent to `desolve('diff(x(t),t,2)+x(t) = sin(t),x(t));`



- At value ... - Replace a variable in an expression. In this example the replacement takes place in the solution to an ODE.





Managing a *wxMaxima* session

In this section we illustrate the use of inputs and outputs and of the command history to perform operations on algebraic expressions.

Inputs and outputs

If you have been trying the examples shown above, your *wxMaxima* interface would show a number of inputs and outputs. Inputs are shown by the prompt `(%i...)` with an associated number, e.g.,

```
(%i84) desolve(['diff(x(t),t,2)+x(t)=sin(t)],[x(t)]);
(%i85) atvalue(x(t), t = %pi/2, 1);
```

Outputs are shown by the prompt `(%o...)` with an associated number, e.g.,

$$(\%o84)\quad x(t)=-\frac{\sin(t)}{2}-\frac{t\cos(t)}{2}+\cos(t)$$

$$(\%o85)\quad 1$$

Restarting *Maxima*

Since the inputs and outputs in your *wxMaxima* interface will be different than this example, let's restart *Maxima* by using the menu option *Edit* > *Restart Maxima*, and press [OK] at the prompt. This action will clear *Maxima*'s memory and reset the interface to that shown in Figure 1.2 (see above). At this point, only input `(%i1)` will be available. Let's try the following session. Type the commands as shown next:

```
(%i1)  (x-2)*(x-5)*(x-3)^2;
(%o1)  (x - 5)(x - 3)^2(x - 2)

(%i2)  sin(x)+cos(x+y);
(%o2)  cos(y + x)+ sin(x)

(%i3)  (x+2)/((x-2)*(x+5));
                  x + 2
(%o3)  ---------------------
       (x - 2)(x + 5)
```

Operations on input and output references

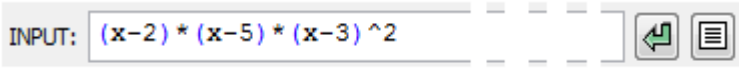
Now, we are going to use the input and output references to perform operations. Try the following commands:

```
(%i4)  expand(%i1);
(%o4)  x^4 - 13 x^3 + 61 x^2 - 123 x + 90

(%i5)  solve(%o1,x);
(%o5)  [x = 2, x = 3, x = 5]

(%i6)  trigexpand(%o2);
(%o6)  - sin(x)sin(y)+ cos(x)cos(y)+ sin(x)

(%i7)  expand(%o3^2);
                    x^2                      4 x                    4
(%o7)  ------------------------- + ------------------------- + -------------------------
       x^4 + 6 x^3 - 11 x^2 - 60 x + 100   x^4 + 6 x^3 - 11 x^2 - 60 x + 100   x^4 + 6 x^3 - 11 x^2 - 60 x + 100
```

Thus, references such as %i1, %o1, etc., act like variable names that can be operated upon as any other variable. An alternative would be to actually assign variable names to the expressions entered. To try this approach we will also illustrate the use of the command history for performing operations.

Using the command history in *wxMaxima*

Every single command that you enter in *wxMaxima* gets stored into a *command history* buffer. This buffer is accessible by clicking on the *INPUT* line and using the up and down arrow keys in your keyboard. As you press the upper arrow key once, the last command will be shown. As you keep pressing that key, the second-to-last command, third-to-last command, etc., will be shown in the *INPUT* line. For example, for the present exercise, click in the *INPUT* line, and press the upper-arrow key until you recover the very first expression entered, namely:



Then, use the left-arrow key to move the cursor to the left of the first parentheses, and type:

```
a:
```

then press [ENTER], or click on the *Enter command* button:  . The result is the following:

```
(%i8)  a:(x-2)*(x-5)*(x-3)^2;
(%o8)  (x - 5)(x - 3)^2(x - 2)
```

Now, we can refer to variable *a* for performing operations on this expression, e.g.,

```
(%i9)  expand(a);
(%o9)  x^4 - 13 x^3 + 61 x^2 - 123 x + 90

(%i10)  solve(a,x);
(%o10)  [x = 2, x = 3, x = 5]
```

The command history can be accessed, therefore, through the use of the up- and down-arrow keys in your keyboard. Once a command is accessed this way, you can either press [ENTER] to repeat it, or edit it in the INPUT line in order to perform a different operation.

