

# wxMaxima

## Contents

wxMaxima 用户手册	2
1 wxMaxima 简介	3
1.1 Maxima 和 wxMaxima	3
1.1.1 Maxima	4
1.1.2 wxMaxima	4
1.2 工作簿 (Workbook) 基础	5
1.2.1 工作簿的使用	5
1.2.2 单元格	6
1.2.3 水平光标和垂直光标	6
1.2.4 向 Maxima 发送单元格	8
1.2.5 命令自动补全	8
1.2.6 侧边栏	11
1.2.7 MathML 输出	11
1.2.8 标记语言 (Markdown) 支持	12
1.2.9 快捷键	12
1.2.10 导出 TeX 时的原始 TeX 代码	12
1.3 文件格式	12
1.3.1 .mac	12
1.3.2 .wxm	14
1.3.3 .wxmx	14
1.4 设置选项	14
1.4.1 默认动图帧率	14
1.4.2 新 Maxima 会话的默认绘图尺寸	14
1.4.3 在文本中匹配括号	14
1.4.4 不自动保存工作簿	14
1.4.5 设置信息保存的位置	16
2 对 Maxima 功能的扩充	16
2.1 为变量添加下标	16
2.2 在状态栏中提供反馈信息	16
2.3 绘图	17
2.3.1 将绘制的图形嵌入工作簿	17
2.3.2 设置嵌入图形的大小	17
2.3.3 更好的绘图质量	17

2.3.4	在交互式 <code>gnuplot</code> 窗口中打开嵌入的绘图	18
2.3.5	在绘图窗口中打开 <code>gnuplot</code> 控制台	18
2.3.6	在工作簿中嵌入动图	18
2.3.7	同时打开多个绘图窗口	20
2.3.8	“用 <code>draw</code> 函数绘图” 侧边栏	20
2.4	嵌入图片	22
2.5	启动时读取的文件	22
2.6	以 <code>wx</code> 开头的特殊变量	22
2.7	输出美观的二维公式	22
2.8	错误 (bug) 报告	23
2.9	让输出显示为红色	23
3	疑难解答	23
3.1	无法连接到 Maxima	23
3.2	从损坏的 <code>wxmx</code> 文件中取回信息的方法	24
3.3	在命令完成前显示一些调试信息	24
3.4	绘图结果是空的绘图框和错误信息	25
3.5	绘制动图时得到 “error: undefined variable (错误: 未定义的变量)”	25
3.6	删除了一个单元格, 而撤销操作不起作用	25
3.7	<code>wxMaxima</code> 启动时显示信息 “Maxima process terminated (Maxima 进程已终止)”	25
3.8	Maxima 永远在计算中而不响应输入	26
3.9	基于 SBCL 的 Maxima 耗尽内存	26
3.10	在 Ubuntu 系统上, 有时输入无响应或响应迟缓	26
3.11	<code>wxMaxima</code> 在处理希腊字符或元音变音符时停止	26
3.12	绘图	26
3.12.1	是否可以同时输出和嵌入绘图文件?	26
3.12.2	是否可以指定绘制图形的相对比例?	28
4	常见提问 (FAQ)	28
4.1	有没有办法在 LaTeX 页面中显示更多的文字?	28
4.2	是否有深色模式?	28
4.3	<code>wxMaxima</code> 有时在启动后的一段时间内无响应	29
5	命令行参数	29

## wxMaxima 用户手册

wxMaxima 是计算机代数系统 (CAS, computer algebra system) Maxima 的图形用户界面 (GUI, graphical user interface)。在 wxMaxima 中, 用户可以使用 Maxima 所有的命令。此外, wxMaxima 还提供了一些向导以方便用户使用最常用的功能。本手册描述了一些 wxMaxima 的特性, 这些特性使其成为最流行的 Maxima 图形用户界面之一。

在介绍本手册的内容之前, 先给出一些关于文档导航的说明。正下方是一个简短的目录。单击任何列出的条目, 即可跳转到手册对应节的顶部。单击目录可定位到详细的目录。此扩展目录可用于更直接地跳转到手册的各小节、小小节, 以方便查看内容。



Figure 1: wxMaxima 的 logo

- 简介：

wxMaxima 基础知识

- 扩充内容：

wxMaxima 向 Maxima 添加的命令

- 疑难解答：

当 wxMaxima 不按预期运行时的解决办法

- 常见提问 (FAQ)：

常见提问 (Frequently asked questions)

- 命令行：

wxMaxima 支持的命令行参数

---

## 1 wxMaxima 简介

### 1.1 Maxima 和 wxMaxima

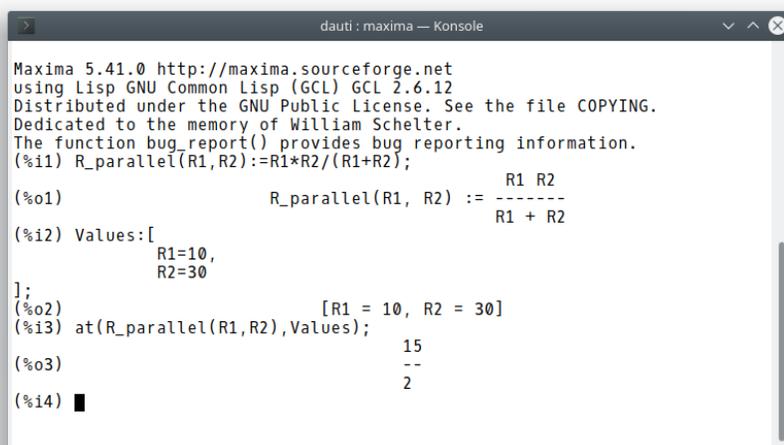
在开源领域，大型项目通常被分割成小型项目，后者对于小规模开发组来说更容易处理。例如，一个 CD 刻录机程序会包括一个实际刻录 CD 的命令行程序和一个图形用户界面，该界面使得用户不必了解所有的命令行参数就能刻录 CD，从而实际上避开了使用命令行。这种方法的一个优点是，投入到命令行程序中的开发工作可以被许多程序共享：同一个刻录 CD 的命令行程序可以用作文件管理器应用程序的“发送到 CD”插件、音乐播放器的“刻录到 CD”功能以及 DVD 备份工具的刻录 CD 功能。另一个优点是将一个大型任务分解成更小的部分时，开发人员就可以为同一个程序提供多个用户界面。

