

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

21 septembre 2013

## 1 Introduction

Pour ceux qui ne connaissent pas *Maxima*, il s'agit d'un programme de calcul formel qui peut être utilisé pour calculer des dérivées et des intégrales, résoudre des équations, trouver des limites, travailler avec des vecteurs et des matrices et créer des graphiques, entre autres choses. Il offre également la possibilité d'écrire des programmes, étendant ainsi ses capacités. Comme si tout cela ne suffisait pas, il est également publié sous la licence publique générale GNU et peut être téléchargé gratuitement sur <http://maxima.sourceforge.net>, où se trouve également une documentation en plusieurs langues (dont l'espagnol).

L'objectif de ce package L<sup>A</sup>T<sub>E</sub>X est de fournir des capacités de « programmation » en important les résultats, sans avoir besoin de travailler avec plusieurs fichiers et interfaces. Le code Maxima peut être inclus directement dans le document L<sup>A</sup>T<sub>E</sub>X. Lorsque le document est traité, un fichier avec l'extension `.mac` est généré, lequel peut être directement traité par Maxima pour créer un autre fichier avec l'extension `.mxp` ; lorsque le document L<sup>A</sup>T<sub>E</sub>X est traité à nouveau, ce fichier sera automatiquement inséré.

Des commandes *Gnuplot* peuvent également être insérées, grâce à des commandes supplémentaires ajoutées par J. M. Mira. Ainsi, en plus des fichiers auxiliaires déjà mentionnés, un autre fichier avec l'extension `.gnp` sera créé, lequel, après avoir été traité par *Gnuplot*, pourra être ajouté au document.

## 2 Installation

Il suffit de copier le fichier `maxiplot.sty` dans un emplacement où L<sup>A</sup>T<sub>E</sub>X peut le trouver, ou de le copier dans le répertoire où se trouve votre document. Pour ceux qui ont utilisé des versions précédentes de `maxiplot`, notez que dans cette version, *aucun autre fichier n'est nécessaire*.

## 3 Le package L<sup>A</sup>T<sub>E</sub>X maxiplot

### 3.1 Comment l'utiliser ?

Son utilisation est simple. Traitez votre document comme d'habitude ; par exemple, tapez dans la ligne de commande :

```
latex mondocument.tex
```

Vous constaterez qu'un fichier `mondocument.mac` a été créé dans votre répertoire de travail. Traitez ce fichier avec Maxima :

```
maxima -b mondocument.mac
```

Et si vous avez utilisé des commandes *Gnuplot* dans votre document :

```
gnuplot mondocument.gnp
```

Traitez à nouveau votre document L<sup>A</sup>T<sub>E</sub>X, *et voilà !*. Si votre distribution le permet, vous pouvez activer la commande `write18` pour que *Maxima* et *Gnuplot* soient exécutés automatiquement lors du traitement de votre document L<sup>A</sup>T<sub>E</sub>X (vous devrez préalablement ajouter votre répertoire d'installation au chemin de recherche des exécutables de votre système d'exploitation).

### 3.2 Interface utilisateur

#### 3.2.1 Maxima.

Cette section et les suivantes présentent quelques exemples d'utilisation du package `maxiplot`. Il serait pratique d'avoir quelques connaissances de base de Maxima pour suivre les exemples. Ce package possède une option (pour le moment) pour permettre la compatibilité avec l'environnement `pmatrix` du package `amsmath`. Par conséquent, si vous allez créer des matrices avec cet environnement, vous devriez ajouter les lignes suivantes au préambule du document :

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

Les environnements les plus importants sont `maxima` et `maximacmd`. Le contenu de ces environnements sera transmis à un fichier avec l'extension `.mac` pour être traité ultérieurement avec Maxima. Par conséquent, il ne peut y avoir de commentaires de style L<sup>A</sup>T<sub>E</sub>X à l'intérieur de ces environnements ; à savoir, le symbole `%` ne peut pas être utilisé pour commencer un commentaire car ce symbole a une signification spéciale dans Maxima. Au lieu de cela, les commentaires à l'intérieur de ces environnements doivent suivre la syntaxe du langage C (*/\* commentaire \*/*). Les commandes seront insérées comme arguments d'une fonction ; elles doivent donc être séparées par des virgules.