End-of-line characters

It was mentioned earlier that every *Maxima* command ends in a semi-colon (;), and that if one fails to enter that end-of-line character, *wxMaxima* will enter it automatically. The fact is that, besides the semi-colon, there is also a *supress-output* character, namely, the dollar sign ($), which can be used as end-of-line character. Using the dollar sign ($) to end a *Maxima* statement suppresses the output of the command. However, the command gets executed in memory. For example, try the following commands:

```
(%i11)  r:25$ s:32$
(%i13)  r^2+s^2;
(%o13)  1649
```

In input (%i11), above, variables *r* and *s* are assigned the values *25* and *32*, respectively, but no output is shown because the statements end in a dollar sign ($), rather than in a semi-colon (;). However, output (%o13) shows that the statement r^2+s^2; was evaluated properly.

The use of the dollar sign ($) as end-of-line character saves space in the *wxMaxima* interface as illustrated in the following example:

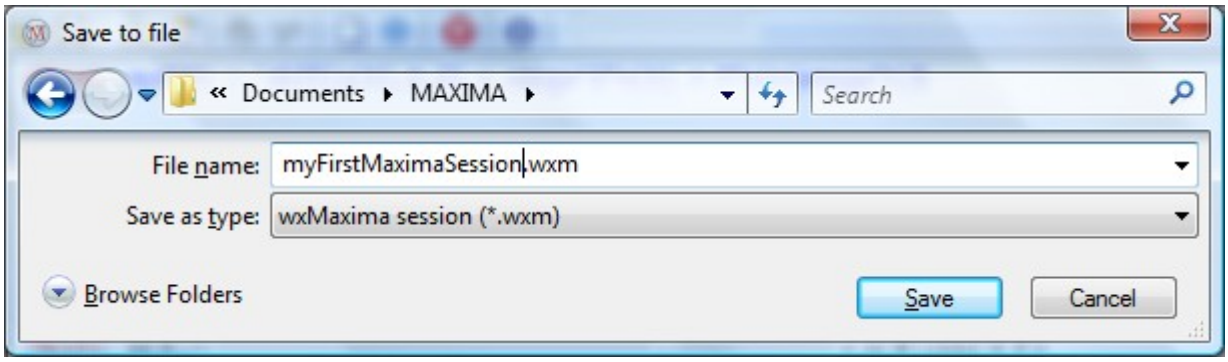
```
(%i14) myODE1 : 'diff(x(t),t,2) + omega^2*x(t) = F*sin(omega*t)$

(%i15) mySOL1 : desolve(myODE1,x(t));
Is  ω  zero or nonzero? nonzero;
```

$$(\%o15)\quad x(t)=\frac{\sin(\omega t)\left(F+2\omega\left(\frac{d}{dt}x(t)\Big|_{t=0}\right)\right)}{2\omega^2}-\frac{t\cos(\omega t)F}{2\omega}+x(0)\cos(\omega t)$$

Notice that the dollar sign ($) in input (%i14) suppresses the output for the differential equation *myODE1*. Also, notice the use of the Greek character *omega* (ω) as a coefficient in the differential equation *myODE1*. Furthermore, notice that, in attempting a solution for *myODE1*, *Maxima* doesn't know a-priori what the value of ω is. So, *Maxima* asks from the user whether ω is zero or nonzero. In this example, the user types *nonzero*, and *Maxima* returns the solution.