一个像 Maxima 这样的计算机代数系统 (CAS, computer algebra system) 也适合前述框架。Maxima 可以用作任意精度计算器程序的底层软件，也可以在更大计算系统中作为公式转换程

序来帮助计算（例如 Sage）。Maxima 也可以直接作为独立的计算系统使用，它可以通过命令行来访问。然而，在通常情况下，像 wxMaxima 这样的用户界面是使用 Maxima 的一种更有效的方法，特别是对于新手而言更是如此。

### 1.1.1 Maxima

Maxima 是一个功能齐全的计算机代数系统（CAS）。CAS 是一个程序，它可以通过对公式变形并找到问题的解来求解数学问题，而不是仅仅输出结果的数值。换言之，Maxima 可以作为一个计算器，给出变量的数值表示，也可以提供解析解。此外，它还提供了一系列数值分析方法，用于求解没有解析解的方程或方程组。



```
dauti: maxima — Konsole
Maxima 5.41.0 http://maxima.sourceforge.net
using Lisp GNU Common Lisp (GCL) GCL 2.6.12
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) R_parallel(R1,R2):=R1*R2/(R1+R2);
(%o1)
          R1 R2
R_parallel(R1, R2) := -----
          R1 + R2
(%i2) Values:[
          R1=10,
          R2=30
];
(%o2) [R1 = 10, R2 = 30]
(%i3) at(R_parallel(R1,R2),Values);
(%o3)
          15
          --
          2
(%i4) █
```

Figure 2: Maxima 截图，命令行

Extensive documentation for Maxima is available in the internet. Part of this documentation is also available in wxMaxima's help menu. Pressing the Help key (on most systems the F1 key) causes wxMaxima's context-sensitive help feature to automatically jump to Maxima's manual page for the command at the cursor.

### 1.1.2 wxMaxima

wxMaxima 是一个图形用户界面，它提供了 Maxima 的全部功能和灵活性。wxMaxima 为用户提供了图形显示和许多其他功能，使得 Maxima 更易于使用。例如，wxMaxima 允许用户通过右键单击将任何单元格的内容（或者，如果需要，也可以导出公式的任何部分）作为文本、LaTeX 代码或 MathML 代码导出。实际上，用户可以将整个工作簿（workbook）导出为 HTML 文件或 LaTeX 文件。有关 wxMaxima 的文档，包括说明其各种使用方法的工作簿，都可以在 wxMaxima 的帮助站点上找到，也可以通过“帮助”菜单查看相关说明。

在 wxMaxima 中输入的计算式是由 Maxima 命令行工具在后台执行的。

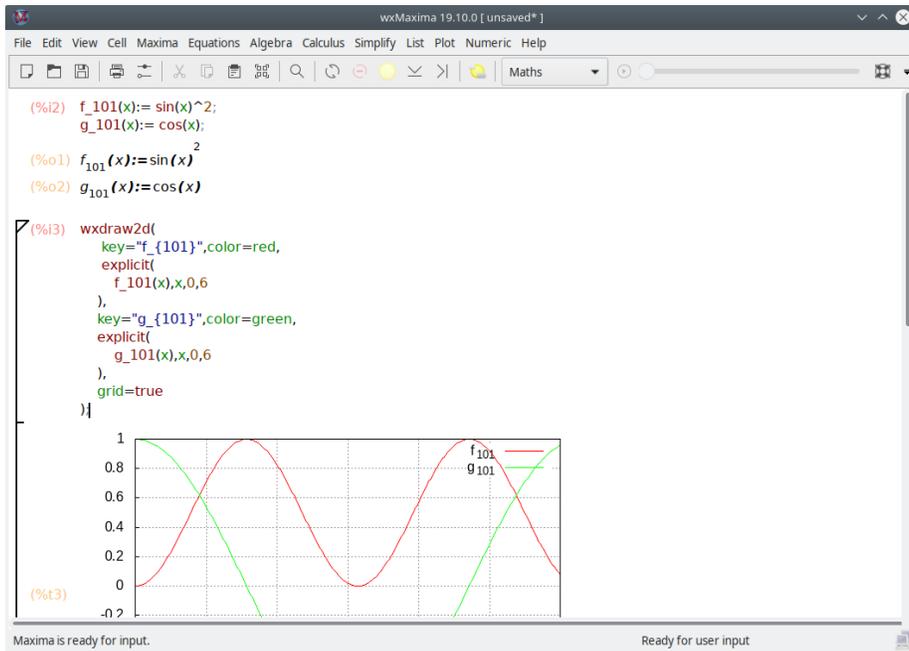


Figure 3: wxMaxima 界面

## 1.2 工作簿 (Workbook) 基础

大部分 wxMaxima 功能是不言自明的，但有些细节需要注意。这个网站包含多个工作簿，这些工作簿涉及 wxMaxima 的各个方面。学习其中的一些（特别是“10 minute (wx)Maxima tutorial”，即“十分钟 (wx)Maxima 教程”）将增加用户对 Maxima 的内容和 wxMaxima 与 Maxima 交互过程的熟悉度。本手册集中描述 wxMaxima 的某些功能，这些功能可能并非不言而喻的，也可能不会被在线文档涵盖。

