Maxiplot: Maxima and Gnuplot in LaTeX.

September 21, 2013

1 Introduction

For those who do not know *Maxima*, it is a symbolic calculation program which can be used to compute derivatives and integrals, solve equations, find limits, work with vectors and matrices and create graphics, among many other things. It also adds the possibility to write programs, thus expanding its capabilities. As if all this was not enough, it is also released under the GNU General Public License and it can be downloaded for free at `http://maxima.sourceforge.net`, where there is also documentation in several languages (including Spanish).

The purpose of this LaTeX package is to provide "programming" capabilities importing the results, without the need of working with various files and interfaces. Maxima code can be included within the LaTeX document. When the document is processed, a file with extension `.mac` is generated, which can be directly processed by Maxima, creating another file with extension `.mxp`; when the LaTeX document is processed again, that file will be automatically inserted.

Gnuplot commands can also be inserted, thanks to some additional commands added by J. M. Mira. Thus, in addition to the auxiliary files already mentioned, another file with extension `.gnp` will be created, which after being processed by *Gnuplot* can be added to the document.

2 Installation

Just copy the file `maxiplot.sty` into a place where LaTeX can find it, or copy it into the same directory where you have your document. For those who have used previous versions of `maxiplot`, notice that in this version *no other files are needed.*

3 LaTeX package `maxiplot`

3.1 How is it used?

Its usage is simple. Process your document as normally done; for instance, write in the command line:

```
latex mydocument.tex
```

You will find out that a file `mydocument.mac` has been created in your working directory. Process that file with Maxima:

```
maxima -b mydocument.mac
```

And if you have used *Gnuplot* commands in your document:

```
gnuplot mydocument.gnp
```

Process your LaTeX document again, *et voilà!*.

If your distribution allows it, you can enable the command `write18` to have *Maxima* and *Gnuplot* automatically run when you process your LaTeX document (you should previously add your installation directory to the executable search path in you operating system).

3.2 User interface

3.2.1 Maxima.

This section and the following ones show some examples of the use of package `maxiplot`. It would be convenient to have some basic knowledge of Maxima to follow the examples.

This package has an option (for the time being) to allow compatibility with the `pmatrix` environment of the `amsmath` package. Therefore, if you are going to create matrices with that environment, you should add the following lines to the document preamble

```
\usepackage{amsmath}
\usepackage[amsmath]{maxiplot}
```

The most important environments are `maxima` and `maximacmd`. The contents of those environments will be passed to a file with extension `.mac` to be processed later on with Maxima. Hence, there can be no LaTeX-style comments within those environments; namely, % cannot be used to start a comment since that symbol has a special meaning in Maxima. Instead, comments within those environment should follow the C language syntax (/* *comment* */). Commands will be inserted as arguments for a function; therefore, they must be separated by commas.

Let us start with a simple example:

```
\[   %Math mode begin
\begin{maxima}
  f: x/(x^3-3*x+2),     /* Integrating it */
  tex('integrate(f,x)), /* will show its integral... */
  print("="),
  tex(integrate(f,x)),  /* ...and the result */
  print("+K")
\end{maxima}
\]   %Math mode end
```

In the place with this code is found, the result will be:

$$\int \frac{x}{x^3-3\,x+2}\,dx = -\frac{2\,\log\left(x+2\right)}{9} - \frac{1}{3\,x-3} + \frac{2\,\log\left(x-1\right)}{9} + K$$

There are some environments where a `maxima` block cannot be included. In those cases the `maxima*` variant can be used, which gives output immediately. That output can be then inserted with the command `\maximacurrent`, as in the following example:

```
\begin{maxima*}
  suml(L):=lsum(i,i,L),
  printrow(L):=block(
    [str:""],
    for i:1 step 1 thru length(L)-1 do(
        str:concat(str,L[i],"&")),
    str:concat(str,L[length(L)],"\\\\"),
    print(str)),
  xi:[1,2,3,4,5,6],
  fi:[3,4,7,10,8,2],
  for i:1 while i<=length(xi) do (
    printrow([xi[i],fi[i],(fi*xi)[i],(fi*xi^2)[i]])
    ),
  print("\\hline"),
  printrow(["",N:suml(fi),fx:suml(fi*xi),fx2:suml(fi*xi^2)])
\end{maxima*}

\begin{center}
  \begin{tabular}{|c|c|c|c|c|}
  $x_i$&$n_i$&$n_i\cdot x_i$&$n_i\cdot x_i^2$\\
  \hline
  \maximacurrent
```

```
  \end{tabular}
\end{center}
```

x_i	n_i	$n_i \cdot x_i$	$n_i \cdot x_i^2$
1	3	3	3
2	4	8	16
3	7	21	63
4	10	40	160
5	8	40	200
6	2	12	72
	34	124	514

It is important to keep in mind that the `\maximacurrent` command will be replaced by the result of *the last* ***maxima*** *block*, so it must be used before any other `maxima` blocks.