Saving your session

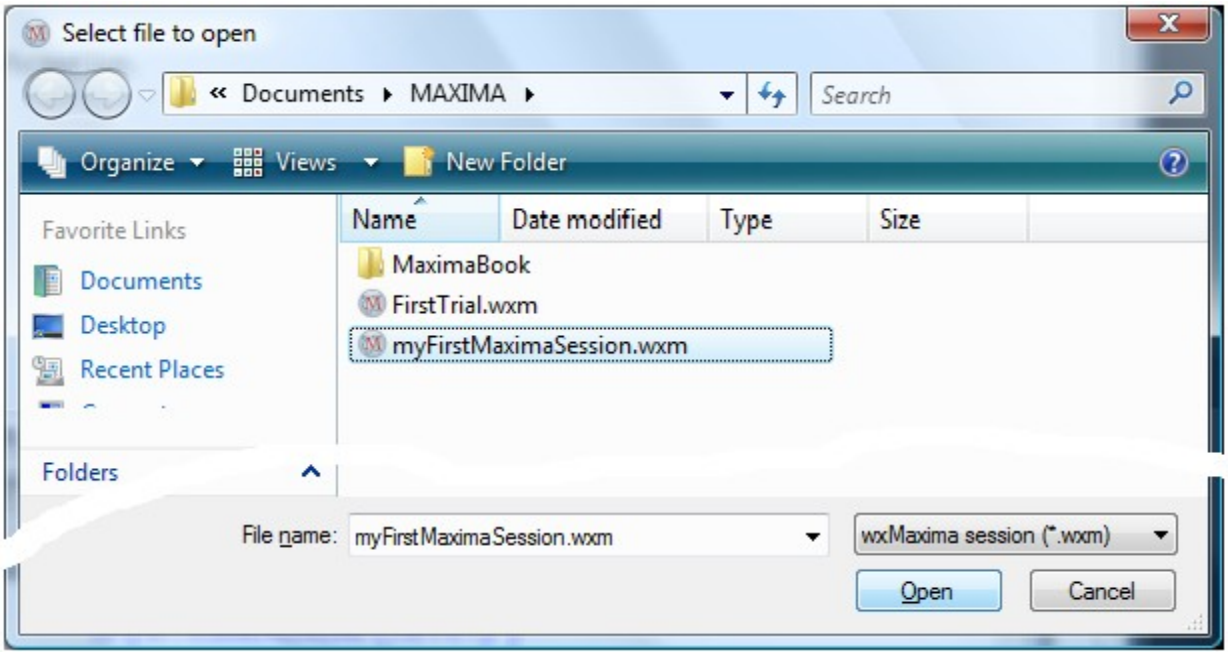
To save your session use the menu option *File* > *Save As* ... and give a name to the file into which you will save your session. The following dialog form was used in a *Windows Vista* environment to save the current session.



The file storing the session will be located in the folder `../Documents/MAXIMA/`, and will be named `myFirstMaximaSession.wxm`.

Reloading your session

Restart *Maxima* (*Edit* > *Restart Maxima*) and use the menu item *File* > *Open* to browse your computer file system. For example, in a *Windows Vista* environment, I located the file I want to load in the following dialog form:



In this case, *Maxima* opens the file and executes every command, stopping at input (%i15) where it asks again about the value of coefficient w in variable *myODE1*. Repeating the response *nonzero* allows *Maxima* to continue evaluating the file to recover the entire session saved.

Printing your session

To produce a hard-copy of your session use the menu item *File* > *Print*.

Loading a session without executing it

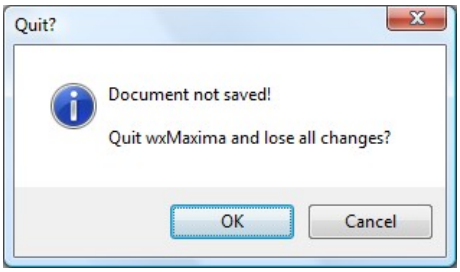
An alternative way to load a saved session is by using the menu item *File* > *Read File*. Using this option will list all the commands in the session without executing it. The commands will be available in the command history, and could be reactivated by using the up- and down-arrow keys, and pressing [ENTER] when the proper command is in the *INPUT* line.

Interrupting a calculation

If, for some reason, *wxMaxima* seems to be hung up in a calculation, you can interrupt the processing by using the menu item *Maxima* > *Interrupt*, or type *Cntl-G*. Alternatively, use the interrupt button in the menu bar:

Ending your session

To end your session use the menu item *File* > *Exit*, or click the [x] in the upper right corner of the *wxMaxima* window. This action produces the dialog form shown below.



Select [OK] if you don't want to save your current session. Otherwise, press [Cancel], save your session as indicated above, and exit *wxMaxima* once more.