### 1.2.1 工作簿的使用

wxMaxima 的特性之一是可以将发送给 Maxima 的命令提前组织到单元格中，这些单元格只在用户请求时才被处理（这意味着：把命令发送给 Maxima）。在处理单元格时，该单元格（且仅该单元格）中的所有命令将作为批处理进行计算。（前面的语句并不十分准确，因为可以选择一组相邻的单元格并且一起计算它们，或者让 Maxima 一次性算完工作簿中的所有单元格。）wxMaxima 提交命令的方法乍一看可能会觉得陌生。但是，这种做法确实大大简化了处理大型文档的工作（用户不希望每次更改都自动触发对整个文档的重新计算）。而且，该方法用于调试命令时非常方便。

如果输入文本到 wxMaxima 中，则会自动创建一个新的工作簿单元格。可以在工具栏中选择此单元格的类型。如果创建了一个（数学）代码单元格，则可以将该单元格发送到 Maxima，这将使得计算结果显示在代码下方。下面显示了一对这样的命令：

在计算（数学）输入单元格的内容时，Maxima 会为该输入指定一个标签（默认情况下以红

```
(%i2) f_101(x):= sin(x)^2;
      g_101(x):= cos(x);

(%o1) f_101(x):= sin(x)^2
(%o2) g_101(x):= cos(x)
```

Figure 4: 输入、输出单元格

色显示，并且可以由“%i”加数字识别），以便在 wxMaxima 会话的后续内容中引用该标签。Maxima 生成的输出也会获得以“%o”开头的标签，而当用户为输出指定了名称（如变量名）时，该标签是隐藏的。在后一种情况下，默认显示用户定义的名称。不过，此时仍可以访问 Maxima 自动生成的“%o”型标签。

除了（数学）输入单元格外，wxMaxima 中还可以使用文本单元格、图片单元格、标题单元格、节单元格和小节单元等等。每个单元格都有独立的撤销缓冲区，因此通过更改多个单元格的内容，然后逐步撤销不需要的更改来进行调试，是非常容易的。此外，工作簿本身有一个全局撤销缓冲区，可以撤销单元格编辑、添加和删除操作。

下图显示了不同的单元格类型（标题单元格、节单元格、小节单元格、文本单元格、输入/输出单元格和图片单元格）：

### 1.2.2 单元格

工作簿由单元格组成。每个单元格可以包含以下类型的内容：

- 一行或多行 Maxima 命令；
- 一个或多个图片；
- Maxima 的输出（含提问、报错信息）；
- 文本块，例如可用于组成文档的段落；
- 文档标题、节标题或小节标题等。

当输入文本时，wxMaxima 的默认行为是自动创建一个数学单元格。可以使用“单元格”菜单、菜单中显示的热键或工具栏中的下拉列表创建其他类型的单元格。一旦创建了非数学单元格，输入到文档的任何内容都将被解释为文本。

额外的注释文本可以输入到一个数学单元格中，格式为：/\* 此注释将不会发送到 Maxima 中进行计算 \*/。

### 1.2.3 水平光标和垂直光标

在一般的文字处理程序中，如果用户试图选择多个行，该程序将尝试扩展选择范围，使其自动包括这些行的所有内容。类似地，在 wxMaxima 中，如果同时选择了多个单元格，则 wxMaxima 会将选择范围扩展到所有这些单元格的内容。

The screenshot shows the wxMaxima 19.10.0 interface. The main window contains the following content:

- Title cell:** **This is a title cell**
- Section cell:** **1 This is a section cell**
- Subsection cell:** **1.1 This is a subsection cell**
- Subsubsection cell:** **1.1.1 This is a subsubsection cell**
- Heading level 5 cell:** **1.1.1.1 This is a heading of level 5**
- Heading level 6 cell:** **1.1.1.1.1 This is a heading of level 6**

Below the headings, there is a text cell: "Now a Maxima input cell (and the computed output):"

Input cell: `(%i1) 'integrate(x^2,x)=integrate(x^2,x);`

Output cell: `(%o1)  $\int x^2 dx = \frac{x^3}{3}$`

Text cell: "This is a text only cell. Now comes a pagebreak cell and then a image cell:"

Image cell: 

Figure 1: `./home/dauti/Software/wxmaxima/info/wxmaximaLogo.png`

Table of Contents (right sidebar):

- This is a title cell
- This is a section cell
- This is a subsection cell
- This is a subsubsection cell
- This is a heading of level 5
- This is a heading of level 6**

Insert menu (bottom right):

- Text
- Title
- Section
- Subsection
- Subsubsection
- Heading 5
- Heading 6
- Image
- Pagebreak

Status bars: "Maxima is ready for input." (bottom left), "Ready for user input" (bottom right)

Figure 5: wxMaxima 的不同类型单元格

与一般程序不同的是，wxMaxima 通过定义两种类型的光标，提供了操作灵活性。wxMaxima 将在需要时自动在它们之间切换：

- 如果光标在两个单元格之间移动或单击，则会使用水平光标；
- 如果在单元格内输入时，则会使用垂直光标。此种光标是当使用鼠标指针或方向键在单元格内移动光标时被激活的，与文本编辑器中光标的激活方式非常相似。

#### 1.2.4 向 Maxima 发送单元格

只要按下键盘上的 CTRL+ENTER、SHIFT+ENTER 或 ENTER 键，(数学) 代码单元格中的命令就会执行。wxMaxima 的默认设置是 CTRL+ENTER 或 SHIFT+ENTER 键会使得命令由 Maxima 执行，但也可以设置为按下 ENTER 键执行命令。

#### 1.2.5 命令自动补全

wxMaxima 包含自动补全功能，该功能可以通过菜单“单元格->自动补全”(Cell->Complete Word)或按组合键 CTRL+K 来触发。自动补全是上下文相关的。例如，如果在使用 ezUnits (单位换算包) 的单元格中激活，它将提供适用的单位列表。