If you would like to use that result later on, or if you are going to use it at several places in the document, you can add an optional command with a variable name that will save that content. The previous example might as well had been implemented in the following way:

```
\begin{maxima*}[table]
  suml(L):=lsum(i,i,L),
  printrow(L):=block(
    [str:""],
    for i:1 step 1 thru length(L)-1 do(
        str:concat(str,L[i],"&")),
    str:concat(str,L[length(L)],"\\\\"),
    print(str)),
  xi:[1,2,3,4,5,6],
  fi:[3,4,7,10,8,2],
  for i:1 while i<=length(xi) do (
    printrow([xi[i],fi[i],(fi*xi)[i],(fi*xi^2)[i]])
    ),
  print("\\hline"),
  printrow(["",N:suml(fi),fx:suml(fi*xi),fx2:suml(fi*xi^2)])
\end{maxima*}

\begin{center}
  \begin{tabular}{|c|c|c|c|c|}
  $x_i$&$n_i$&$n_i\cdot x_i$&$n_i\cdot x_i^2$\\
  \hline
  \table
  \end{tabular}
\end{center}
```

Notice that when passing "`table`" as a parameter to `maxima*` *there is no need to use a backslash* (\).

There is a line-mode version of the `maxima` environment, with similar usage and options: the command `\imaxima` (from "inline maxima").

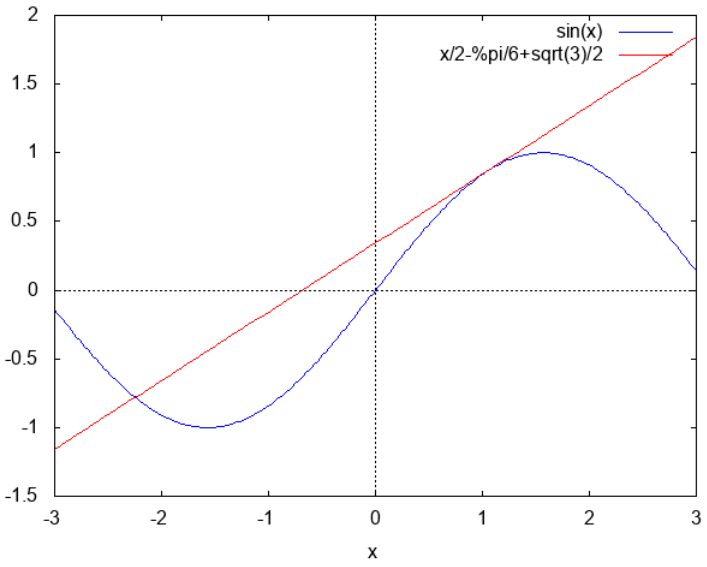
```
\[
\overline{x}=\imaxima{tex(xx:fx/N)}\qquad
\sigma^2=\imaxima{tex(sx2:fx2/N-xx^2)}\qquad
\sigma=\imaxima{tex(sqrt(sx2))}
\]
```

$$\overline{x} = \frac{62}{17} \qquad \sigma^2 = \frac{525}{289} \qquad \sigma = \frac{5\sqrt{21}}{17}$$

In cases when no output is expected, such as defining a function or loading *Maxima* packages, the `maximacmd` environment or `\imaximacmd` command should be used. These two do not have `*` variant nor any options. Furthermore, the *Maxima* commands inside these must be separated by a semicolon (;) or, better yet, a dollar sign ($). As an example, let us look at some of the features of the *Maxima*/*Gnuplot* interface. This example shows the plots of the sin function and its tangent at $\frac{\pi}{3}$:

```
\begin{maximacmd}
  tangent(fx,a):=expand(ev(fx,x=a)
                  +subst(a,x,diff(fx,x))*(x-a))$
  plot2d([sin(x),tangent(sin(x),%pi/3)], [x,-3,3],
         [gnuplot_preamble,"set zeroaxis;"],
         [gnuplot_term, png],
         [gnuplot_out_file,"./\jobname2D.png"])$
\end{maximacmd}
\begin{center}
  \mxpIncludegraphics[scale=0.60]{\jobname2D.png}
\end{center}
```

This code creates a `png` format file `maxiplot_en2D.png`:



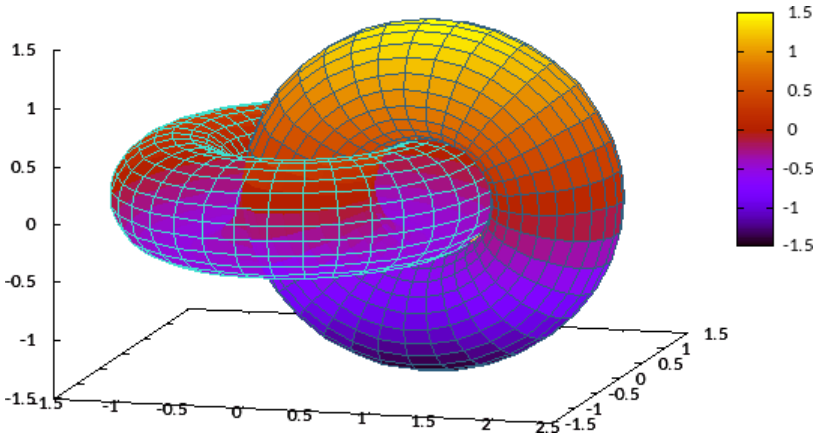
The environments introduced so far can contain LaTeX commands which will be replaced before passing them to the `mac` file. Sometimes this feature might not be desired and could lead to problems with certain strings. The environments `vmaxima` and `vmaximacmd` solve this problem; their usage is similar to the previous environments, but their content is passed literally. These environments are based on the `verbatim` LaTeX package.

3.2.2 Gnuplot

While *Maxima* can create graphics via *Gnuplot*, sometimes it might be preferable to work directly with this last program. In order to do that, the environments `gnuplot` and its verbatim version `vgnuplot` are used.

Here is a 3D example

```
\begin{gnuplot}
  set term png crop enhanced font "calibri, 10"
  set output "toros.png"
  set parametric
  set urange [0:2*pi]
  set vrange [-pi:pi]
  set isosamples 36,24
  set hidden3d
  set view 75,15,1,1
  unset key
  set ticslevel 0
  x1(u,v)=cos(u)+.5*cos(u)*cos(v)
  y1(u,v)=sin(u)+.5*sin(u)*cos(v)
  z1(u,v)=.5*sin(v)
  x2(u,v)=1+cos(u)+.5*cos(u)*cos(v)
  y2(u,v)=.5*sin(v)
  z2(u,v)=sin(u)+.5*sin(u)*cos(v)
  set multiplot
  splot x1(u,v), y1(u,v), z1(u,v) w pm3d, x2(u,v), y2(u,v), z2(u,v) w pm3d
  splot x1(u,v), y1(u,v), z1(u,v) lt 3,   x2(u,v), y2(u,v), z2(u,v) lt 5
\end{gnuplot}
\begin{center}
  \mxpIncludegraphics[scale=0.75]{toros.png}
\end{center}
```



Let us examine the `\mxpIncludegraphics` command: its usage is the same as `includegraphics` from package `graphicx`; in fact, it just makes sure that the graphic file exists before invoking that macro.

3.3 Problems

This is an experimental version; many of the capabilities of Maxima have not been tested and it has not been tried with the most important LaTeX packages. Thus, it will surely need some tweaking.

However, I think that most of the problems will appear when showing certain outputs. For instance, if the result of a computation is too long, it will not be easy to break it into several lines (except if one works in Maxima and then copies the result to the document, of course).

Other possible problems can be addressed from within the LaTeX document. By default, Maxima orders expressions by inverse alphabetical order; hence, if we type:

```
$$\imaxima{tex(x+y+z+t=0)}$$
```

we get:

$$z + y + x + t = 0$$

That can be avoided by using Maxima functions `ordergreat` and `unorder`:

```
\imaximacmd{ordergreat(x,y,z,t)$}
$$\imaxima{tex(x+y+z+t=0)}$$
\imaximacmd{unorder()$}
```

Furthermore, if we would like to align several equations, we will need to dive a little deeper:

```
\begin{maximacmd}
  ordergreat(x,y,z)$
  :lisp(defprop mequal (&=) texsym)
\end{maximacmd}

\begin{maxima*}
  eq1:a-2*b=x+y,
  eq2:b=2*x-3*y+2*z,
  tex(eq1),
  print("\\\\"),
  tex(eq2)
\end{maxima*}

\begin{maximacmd}
  unorder()$
  :lisp(defprop mequal (=) texsym)
\end{maximacmd}
```

$$a - 2\,b = x + y \tag{1}$$
$$b = 2\,x - 3\,y + 2\,z \tag{2}$$

4 A few last words

As I mentioned before, this is an experimental package that will probably need some amendments and additions, so any ideas or comments will be welcome.

José Miguel M. Planas
<nohaim@gmail.com>

(English translation by Jaime Villate)