Formatting your session

This section includes some examples of the use of text for commenting your session, as well as inserting sections and titles in your session.

Inserting text (comments) in *wxMaxima*

To enter text in *wxMaxima* use the menu item *Edit* > *Insert* > *Text*. The characters /* will be shown above the next input reference. Type one or more lines of text at the current cursor location. This line (or lines) of text can be used to comment your session. An example is shown next:

```
/*
This is an example of an integration:

(%i1)  integrate(x^3*exp(x),x);
(%o1)  (x^3 - 3 x^2 + 6 x - 6)%e^x
```

Text lines contained in saved session files get loaded with the rest of the commands when using *File* > *Open* or *File* > *Read file*.

Inserting a title or a section in *wxMaxima*

To insert a title use the menu item *Edit* > *Insert* > *Title*. This operation is similar to inserting text, except that the text is provided in a larger font.

To insert a section use the menu item *Edit* > *Insert* > *Section*. This operation is also similar to inserting text, except that the text is provided in a larger font and with an underline.

The following example shows a title and a section insertion in a *wxMaxima* session.

/*

This is a title

/*

This is a section

Inserting input

The menu item *Edit* > *Insert* > *Input* produces a prompt input as illustrated in the following example:

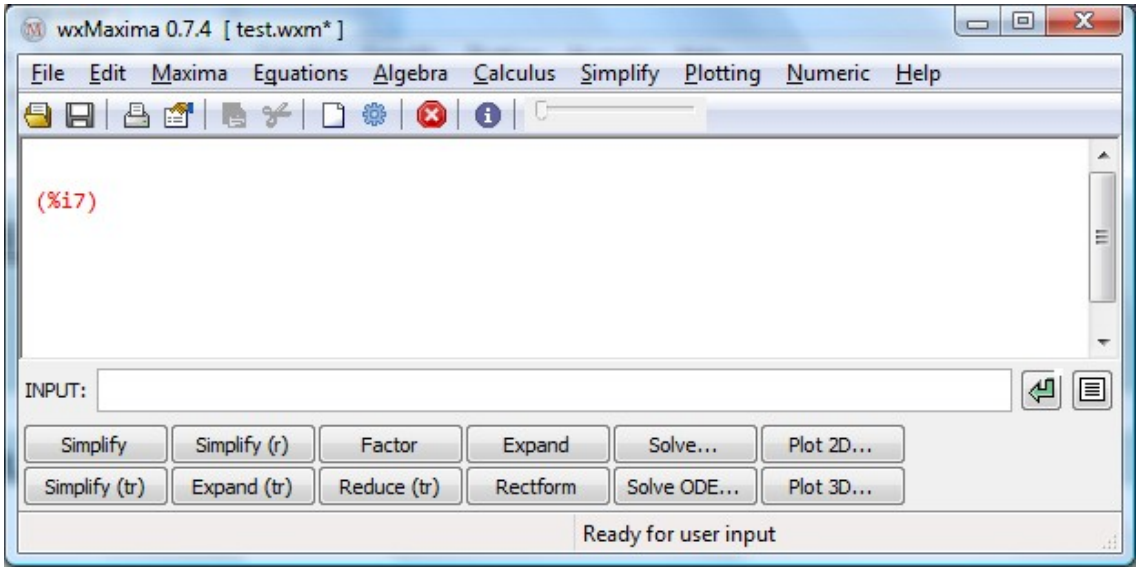
```
>> x:2$  y:x^2

(%i4)
```

If you enter a new command in the INPUT line, then the statement in front of the input prompt remains unevaluated. However, if you click on the input prompt statement, thus selecting it, and do a right-click, you can evaluate the command by selecting the option *Re-evaluate input*. In this case, the input gets evaluated as follows:

```
(%i4)  x:2$  y:x^2;
(%o5)  4
```

Clearing the screen

The option *Edit* > *Clear screen* clears the current *wxMaxima* screen, showing at the top of the screen the current input reference, e.g.,