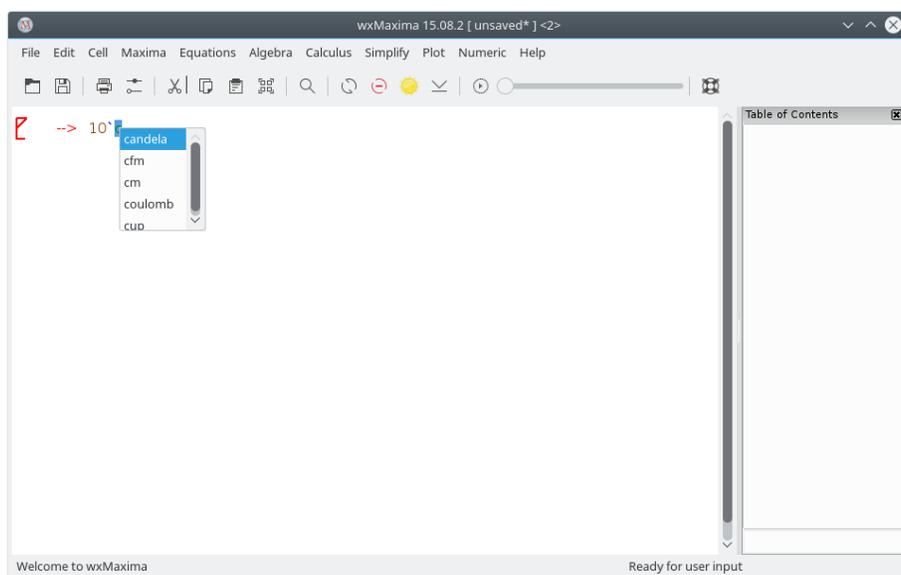


Figure 6: ezUnits

除了补全文件名、单位名、当前命令或变量的名称之外，自动补全功能还可以显示大多数命令的模板，指示该命令所需参数的类型(和含义)。要激活此功能，可按下 SHIFT +CTRL +K 或选择相应的菜单项(“单元格->显示模板”，Cell->Show Template)。

##### 1.2.5.1 希腊字母

计算机一般以 8 位值 (8-bit) 存储字符, 所以最多可以储存 256 个不同字符。大部分语言中的所有字母、数字和控制符号 (传输结束、字符串结束、为菜单绘制矩形的线条和边缘等) 都符合该限制。

不过, 对于大多数国家来说, 256 个字符的代码页并没有包括希腊字母, 而这些字母是数学上常用的。为了克服这种限制, Unicode 被发明了: 一种使英文文本储存方式不变的编码, 可使用的字符数远超 256 个。

如果使用支持 Unicode 或不关心字体编码的 Lisp 编译器来编译 Maxima, 则在 Maxima 中可以使用 Unicode。前述两种情况中一般至少有一种是真的。wxMaxima 提供了一种使用键盘输入希腊字母的方法:

- 可以通过按下 ESC 键, 然后开始键入希腊字母的英文名称来输入该字母。
- 或者, 也可以通过按下 ESC、一个英文字母 (对希腊字母 omicron 而言是两个) 并再次按下 ESC 来输入。在这种情况下, 支持以下字母:

按键	希腊字母	按键	希腊字母	按键	希腊字母
a	alpha	i	iota	r	rho
b	beta	k	kappa	s	sigma
g	gamma	l	lambda	t	tau
d	delta	m	mu	u	upsilon
e	epsilon	n	nu	f	phi
z	zeta	x	xi	c	chi
h	eta	om	omicron	y	psi
q	theta	p	pi	o	omega
A	Alpha	I	Iota	R	Rho
B	Beta	K	Kappa	S	Sigma
G	Gamma	L	Lambda	T	Tau
D	Delta	M	Mu	U	Upsilon
E	Epsilon	N	Nu	P	Phi
Z	Zeta	X	Xi	C	Chi
H	Eta	Om	Omicron	Y	Psi
T	Theta	P	Pi	O	Omega

用同样的方法还可以输入一些其他的数学符号:

keys to enter	mathematical symbol
hbar	Planck's constant: a h with a horizontal bar above it
Hbar	a H with a horizontal bar above it
2	squared
3	to the power of three
/2	1/2
partial	partial sign (the d of dx/dt)
integral	integral sign
sq	square root

keys to enter	mathematical symbol
ii	imaginary
ee	element
in	in
impl implies	implies
inf	infinity
empty	empty
TB	big triangle right
tb	small triangle right
and	and
or	or
xor	xor
nand	nand
nor	nor
equiv	equivalent to
not	not
union	union
inter	intersection
subsetq	subset or equal
subset	subset
notsubsetq	not subset or equal
notsubset	not subset
approx	approximately
propto	proportional to
neq != or #	not equal to
+/- or pm	a plus/minus sign
<= or leq	equal or less than
>= or geq	equal or greater than
<< or ll	much less than
>> or gg	much greater than
qed	end of proof
nabla	a nabla operator
sum	sum sign
prod	product sign
exists	there exists sign
nexists	there is no sign
parallel	a parallel sign
perp	a perpendicular sign
leadsto	a leads to sign
->	a right arrow
->	a long right arrow

如果某个特殊的 Unicode 字符不在上表中，则可以用如下方式输入：按下 ESC 后，输入该字符的 16 进制编码，再按下 ESC。

因此 ESC 61 ESC 输入了字符 a。

请注意，这些符号中的大多数（逻辑符号除外）在 Maxima 中没有特殊含义，因此将被解释为普通字符。如果使用不支持 Unicode 字符的 Lisp 编译器编译出 Maxima，则这些符号可能会导致输出错误消息。

