# Maxiplot: 在 LaTeX 中使用 Maxima 和 Gnuplot in

.

2022 年 01 月 31 日

## 1 简介

众所周知, *Maxima* 是一个可以被用来计算微积分, 解方程, 求极限, 计算矢量或矩阵, 创建图表以及许多许多事的符号计算程序. 经过功能拓展, 它甚至具有编写程序的功能 (通过 Lisp 语言). 如果这些都不足以满足你的需求, 它是基于 GNU GPL 许可发行, 所以你可以自由的从 `http://maxima.sourceforge.net` 获取. 同时, 那里有许多不同语言编写的说明文档.

这个项目提供了一个 LaTeX 宏包以提供直接通过' 编程' 向 Tex 文件导入结果的功能, 这个过程不需要再依靠其他文件和借口. Maxima 代码可以直接嵌入到 LaTeX 文档中. 当它在处理文档时, 会生成一个后缀名为 `.mac` 的 Maxima 脚本文件. 这个新的文件可以直接被 Maxima 处理, 并自动创建另一个后缀名为 `.mxp` 的文件供 LaTeX 做进一步处理.

*Gnuplot* 指令也可以被潜入到 Tex 文件中, 这归功于 J. M. Mira 的工作. 使用 Gnuplot 将会引入另一个后缀名为 `.gnp` 辅助文件. Gnuplot 将如同 Maxima 处理 `.mac` 文件一样处理它.

最新版本的 Maxiplot 是在 2013 年发布的, 今天许多程序都发生了变化. 起初得知 Maxiplot 时我是无比兴奋的, 但后来发现它无法在我的膝上计算机上正常工作. 这份分支代码修复了许多我遇到的问题, 并且添加了一些方便的自动化构建的脚本. 另外, 根据修改后的代码重新编写了帮助文件. (这个分支代码使用 XeLaTex 作为编译器)

## 2 安装

使用新的版本需要拷贝 `maxiplot.sty`, `Makefile`, `build.sh` 或者 `build.csh`. 并且你需要预先安装 maxima, xelatex, bash 或者 csh. 如果你使用 openSUSE Leap, 可以使用 sudoers 组的用户执行下面的指令进行安装:

```
zypper ref
zypper ins texlive make maxima bash csh
```

原始的项目只需要拷贝 `maxiplot.sty` 到 LaTeX 可以找到的路径, 或者与你的文档相同的目录下.

## 3 LaTeX 宏包 maxiplot

### 3.1 如何使用?

要使用这个分支代码, 只需要使用下面的指令:

```
bash build.sh mydocument.tex
```

或使用 C SHELL:

```
csh build.csh mydocument.tex
```

然后, 你就会在 build 目录下得到 `mydocument.pdf` 啦.

如果你也使用 KDE Kile 作为集成编辑器, 你可以参考下面的步骤将这份分支的构建脚本添加到编译工具中:

1. 打开' 设置 - 配置 Kile - 工具 - 构建';

2. 在' 选择工具' 栏中点击' 新建', 键入一个像'Make' 这样的名字, 选择类为'XeLatex'. 然后点击完成;

3. 选择你添加的新工具, 在' 常规' 一栏中键入指令'bash build.sh' 或'csh build.csh'. 并在选项中键入'%source';

4. 最后一步, 设置' 高级' 一栏中' 类型' 为' 在 Konsole 中运行' , ' 类' 为' 编译'. 现在, 你将可以在构建选项卡中找到它.

这份分支的脚本将会刷新你的 Tex 文件, 这会使得 Kile 无法自动重载. 但在一般情况下无需担心.

以下是原始分支的构建方法:

用法非常简单, 像平时一样处理完你的文档. 然后运行下面的指令:

```
latex mydocument.tex
```

你将看到 `mydocument.mac` 出现在了你的工作目录下. 用 Maxima 处理它:

`maxima -b mydocument.mac`

如果你用到了 *Gnuplot* 指令:

`gnuplot mydocument.gnp`

重新处理 LaTeX 文档, *et voilà!* (给你吧!).

如果你的发行版允许, 你可以使能 `write18` 指令来允许 *Maxima* 和 *Gnuplot* 在你处理 LaTeX 文档时自动运行 (你需要提前添加可执行文件的安装目录到系统).