Additional session management in *wxMaxima*

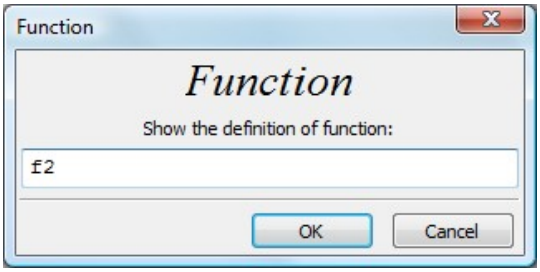
In this session we explore some of the menu items under the *Maxima* menu, namely:

- *Clear memory*: clears all variables user-defined functions - equivalent to `kill(all);`
- *Add to path*: allows user to select folders to add to the search path for *Maxima*
- *Show functions*: lists all user-defined functions in the current session (`functions;`)
- *Show definition*: provides a dialogue form to request function definitions in session
- *Show variables*: lists all variables active in the current session (`values;`)
- *Delete function*: delete selected user-defined function or functions
- *Delete variable*: delete selected variables

The following example shows the definition of variables and functions and the listing of their names:

```
(%i1)  x1 : 2$ x2 : -4$ y1 : -3$ y2 : 3.5$

(%i5)  f1(x) := x^3 + 1 $ f2(x) := 1/(1+x^2) $ g1(y) := sqrt(y+1)$
/*
Show functions:

(%i8)  functions;
(%o8)  [ f1(x), f2(x), g1(y)]
/*
Show variables

(%i9)  values;
(%o9)  [ x1 , x2 , y1 , y2 ]
```

The option *Show definition* is used next to find the definition of function *f2*:

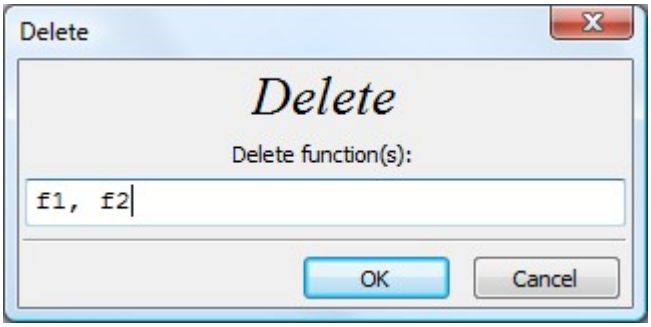


The result is shown below:

```
(%i10)  fundef(f2);
```

$$(\%o10)\quad f2(x) := \frac{1}{1+x^2}$$

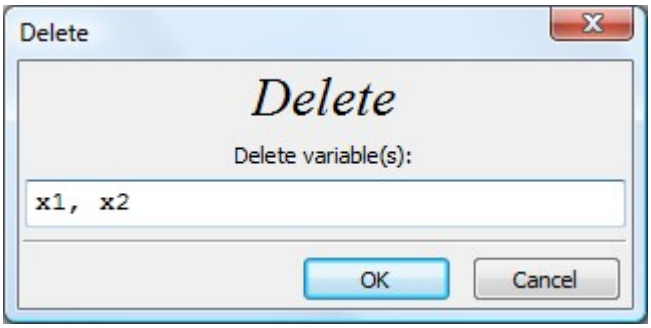
The option *Delete functions* produce the dialogue:



To delete functions *f1* and *f2* we enter those names in the dialogue. The result is shown below:

```
(%i11) remfunction(f1, f2);
(%o11) [ f1 , f2 ]
```

The option *Delete variables* produces the following dialogue:



To delete functions *x1* and *x2* we enter those names in the dialogue. The result is shown below:

```
(%i12) remvalue(x1, x2);
(%o12) [ x1 , x2 ]
```

An alternative way to delete user-defined functions or variables is to use function *kill*. This function basically clears any value or definition associated with a variable or function name. For example, to clear the contents of variable *y1*, use:

```
(%i13) kill(y1);
(%o13) done
```