### 1.2.6 侧边栏

侧边栏可以用来访问一些常用的 Maxima 命令快捷方式、目录、包含调试消息的窗口或已执行命令的历史记录等。使用“查看”菜单可以启用这些侧边栏。它们都可以移动到 wxMaxima 窗口内部或外部。其中一个常用的是可用鼠标点选输入希腊字母的侧边栏。

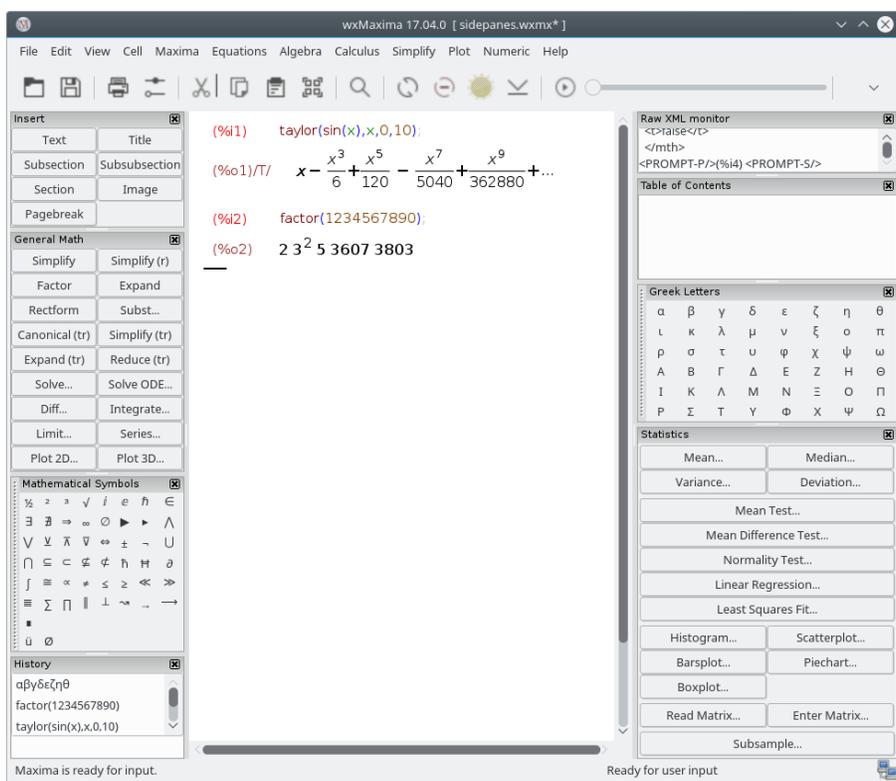


Figure 7: 各种类型的侧边栏

### 1.2.7 MathML 输出

一些文字处理程序可以识别 MathML 输入并自动将其作为可编辑的数学公式插入，或者（像 LibreOffice 7.1 那样）有一个提供“从剪贴板导入 MathML”功能的公式编辑器。有些程序也支持 RTF 公式。因此，wxMaxima 在右键单击菜单中提供了多种复制公式的方法。

## 1.2.8 标记语言 (Markdown) 支持

wxMaxima 提供了一些标准的标记 (markdown) 语言支持, 而这些记号不会与数学符号相冲突。其中一个支持是项目符号列表。

普通文本

- \* 一个条目, 缩进级别为 1
- \* 另一个条目, 缩进级别为 1
  - \* 一个条目, 缩进级别为 2
  - \* 另一个条目, 缩进级别为 2
- \* 第三个条目, 缩进级别为 1

普通文本

wxMaxima 也能将以 > 字符开头的文本识别为引用块:

普通文本

> 引用文字

> 引用文字

> 引用文字

普通文本

wxMaxima 的 TeX 和 HTML 输出也能识别 =>, 并将其替换为相应的 Unicode 符号:

cogito => sum.

导出 TeX 和 HTML 文件时将识别的其他符号还有不等号 ( $\leq$  和  $\geq$ )、双线箭头 ( $\Leftrightarrow$ )、单线箭头 ( $\leftarrow$ 、 $\rightarrow$  和  $\leftrightarrow$ ) 以及  $+/-$ 。对于 TeX 输出, 还可以识别  $\ll$  和  $\gg$ 。

## 1.2.9 快捷键

大多数快捷键可以在相应菜单的文本中找到。由于它们实际上是从菜单文本中获取的, 所以可以通过 wxMaxima 的翻译文本进行定制, 以满足本地用户的需求, 因此我们不在这里记录它们。但是, 一些快捷键或其别名并没有记录在菜单中:

- CTRL+SHIFT+DELETE 删除一个单元格。
- CTRL+TAB 或 CTRL+SHIFT+TAB 开启自动补全功能。
- SHIFT+SPACE 插入一个不可断行的空格。

### 1.2.10 导出 TeX 时的原始 TeX 代码

如果文本单元格以 TeX: 开头, 则导出 TeX 时, 文件将包含 TeX: 标记后面的文本。使用此功能可以在 wxMaxima 工作簿中输入 TeX 代码。

## 1.3 文件格式

在 wxMaxima 中输入的内容可以存为以下三种文件格式之一, 以供以后使用:

### 1.3.1 .mac

.mac files are ordinary text files that contain Maxima commands. They can be read using Maxima's `batch()` or `load()` command or wxMaxima's File/Batch File menu entry.