## 3.2 用户接口

### 3.2.1 Maxima.

这一节和下一节将使用一些例子来展示 `maxiplot` 的用法. 这需要一些 Maxima 的基础知识.

这个宏包有一个选项来允许兼容 `amsmath` 提供的 `pmatrix` 环境. 因此, 如果你需要用该环境创建矩阵, 可以在你的文档中添加下面的内容:

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

最重要的环境是 `maxima` 和 `maximacmd`. 他们提供的内容将用语产生 `.mac` 文件以供 Maxima 使用. 因此, 这些环境中应该使用非 LaTeX 风格的注释符号% . 也就是说, % 不能在 Maxima 代码中用于注释. 你应该使用 C 语言风格的语法 (/* 注释 */). 代码将被以参数的形式传输给处理函数, 因此, 他们应该以逗号分隔.

这是一个简单的例子:

```
\[   %进入数学编辑模式
\begin{maxima}
  f: x/(x^3-3*x+2),     /* 求积分 */
  tex('integrate(f,x)), /* 这里输出积分表达式... */
  print("="),
  tex(integrate(f,x)),  /* ...这里是结果 */
  print("+K")
\end{maxima}
\]   %离开数学编辑模式
```

如果使用本项目处理这段代码, 将会得到:

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

有一些 maxima 不能被包含的情况. 这时可以使用能马上给出结果的 maxima* . 它的输出可以在稍后通过 \maximacurrent 指令插入, 像这样:

```
\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 |

需要留心 `\maximacurrent` 指令将替换最后一个 *maxima* 代码块的输出结果, 所以必须在所有 `maxima` 代码块之前使用.

如果你想要稍后使用计算结果或者你想到别处使用它们, 你可以增加一些选项来存储它们. 上面的例子可以通过以下的方式重新实现:

```
\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}
```

注意, 当输入 “table” 作为 maxima* 的参数时 反斜杠 (\) 是不需要的.

这是编辑模式下具有同样用途的例子: \imaxima 指令 (命名来自 “inline maxima” , 嵌入的 Maxima).

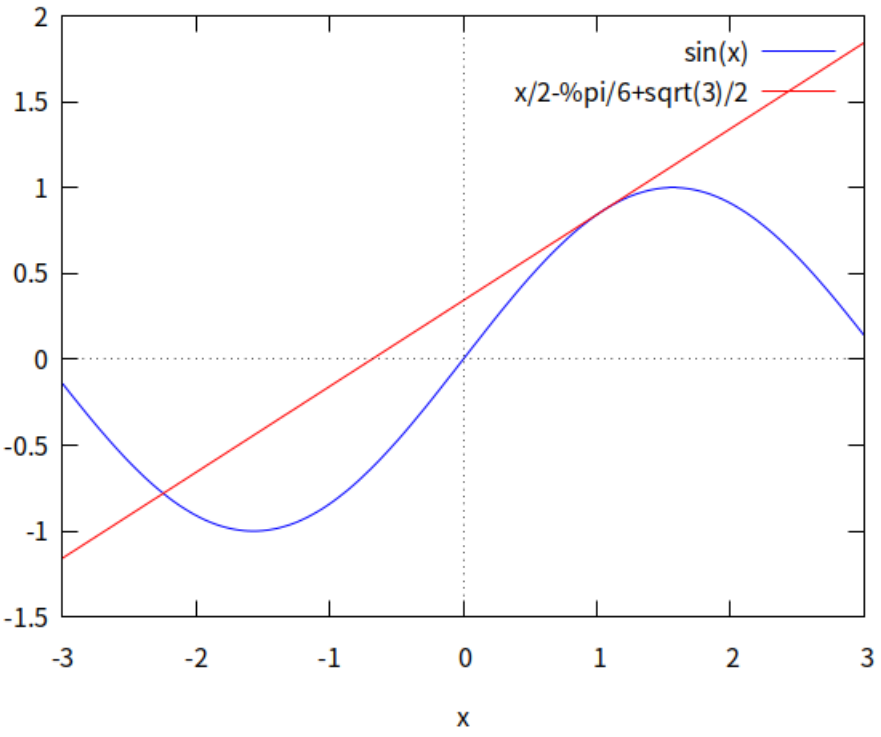
```
  \[
  \overline{x}=\imaxima{tex(xx:fx/N)}\qquad
  \sigma^2=\imaxima{tex(sx2:fx2/N-xx^2)}\qquad
  \sigma=\imaxima{tex(sqrt(sx2))}
  \]
```

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

当没有输出文件时 (如定义函数或加载 Maxima 库), 应该使用 maximacmd 环境或\imaximacmd 指令. 他俩没有 * 变体或任何其他选项. 进一步的, *Maxima* 内置指令必须通过分号 (;) 分开. 如果可以, 用美元符号 ($) 则更好. 让我们来看一下 *Maxima*/*Gnuplot* 接口的例子. 这个例子展示了如何绘制 sin 函数和它在 $\frac{\pi}{3}$ 处的切线:

```
\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}
```

这个代码创建了一个png 格式的图像文件 maxiplot_zh2D.png:



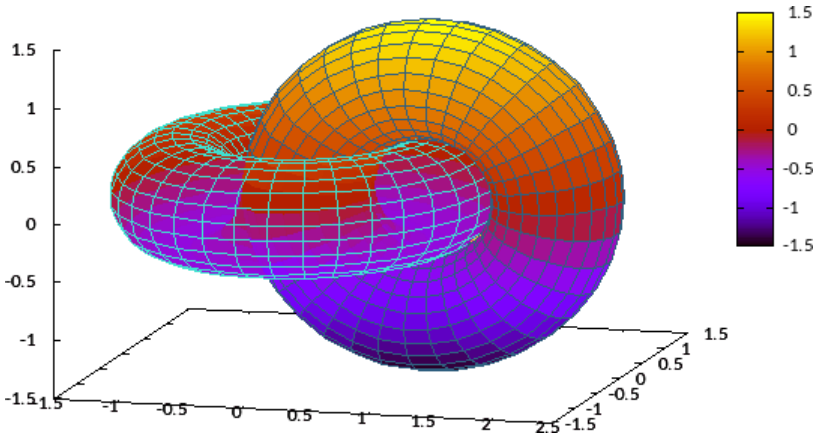

这个环境包含的 LaTeX 命令在加入 `mac` 文件之前将被替换. 有时遇到具体的字符串可能会引入麻烦, 所以这是可能并不需要这个功能. `vmaxima` 和 `vmaximacmd` 环境可以解决这个问题. 它们的用法与之前一样, 但是具体的文本将被直接传送. 本环境就基于 `verbatim` LaTeX 宏包.

### 3.2.2 Gnuplot

*Maxima* 通过 *Gnuplot* 创建图表, 有时我们需要直接使用后者. 这时, 可以使用 `gnuplot` 和它的详细版本 `vgnuplot` .

这是一个绘制 3 维图像的例子:

```
\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}
```




让我们来测试一下 `\mxpIncludegraphics` 指令: 它的用法与`graphicx` 包中的 `includegraphics` 相同; 事实上, 他仅仅是在引用宏之前调用已经存在的绘图文件.

### 3.3 问题

这是一个专业的版本, 许多 Maxima 功能没有经过测试, 并且也没有与更多的 LaTeX 包共同测试. 因此, 它一定需要调整.

然而, 我想大多数问题都将会有详细的输出. 如果计算结果表达式过长将不容易被分割成多行 (当然, 你可以在 Maxima 中处理好并且粘贴到文档中).

问题也可能是 LaTeX 文档引起的. 一般而言, Maxima 反序处理字符输入. 如果我们输入:

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

事实上将得到:

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

可以通过 Maxima 函数 `ordergreat` 和 `unorder` 解决:

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

进一步的, 如果希望对齐多个方程, 我们需要一些跟进一步的知识 (内嵌 Lisp 语言):

```
\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 最后一点话

就像我之前提到的, 这是一个专业的宏包, 它需要进一步的修改. 所以欢迎任何想法和评论.

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

(英语翻译由 Jaime Villate 提供)
(中文翻译由 Umaru Aya(凝萌 · 曦子) 提供)

## 5 关于这个分支

这个文档实在原始文档的基础上直接修改的, 所以你在阅读时应该多加小心. 新修改的内容只在我的膝上计算机上测试通过. 同样欢迎任何想法和评论.

凝萌 · 曦子
<umaru@umaru.science>