Check that the value of *y1* is cleared:

```
(%i14) y1;
(%o14) y1
```

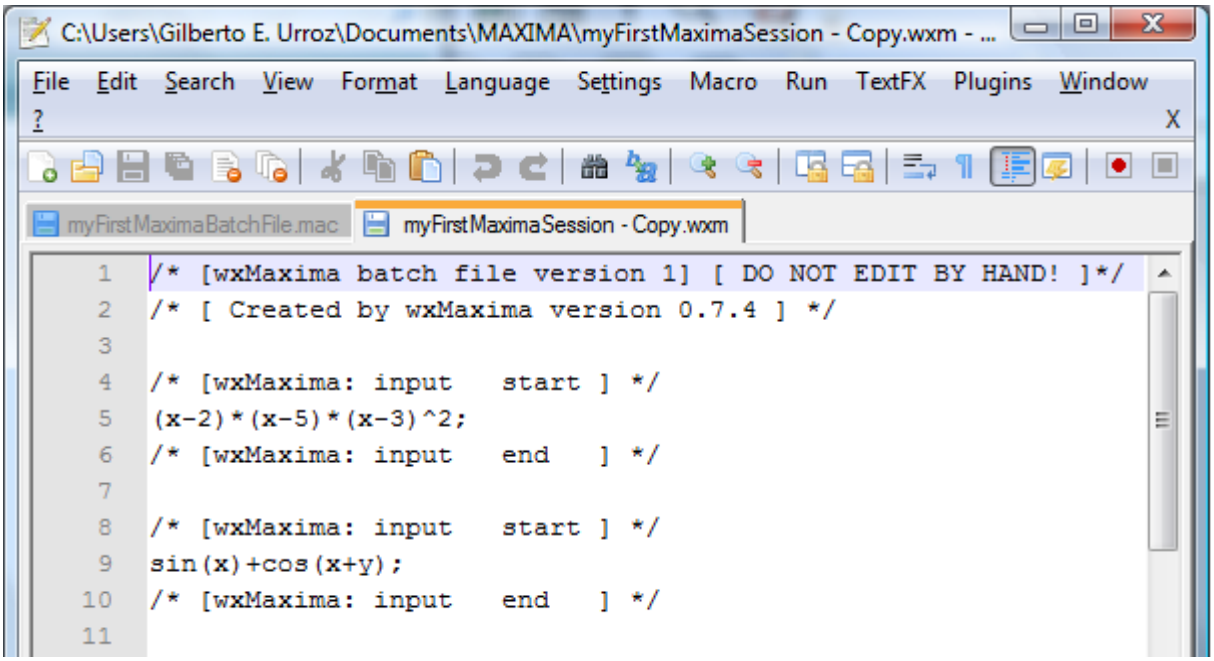
Creating a batch file

In an earlier exercise we saved a file called *myFirstMaximaSession.wxm*. In this section we will show you how to create a *Maxima* batch file out of your saved session.

In order to create a batch file we need to edit the session file using a text editor. In this example I will use the *Notepad++* text editor to open the session file. *Notepad++* is available at

http://notepad-plus.sourceforge.net/uk/site.htm .

When opened with *Notepad ++*, the file *myFirstMaximaSession.wxm* looks as follows:



Notice that you are warned in the very first line of the file to not edit the file by hand. This is for the *wmx* file. If you change anything in the file it may not be readable by *wxMaxima* again. The way to proceed is to save the file as a batch file, with the *.mac* suffix. Save it, for example, as *myFirstMaximaBatchFile.mac*, and edit it to look as shown below. This is the batch file that includes a number of comment lines (text between /* and */), and *Maxima* commands.

```
/* Batch file example */
/* Load a few expressions */
(x-2)*(x-5)*(x-3)^2;
sin(x)+cos(x+y);
(x+2)/((x-2)*(x+5));
/* Apply operations to those expression */
expand(%i1);
solve(%o1,x);
expand(%o3^2);
/* Define an expression as a variable */
a:(x-2)*(x-5)*(x-3)^2;
/* Apply operations to variable a */
expand(a);
solve(a,x);
/* Example of a differential equation solution */
myODE1 : 'diff(x(t),t,2) + omega^2*x(t) = F*sin(omega*t)$
mySOL1 : desolve(myODE1,x(t));
/* End of batch file */
```