Commençons par un exemple simple :

```
\[ %Début du mode mathématique
\begin{maxima}
  f: x/(x^3-3*x+2),      /* Intégration */
  tex('integrate(f,x)), /* affichera l'intégrale... */
  print("="),
  tex(integrate(f,x)), /* ...et le résultat */
  print("+K")
\end{maxima}
\] %Fin du mode mathématique
```

À l'endroit où ce code se trouve, le résultat sera :

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

Il existe certains environnements où un bloc `maxima` ne peut pas être inclus. Dans ces cas, la variante `maxima*` peut être utilisée, laquelle donne une sortie immédiate. Cette sortie peut ensuite être insérée avec la commande `\maximacurrent`, comme dans l'exemple suivant :

```
\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

Il est important de garder à l'esprit que la commande `\maximacurrent` sera remplacée par le résultat du *dernier bloc maxima*, elle doit donc être utilisée avant tout autre bloc `maxima`. Si vous souhaitez utiliser ce résultat plus tard, ou si vous allez l'utiliser à plusieurs endroits du document, vous pouvez ajouter une commande optionnelle avec un nom de variable qui sauvegardera ce contenu. L'exemple précédent aurait pu être implémenté de la manière suivante :

```
\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}
```

Remarquez que lors du passage de « `table` » comme paramètre à `maxima*`, *il n'est pas nécessaire d'utiliser une barre oblique inverse* (`\`). Il existe une version en mode ligne de l'environnement `maxima`, avec une utilisation et des options similaires : la commande `\imaxima` (pour « inline maxima »).

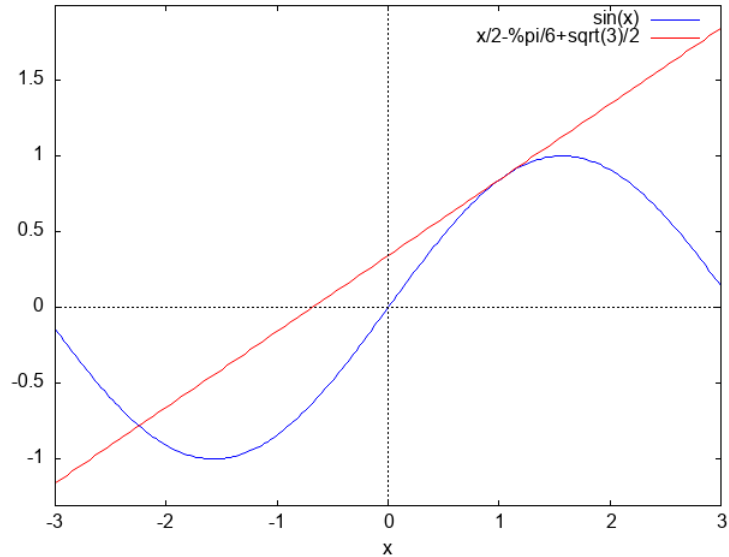
```
\[
\overline{x}=\imaxima{tex(xx:fx/N)}\qquad
\sigma^2=\imaxima{tex(sx2:fx2/N-xx^2)}\qquad
\sigma=\imaxima{tex(sqrt(sx2))}
\]
```

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

Dans les cas où aucune sortie n'est attendue, comme la définition d'une fonction ou le chargement de packages *Maxima*, l'environnement `maximacmd` ou la commande `\imaximacmd` doivent être utilisés. Ces deux-là n'ont pas de variante `*` ni d'options. De plus, les commandes *Maxima* à l'intérieur de ceux-ci doivent être séparées par un point-virgule (;) ou, mieux encore, un signe dollar (\$). À titre d'exemple, examinons quelques fonctionnalités de l'interface *Maxima/Gnuplot*. Cet exemple montre les tracés de la fonction `sin` et de sa tangente en  $\frac{\pi}{3}$  :