One Example is shown below. `Quadratic.mac` defines a function and afterwards generates a plot with `wxdraw2d()`. Afterwards the contents of the file `Quadratic.mac` are printed and new defined function `f()` is evaluated.

```
(%i1) batch("Quadratic.mac");
      read and interpret /tmp/Quadratic.mac
(%i2) f(x):=(x-5)^2
(%i3) wxdraw2d(explicit(f(x),x,-5,15))
(%t3)
      100
      80
      60
      40
      20
      0
      -5      0      5      10      15
(%o4) /tmp/Quadratic.mac
(%i5) printfile("Quadratic.mac")$
      f(x):=(x-5)^2$
      wxdraw2d(explicit(f(x),x,-5,15))$
(%i6) f(7);
(%o6) 4
```

Figure 8: Loading a `.mac` file with `batch()`

Attention: Although the file `Quadratic.mac` has an usual Maxima extension (`.mac`), it can only be read by `wxMaxima`, since the command `wxdraw2d()` is a `wxMaxima`-extension to Maxima.

用户可以使用 `.mac` 文件编写自己的程序包。但是由于它们没有包含足够的结构信息，因此不能作为一个 `wxMaxima` 工作簿来读取。

### 1.3.2 .wxm

.wxm 文件包含除 Maxima 输出之外的工作簿内容。在版本号大于 5.38 的 Maxima 上，可以像 .mac 文件一样，使用 Maxima 的 load() 函数来读取。如果采用这种文本格式，使用含新功能的工作簿时会不可避免地旧版本的 wxMaxima 不兼容。

### 1.3.3 .wxmx

这种基于 XML 的文件格式保存了完整的工作簿内容，包括缩放因子和观察列表等内容。它是首选的文件格式。

## 1.4 设置选项

对于一些常见的可设置变量，wxMaxima 提供了两种设置方法：

- 下面的设置对话框允许用户更改当前和后续会话的默认值。
- 对于当前会话，要变更多数可设置变量的值只能通过从工作簿添加相应的命令，见后续说明。

### 1.4.1 默认动图帧率

用于新生成动图的帧率保存在变量 wxanimate\_framerate 中，可以使用设置对话框更改此变量在新工作簿中的初始值。

### 1.4.2 新 Maxima 会话的默认绘图尺寸

如果 wxplot\_size 的值没有被 Maxima 更改，则使用此大小创建绘图并嵌入工作簿。

为了仅对单个绘图命令设置绘制图形的大小，可以使用以下代码：

```
wxdraw2d(
  explicit(
    x^2,
    x, -5,5
  )
), wxplot_size=[480,480]$
```

### 1.4.3 在文本中匹配括号

该选项使以下两个功能生效：

- 如果输入了左括号、左方括号或左双引号，wxMaxima 将自动在其后插入与之配对的右括号、右方括号或右双引号。
- 如果在输入前述字符中的任何一个时选择了文本，则所选文本将被放在配对的两个字符之间。

### 1.4.4 不自动保存工作簿

如果设置了此选项，则只有在用户请求时才会覆盖工作簿所在的文件。为了防止发生碰撞、断电等突发情况，临时目录 (temp) 中仍保留了一个最近的备份文件。

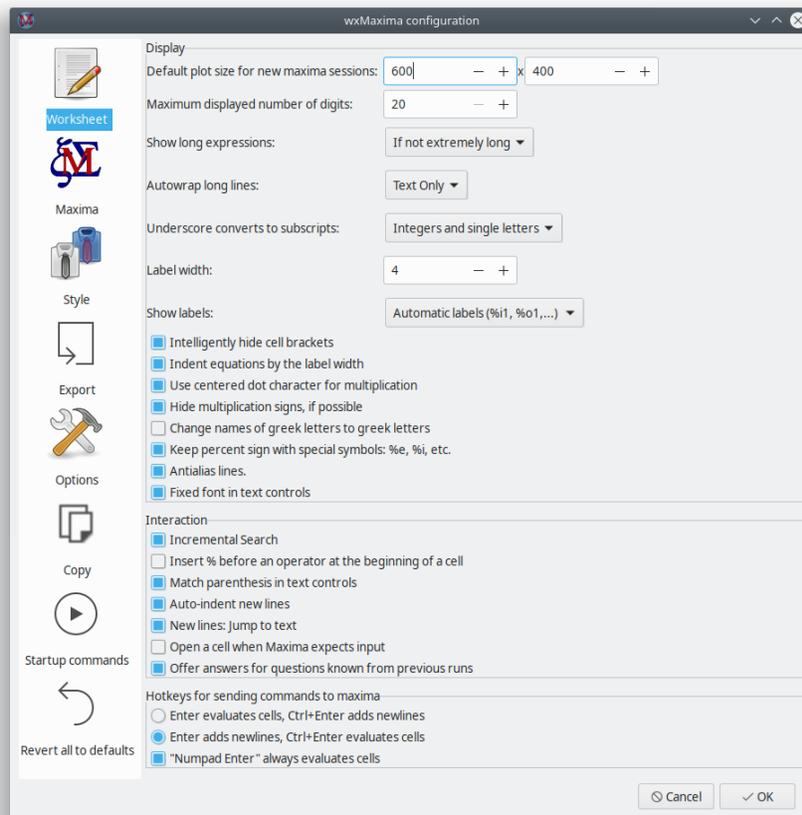


Figure 9: wxMaxima 设置对话框

如果未设置此选项，则 wxMaxima 的行为更像现代手机软件：

- 文件在退出时被自动保存；
- 每 3 分钟文件被自动保存一次；

#### 1.4.5 设置信息保存的位置

当使用 Unix/Linux 系统时，设置信息将保存在主目录中的 .wxMaxima 文件里（如果 wxWidgets 版本号小于 3.1.1）或 .config/wxMaxima.conf 文件里（如果 wxWidgets 的版本号大于等于 3.1.1（XDG 标准））。用户可以通过命令 wxbuild\_info()；或使用菜单选项“帮助->关于”（Help->About）来检索 wxWidgets 版本号。wxWidgets 是跨平台的 GUI 库，也是 wxMaxima 的底层库（因此 wxMaxima 的名称中带“wx”）。（由于文件名以点“.”开头，文件 .wxMaxima 或 .config 将被隐藏）。