To load a batch file use the menu item *File* > *Batch file*, and select the proper file to load. The result of the batch file operation will be shown in your *wxMaxima* window. Notice, however, that the comment lines are not shown in the *wxMaxima* window. If you want to show explanatory text from your batch file, you may want to replace the comments by a string, making sure that the string ends in a dollar sign ($) rather than in a semi-colon (;).

```
/* Batch file example */
"Load a few expressions"$
(x-2)*(x-5)*(x-3)^2 $
sin(x)+cos(x+y) $
(x+2)/((x-2)*(x+5)) $
"Apply operations to those expression"$
expand(%i1);
solve(%o1,x);
expand(%o3^2);
"Define an expression as a variable"$
a:(x-2)*(x-5)*(x-3)^2 $
"Apply operations to variable a "$
expand(a);
solve(a,x);
"Example of a differential equation solution "$
myODE1 : 'diff(x(t),t,2) + omega^2*x(t) = F*sin(omega*t)$
mySOL1 : desolve(myODE1,x(t));
/* End of batch file */
```

With theses changes, the output in the *wxMaxima* is now well documented, although the comment strings are now part of the input (with no output), rather than inserted text. Part of the output from the batch file is shown below:

```
(%i1) batch("C:/Users/Gilberto E. Urroz/Documents/MAXIMA/myFirstMaximaBatchFile.mac")$
batching #pC:/Users/Gilberto E. Urroz/Documents/MAXIMA/myFirstMaximaBatchFile.mac
(%i2) Load a few expressions
(%i3) (x - 2)(x - 5)(x - 3)^2
(%i4) cos(y + x)+ sin(x)
                2 + x
(%i5) ----------------------
      (x - 2)(5 + x)
(%i6) Apply operations to those expression
(%i7) expand(%i1)
(%o7) batch(C:/Users/Gilberto E. Urroz/Documents/MAXIMA/myFirstMaximaBatchFile.mac)
(%i8) solve(%o1, x)
(%o8) []
(%i9) expand(%o3^2)
(%o9) x^8 - 26 x^7 + 291 x^6 - 1832 x^5 + 7099 x^4 - 17346 x^3 + 26109 x^2 - 22140 x + 8100
(%i10) Define an expression as a variable
(%i11) a:(x - 2)(x - 5)(x - 3)^2
```

A batch file can also be created from scratch. Simply type the *Maxima* commands in a text file and save it with the suffix *.mac*. Here is an example of a batch file created from scratch:

```
1  /* Example of a batch file written from scratch */
2  "Examples of integrals:"$
3  "1 - Indefinite integrals:"$
4  integrate(x^2*log(x),x);
5  integrate(sin(x)^2,x);
6  integrate(exp(-x)*sin(x),x);
7  "2 - Definite integrals:"$
8  integrate(x^2*log(x),x,1,5);
9  integrate(sin(x)^2,x,-%pi/6,+%pi/6);
10 integrate(exp(-x)*sin(x),x,0,%pi/8);
11 /* End of batch file */
```

Important basic functions
This section addresses a few basic functions and operators of general application in mathematical functions and that were not addressed in any of the previous sections.

Evaluation or not evaluation of an operation
In many of the examples presented above related to differential equations we use an apostrophe (') in front of the derivative operator, *diff*, in order to avoid its evaluation. To illustrate the difference between the entry 'diff and diff, see the following example:

```
(%i1) myODE1 : 'diff(x,t,2)+'diff(x,t)+x=exp(-t);
```

(%o1) $$\frac{d^2}{d\,t^2}x + \frac{d}{d\,t}x + x = \%e^{-t}$$

```
(%i2) myODE1 : diff(x,t,2)+diff(x,t)+x=exp(-t);
```

(%o2) $$x = \%e^{-t}$$

In the first expression, using 'diff(x,t,2) produces as output the derivative thus indicated. However, in the second expression, *Maxima* evaluates the required derivatives. Since function *x(t)* has not been defined, the derivatives in the second expression evaluate to zero, and the result is $x = e^{-t}$.