```
\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}
  \mxiIncludegraphics[scale=0.60]{\jobname2D.png}
\end{center}
```

Ce code crée un fichier au format png nommé `maxiplot_fr2D.png` :

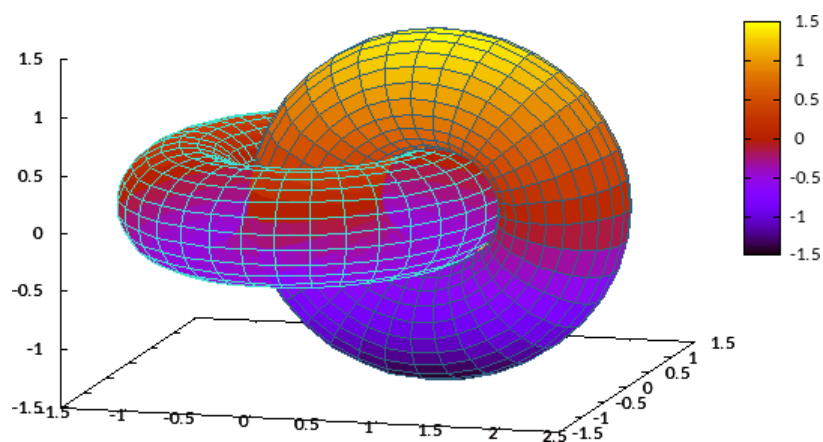


Les environnements introduits jusqu'à présent peuvent contenir des commandes  $\text{\LaTeX}$  qui seront remplacées avant de les transmettre au fichier `mac`. Parfois, cette fonctionnalité peut ne pas être souhaitée et pourrait entraîner des problèmes avec certaines chaînes de caractères. Les environnements `vmaxima` et `vmaximacmd` résolvent ce problème ; leur utilisation est similaire aux environnements précédents, mais leur contenu est transmis littéralement. Ces environnements sont basés sur le package  $\text{\LaTeX}$  `verbatim`.

### 3.2.2 Gnuplot

Bien que *Maxima* puisse créer des graphiques via *Gnuplot*, il est parfois préférable de travailler directement avec ce dernier programme. Pour ce faire, les environnements *gnuplot* et sa version verbatim *vgnuplot* sont utilisés. Voici un exemple en 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}
```



Examinons la commande `\mxpIncludegraphics` : son utilisation est la même que `includegraphics` du package `graphicx` ; en fait, elle s'assure simplement que le fichier graphique existe avant d'invoquer cette macro.

### 3.3 Problèmes

Ceci est une version expérimentale ; de nombreuses capacités de Maxima n'ont pas été testées et elle n'a pas été essayée avec les packages  $\text{\LaTeX}$  les plus importants. Ainsi, elle aura sûrement besoin de quelques ajustements.

Cependant, je pense que la plupart des problèmes apparaîtront lors de l'affichage de certaines sorties. Par exemple, si le résultat d'un calcul est trop long, il ne sera pas facile de le diviser en plusieurs lignes (sauf si l'on travaille dans Maxima puis que l'on copie le résultat dans le document, bien sûr). D'autres problèmes possibles peuvent être résolus à partir du document  $\text{\LaTeX}$ . Par défaut, Maxima ordonne les expressions par ordre alphabétique inverse ; par conséquent, si nous tapons :

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

nous obtenons :

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

Cela peut être évité en utilisant les fonctions Maxima `ordergreat` et `unorder` :

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

De plus, si nous souhaitons aligner plusieurs équations, nous devons aller un peu plus loin :

```
\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 Quelques derniers mots

Comme je l'ai mentionné précédemment, il s'agit d'un package expérimental qui aura probablement besoin de quelques modifications et ajouts, donc toutes les idées ou commentaires seront les bienvenus.

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

(Version française du 20 avril 2026)