当使用 Windows 系统时，配置信息将存储在注册表中。在注册表中的以下位置可以找到 wxMaxima 的相关条目：HKEY\_CURRENT\_USER\Software\wxMaxima。

---

## 2 对 Maxima 功能的扩充

wxMaxima 是 Maxima 的图形用户界面，因此它的主要目的是将命令传递给 Maxima，并输出执行这些命令的结果。但是，在某些情况下，wxMaxima 会为 Maxima 添加新的功能。前面已提到 wxMaxima 可以将工作簿的内容导出为 HTML 和 LaTeX 文件。本节将讨论另外一些增强功能，以及在工作簿中包含图形的方法。

### 2.1 为变量添加下标

如果 wxsubscripts 设置为 true，则当以下条件满足时，将使用下标来显示格式为 x<sub>y</sub> 的变量名：

- y 是一个字母；
- y 是一个整数。

如果变量格式与上述条件不匹配，仍可以使用命令 wxdeclare\_subscript(variable\_name)；或 wxdeclare\_subscript([variable\_name1,variable\_name2,...])；来显示下标。声明为下标的变量可以使用以下命令还原格式：wxdeclare\_subscript(variable\_name,false)；。

### 2.2 在状态栏中提供反馈信息

长时间运行的命令中可以包含 wxstatusbar()，以用于在状态栏中向用户提供反馈信息。此反馈信息将被任何新生成的反馈信息覆盖（从而可以将其用作运行进度指示器），并在发送到 Maxima 的命令运行结束后立即被删除。即使在只与 Maxima（而非 wxMaxima）一起使用的库中包含命令 wxstatusbar() 也是可行的：如果没有运行 wxMaxima，那么命令 wxstatusbar() 也不会被执行。

```
for i:1 thru 10 do (  
  /* 告诉用户目前的运行位置 */  
  wxstatusbar(concat("Pass ",i)),
```

```

/* (sleep n) 是可接在字符"?" 后使用的 Lisp 函数 */
/* 它推迟了程序的运行, 此处是 3 秒 */
?sleep(3)
)$

```

## 2.3 绘图

因为绘图功能与图形密切相关, 所以图形用户界面理应为原始程序的绘图功能提供一些扩充内容。

### 2.3.1 将绘制的图形嵌入工作簿

Maxima 在创建图形时, 通常会指示外部程序 `gnuplot` 用单独的窗口展示每个图形。因为在工作簿中嵌入图形通常会更方便, 所以 `wxMaxima` 提供了自己的一组绘图函数。这些函数与其名称所对应的 Maxima 函数在使用时没有区别, 但是都以“wx”为前缀。例如, `wxplot2d` 对应于 `plot2d`, `wxplot3d` 对应于 `plot3d`, `wxdraw` 对应于 `draw`, `wxhistogram` 对应于 `histogram`。

### 2.3.2 设置嵌入图形的大小

如前面所述, 设置对话框提供了一种更改默认绘图大小的方法, 该方法设置了 `wxplot_size` 的默认值。绘图时, `wxMaxima` 将该变量识别为图片尺寸 (以像素为单位)。可以查询该变量, 或以如下方式设置所绘制图片的大小:

```

wxplot_size:[1200,800]$
wxdraw2d(
  explicit(
    sin(x),
    x,1,10
  )
)$

```

如果只更改一个绘制图片的大小, `wxMaxima` 提供了一种只为当前单元格更改属性的方法。具体用法为, 将 `wxplot_size=[value1,value2]` 附加到 `wxdraw2d()` 命令结尾, 而不是作为该命令的一部分:

```

wxdraw2d(
  explicit(
    sin(x),
    x,1,10
  )
),wxplot_size=[1600,800]$

```

### 2.3.3 更好的绘图质量

`Gnuplot` 似乎没有提供一种通用的方法来确定它是否支持 `cairo` 库提供的高质量位图输出。如果所编译的 `gnuplot` 使用了 `cairo` 库, 勾选设置菜单中的 `pngcairo` 选项 (可被变量 `wxplot_pngcairo` 覆盖) 可以提供抗锯齿和其他线条样式支持。如果设置了 `wxplot_pngcairo`, 而 `gnuplot` 并未使用 `cairo` 库, 则绘图将会失败并给出错误消息。

### 2.3.4 在交互式 gnuplot 窗口中打开嵌入的绘图

如果绘图时使用 `wxdraw` 类型的命令（不含 `wxplot2d` 和 `wxplot3d`），并且底层的 `gnuplot` 项目文件不是太大，则 `wxMaxima` 提供了一个右键单击菜单，允许在交互式的 `gnuplot` 窗口中打开绘制的图形。

### 2.3.5 在绘图窗口中打开 gnuplot 控制台

在 Windows 系统上，如果 `Maxima` 的变量 `gnuplot_command` 的值“`gnuplot`”被换为“`wgnuplot`”，则 `gnuplot` 会打开一个控制台窗口，在该窗口中可以输入 `gnuplot` 的命令。但是，启用此功能会使得每次绘图时 `gnuplot` 短时间内独占键盘。

### 2.3.6 在工作簿中嵌入动图

在三维图形中，点的坐标往往难以读取。一个可行的替代方法是将第三个参数指定给鼠标滚轮（即绘制多个二维图形）。`with_slider_draw` 命令是 `wxdraw2d` 的一个版本，它可以绘制多幅图片，使用户可以通过移动菜单栏中的滑块（或滚动鼠标滚轮）在这些图片之间切换。`wxMaxima` 支持将此动图导出为 `gif` 格式的图片。