The following example shows an non-evaluated integral:

```
(%i3) 'integrate(exp(-x)*sin(x),x);
```

(%o3) $$\int \%e^{-x}\sin(x)\,dx$$

An example of a summation is shown next:

```
(%i27) 'sum(1/k^2, k, 1, inf);
```

(%o27) $$\sum_{k=1}^{\infty} \frac{1}{k^2}$$

Applications of *ev*
Function *ev* evaluates an expression in a given environment determined by a number of arguments. For complete information on function *ev,* use the menu item *Help* > *Describe*, and enter the name *ev* in the dialogue form. In this document we will present only some specific examples of the use of function *ev.*

- Substituting constants in an equation before solving it:

```
(%i35)  ev(solve(a*x^2+b*x+c=0),a=2,b=-5,c=3);
```

$$(\%o35)\quad \left[x=\frac{3}{2},\ x=1\right]$$

- Force floating-point evaluation of rational numbers:

```
(%i41)  ev(sqrt(2),float);
(%o41)  1.414213562373095

(%i42)  ev(3^(1/3),float);
(%o42)  1.442249570307408
```

- Force derivative calculation after result has been suppressed:

```
(%i47)  'diff(x^3+x,x);
```

$$(\%o47)\quad \frac{d}{dx}\left(x^3+x\right)$$

```
(%i48)  ev(%,diff);
```

$$(\%o48)\quad 3x^2+1$$

- Derivative and integral calculations can be forced with the option *nouns*:

```
(%i55)  'integrate(x*log(x),x);ev(%,nouns);
```

$$(\%o55)\quad \int x\log(x)\,dx$$

$$(\%o56)\quad \frac{x^2\log(x)}{2}-\frac{x^2}{4}$$

```
(%i57)  'diff(x*log(x),x);ev(%,nouns);
```

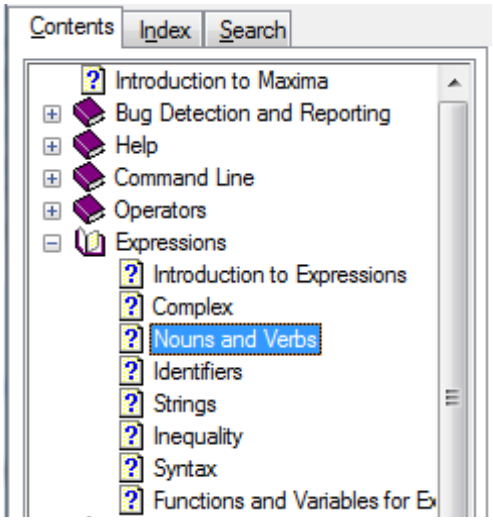
$$(\%o57)\quad \frac{d}{dx}\left(x\log(x)\right)$$

$$(\%o58)\quad \log(x)+1$$

These examples illustrates how to list an expression and their evaluation in the same line. It also introduces the idea of *nouns* in *Maxima* evaluation.

Nouns and verbs

To understand the use of the argument *nouns* in the examples above, please open the *Maxima Manual*, available through the menu item *Help* > *Maxima help*, and find section *6.3 Nouns and verbs* in the *Contents* tab, as shown in the figure below.



Read this section in the *Manual* to understand the idea of verbs and nouns, as well how to convert form one to the other.

Online help

In an earlier section we presented the different options available in the *Help* menu. A quick way to obtain help is by using the *??* operator. For example, if you are interested in finding information about the function *eval*, use:

```
(%i31)  ?? eval;
 0: Assignment operator (evaluates left-hand side)  (General operators)
 1: eval  (General operators)
 2: eval_string  (Functions and Variables for strings)
 3: infeval  (Functions and Variables for Command Line)
 4: lbfgs_nfeval_max  (Functions and Variables for lbfgs)
 5: noeval  (Functions and Variables for Simplification)
 6: timer_devalue  (Functions and Variables for Debugging)
 7: tr_warn_meval  (Functions and Variables for Function Definition)
Enter space-separated numbers, `all' or `none':
```

Maxima reply by listing a number of entries that include the particle *eval*, and requesting additional input from the user. At this point, the user can enter a particular number referring to the 7 options listed, or enter the particles *all* or *none*. Enter *none* to stop the online help process.

The following is another example related to the function *integrate*.

```
(%i33)  ?? integrate;
 0: integrate  (Functions and Variables for Integration)
 1: integrate_use_rootsof  (Functions and Variables for Integration)
Enter space-separated numbers, `all' or `none':  none;
(%o33)  true
```