

# Maxiplot: Maxima y Gnuplot en L<sup>A</sup>T<sub>E</sub>X.

29 de noviembre de 2014

## 1. Introducción.

Para los que aun no conocen *Maxima*, se trata de un programa de cálculo simbólico que permite, entre otras cosas, calcular derivadas, integrales, resolver ecuaciones y límites, manejo de vectores y matrices, generar gráficos... Además es posible crear programas, pudiendo así ampliar sus capacidades. Por si esto fuese poco, está bajo licencia GNU, y se puede descargar gratuitamente desde <http://maxima.sourceforge.net>, donde también se puede encontrar toda la documentación en varios idiomas (incluyendo el español).

El propósito de este paquete L<sup>A</sup>T<sub>E</sub>X es, precisamente, permitir ‘programar’ e importar los resultados sin necesidad de trabajar con varios ficheros y entornos. Dentro del documento L<sup>A</sup>T<sub>E</sub>X se podrá incluir código en Maxima; al compilar el documento se genera un fichero (con extensión `.mac`) directamente procesable por Maxima, que a su vez genera otro fichero (con extensión `.mxp`) que, al recompilar el documento L<sup>A</sup>T<sub>E</sub>X, será automáticamente insertado.

De igual forma se pueden insertar comandos *Gnuplot* gracias a los comandos añadidos por J. M. Mira. Así, además de los ficheros anteriores, se generará un fichero con la extensión `.gnp` que, tras ser procesados por *Gnuplot* podrán agregarse a su documento.

## 2. Instalación.

Simplemente coloque el archivo `maxiplot.sty` en una ruta conocida para L<sup>A</sup>T<sub>E</sub>X o en el mismo directorio de su documento. Para los que hayan usado las primeras versiones del paquete `maxima`, decirles que *ya no es necesario ningún otro fichero*.

### 3. El paquete maxiplot para L<sup>A</sup>T<sub>E</sub>X.

#### 3.1. ¿Cómo se usa?

Sencillo. Compile normalmente su documento, por ejemplo, desde la línea de comandos:

```
latex midocumento.tex
```

Encontrará entonces en su directorio de trabajo que se ha generado un archivo `midocumento.mac`. Procese este documento con Maxima:

```
maxima -b midocumento.mac
```

Y, si contiene comandos *Gnuplot*:

```
gnuplot midocumento.gnp
```

Vuelva ahora a compilar su documento L<sup>A</sup>T<sub>E</sub>X *et voilà!*.

Si su distribución lo permite puede activar el comando `write18` que permitirá que *Maxima* y *Gnuplot* se ejecuten automáticamente al compilar su documento L<sup>A</sup>T<sub>E</sub>X (previamente deberá añadir el directorio de su instalación a la ruta de búsqueda de su sistema operativo).

#### 3.2. La interfaz de usuario.

##### 3.2.1. Maxima.

En esta sección y las siguientes se presentan algunos ejemplos de uso del paquete `maxiplot`. Es conveniente que el usuario tenga algún conocimiento básico sobre Maxima.

Este paquete admite (por el momento) una opción, que permite la compatibilidad con el entorno `pmatrix` del paquete `amsmath`. Así, si va a usar matrices en dicho entorno deberá especificar

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

Los entornos más importantes son `maxima` y `maximacmd`. El contenido de estos entornos será copiado a un archivo con extensión `.mac` para ser procesado después por Maxima. Así, no podrá contener comentarios al estilo L<sup>A</sup>T<sub>E</sub>X (esto es, comenzando por `%`), puesto que este símbolo es utilizado por Maxima, sino que se insertan como en C (`/* comentario */`). Los comandos serán insertados como argumentos de una función, así que deberán ir separados por comas.

Comencemos por un ejemplo sencillo:

```
\[ %Comienzo modo matem\'aticas
\begin{maxima}
  f: x/(x^3-3*x+2),      /* Integrando */
  tex('integrate(f,x)), /* Presenta la integral... */
  print("="),
  tex(integrate(f,x)), /* ...y el resultado */
  print("+K")
\end{maxima}
\[ %F\'in modo matem\'aticas
```

En el lugar donde coloquemos este código obtendremos:

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

Hay entornos dentro de los cuales no podrá incluir un bloque `maxima`. Para estos casos puede usar la version `maxima*`, el cual no produce salida inmediata. Posteriormente podrá insertar dicha salida con el comando `\maximacurrent`:

```
\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{center}
```

```
\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

Es importante tener en cuenta que el comando `\maximacurrent` será sustituido por el resultado *del último bloque maxima*, así que deberá ser usado *antes* del siguiente bloque. Si quiere usar dicha salida posteriormente, o si la va a insertar varias veces en su documento, puede usar como parámetro opcional el nombre de un comando que almacenará su contenido. En tal caso podríamos implementar el ejemplo anterior como:

```
\begin{maxima*}[tabla]
  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
  \end{tabular}
\end{center}
```

Observemos que al pasar ‘tabla’ como parámetro *no debe de usarse la barra invertida* (\).

Existe una versión ‘en línea’ del entorno `maxima`, cuyo uso y opciones es similar: el comando `\imaxima` (de ‘inline maxima’).

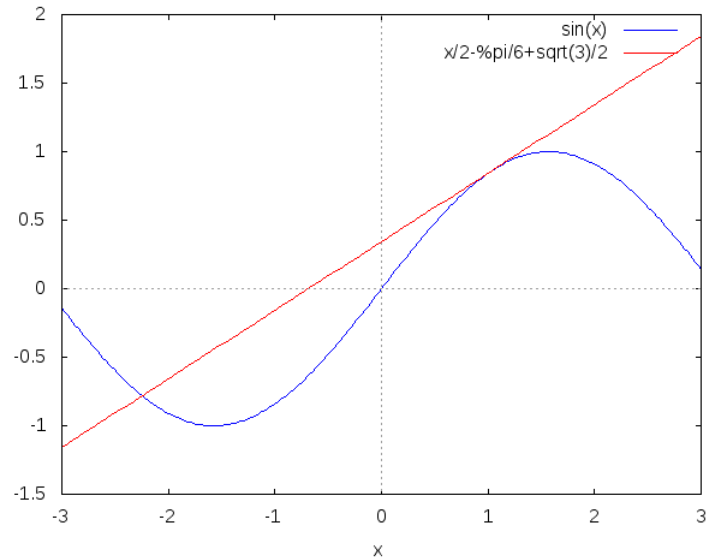
```
\[
\overline{x}=\imaxima{tex(xx:fx/N)}\qquad
\sigma^2=\imaxima{tex(sx2:fx2/N-xx^2)}\qquad
\sigma=\imaxima{tex(sqrt(sx2))}
\]
```

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

Para los casos en que no se espere salida alguna, sino que sólo se pretenda definir funciones, o cargar paquetes de *Maxima*, tenemos el entorno `maximacmd` y su correspondiente comando `\imaximacmd`. Éstos no admiten la versión `*` ni parámetro alguno, ya que no son necesarios. Además los comandos *Maxima* usados aquí deberán ir separados por ‘punto y coma’ (;) o, mejor por el símbolo ‘dólar’ (\$). Como ejemplo, veamos algunas capacidades de la interfaz *Maxima/Gnuplot*. En este ejemplo vemos las gráficas de la función *seno* y su tangente en  $\frac{\pi}{3}$ :

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

Este código genera el archivo en formato png `maxiplot2D.png`:



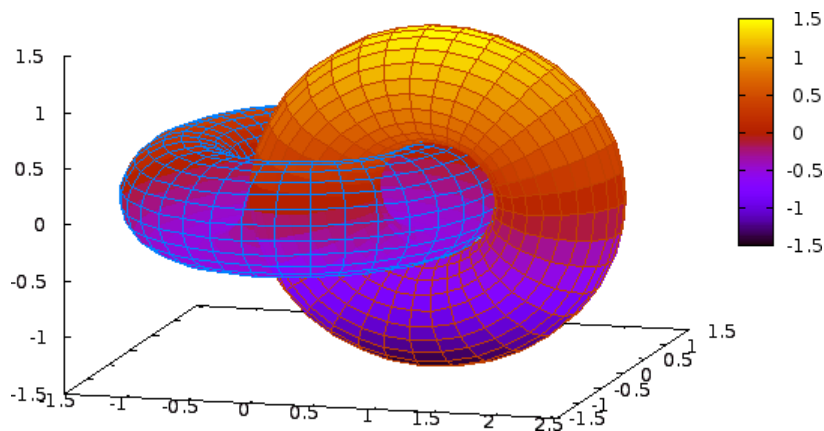
Los entornos presentados hasta ahora pueden contener comandos  $\text{\LaTeX}$  que serán sustituidos antes de pasar al archivo `mac`. Ahora bien, esta capacidad puede, a veces, no ser deseable o puede dar problemas con algunas secuencias de caracteres. Para evitar esto están los entornos `vmaxima` y `vmaximacmd`, cuyo uso es similar a los anteriores, pero el volcado es literal. Estos entornos están basados en el paquete `verbatim` de  $\text{\LaTeX}$ .

### 3.2.2. Gnuplot.

Si bien *Maxima* permite la inclusión de gráficos a través de *Gnuplot*, a veces puede ser preferible trabajar directamente con éste último. Para esto usaremos los entornos *gnuplot* y su versión ‘verbatim’ *vgnuplot*.

Un ejemplo 3D

```
\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}
  \mxiIncludegraphics[scale=0.75]{toros.png}
\end{center}
```



Observemos el comando `\mxpIncludegraphics`: el uso es el mismo que `includegraphics` del paquete `graphicx`, de hecho lo que hace es asegurarse de que existe el archivo gráfico e invocar dicha macro.

### 3.3. Problemas.

Esta es aún una versión en pruebas, y todavía no han sido probadas muchas de las capacidades de Maxima, ni se ha comprobado con los paquetes más importantes de  $\text{\LaTeX}$ , así que seguramente necesite unos cuantos retoques.

Aunque a mi parecer los mayores problemas aparezcan a la hora de presentar algunas salidas. Por ejemplo, si el resultado de una operación es muy largo no será fácil dividir en varias líneas (salvo, claro está, trabajando en Maxima y luego copiando en el documento).

Otros ‘problemas’ sí que se pueden atajar desde  $\text{\LaTeX}$ . Por defecto Maxima ordena las expresiones en orden alfabético inverso, así si escribimos:

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

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

Para evitarlo podemos usar las funciones Maxima `ordergreat` y `unorder`:

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

Si además queremos alinear varias ecuaciones, tendremos que meternos un poco más en profundidad:

```
\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}
```



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

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

#### **4. Para terminar:**

Como he dicho antes, este es un paquete en pruebas y muy posiblemente necesite correcciones y adiciones, así que cualquier idea o comentario será bienvenido.

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