`with_slider_draw` 命令的前两个参数是用于控制绘图的变量名称和变量值的列表，接下来的参数与 `wxdraw2d` 命令的参数相同：

```
with_slider_draw(  
  f,[1,2,3,4,5,6,7,10],  
  title=concat("f=",f,"Hz"),  
  explicit(  
    sin(2*%pi*f*x),  
    x,0,1  
  ),grid=true  
);
```

对三维绘图而言，类似的功能由命令 `with_slider_draw3d` 提供，可用参数来旋转三维图形：

```
wxanimate_autoplay:true;  
wxanimate_framerate:20;  
with_slider_draw3d(  
  a,makelist(i,i,1,360,3),  
  title=sconcat("a=",a),  
  surface_hide=true,  
  contour=both,  
  view=[60,a],  
  explicit(  
    sin(x)*sin(y),  
    x,-pi,pi,  
    y,-pi,pi  
  )  
)$
```

如果重要的是图的大致形状，那么只要稍微移动一点，就可以直观地看出它的三维特征：

```
wxanimate_autoplay:true;
wxanimate_framerate:20;
with_slider_draw3d(
  t,makelist(i,i,0,2*π,.05*π),
  title=sconcat("t=",t),
  surface_hide=true,
  contour=both,
  view=[60,30+5*sin(t)],
  explicit(
    sin(x)*y^2,
    x,-2*π,2*π,
    y,-2*π,2*π
  )
)$
```

对于更熟悉 plot 而不是 draw 的用户来说，还有一组函数：

- with\_slider;
- wxanimate。

在通常情况下，动图会以设置中选择的帧率回放或导出。要设置回放单个动图的速度，可以修改变量 wxanimate\_framerate：

```
wxanimate(a, 10, sin(a*x), [x,-5,5]),
wxanimate_framerate=6$
```

动图函数使用 Maxima 的 makelist 命令，因此存在如下缺陷：只有当滑块变量在绘图表达式中直接可见时，才会将该变量的值替换到表达式中。因此，执行以下例子将显示错误信息：

```
f:sin(a*x);
with_slider_draw(
  a,makelist(i/2,i,1,10),
  title=concat("a=",float(a)),
  grid=true,
  explicit(f,x,0,10)
)$
```

如果修改为明确要求 Maxima 代入滑块变量的值，则修改后的代码可用于绘制动图：

```
f:sin(a*x);
with_slider_draw(
  b,makelist(i/2,i,1,10),
  title=concat("a=",float(b)),
  grid=true,
  explicit(
    subst(a=b,f),
    x,0,10
  )
)$
```

### 2.3.7 同时打开多个绘图窗口

虽然这个 Maxima 特性不是由 wxMaxima 提供的，但是（如果安装的 Maxima 支持此特性）有时使用起来会很方便。以下例子来自 Mario Rodriguez 在 Maxima 邮件列表中的一篇帖子：

```
load(draw);

/* Parabola in window #1 */
draw2d(terminal=[wxt,1],explicit(x^2,x,-1,1));

/* Parabola in window #2 */
draw2d(terminal=[wxt,2],explicit(x^2,x,-1,1));

/* Paraboloid in window #3 */
draw3d(terminal=[wxt,3],explicit(x^2+y^2,x,-1,1,y,-1,1));
```

在同一个窗口中绘制这些图形也是可行的：

```
wxdraw(
  gr2d(
    key="sin (x)",grid=[2,2],
    explicit(sin(x),x,0,2*pi)),
  gr2d(
    key="cos (x)",grid=[2,2],
    explicit(cos(x),x,0,2*pi))
);
```

### 2.3.8 “用 draw 函数绘图” 侧边栏

“用 draw 函数绘图” 侧边栏暗含一个简单的代码生成器，使用它可以利用 Maxima 的 draw 程序包方便灵活地生成一些图形绘制命令模板。

#### 2.3.8.1 二维图形

提供绘制二维图形的 draw() 命令的简单模板。然后可以使用具体的绘图命令填充该模板，例如利用“二维图形”按钮下方的其他按钮导入绘图命令。

One helpful feature of the 2D button is that it allows to setup the scene as an animation in which a variable (by default it is t) has a different value in each frame: Often a moving 2D plot allows easier interpretation than the same data in a non-moving 3D one.

#### 2.3.8.2 三维图形

提供绘制三维图形的 draw() 命令的简单模板。如果没有主动使用此功能和上一功能，则所有其他按钮都将自动提前调用上一功能中的绘制二维图形模板。

#### 2.3.8.3 表达式

在光标当前所在的 `draw()` 命令中添加绘图函数表达式, 如 `sin(x)`, `x*sin(x)` 或者 `x^2+2*x-4` 等。每个图中可以绘制任意数量的图形。如果没有 `draw()` 命令, 将调用绘制二维图形的模板。

#### 2.3.8.4 隐式图形

尝试寻找满足  $y=\sin(x)$ ,  $y*\sin(x)=3$  或  $x^2+y^2=4$  之类的表达式的所有解, 并在光标当前所在的 `draw()` 命令中绘制对应曲线。如果没有 `draw()` 命令, 将调用绘制二维图形的模板。

#### 2.3.8.5 参数式图形

将变量 (如  $t$ ) 从下限变化到上限, 并在当前绘图命令中使用两个表达式 (如  $t*\sin(t)$  和  $t*\cos(t)$ ), 以生成曲线上点的横、纵坐标  $(x,y)$  (在三维图形中, 第三个表达式生成竖坐标  $z$ )。

#### 2.3.8.6 点

绘制多个点, 并且可以选择顺次连接它们。点的坐标取自列表的列表、二维数组或为每个轴指定的一个列表或数组。

#### 2.3.8.7 图形标题

在图形上方绘制标题。

#### 2.3.8.8 坐标轴

设置坐标轴。

#### 2.3.8.9 等高线

(仅适用于三维绘图) 将类似于山地图中的等高线添加到当前 `draw()` 命令绘制的三维图形中, 或图形在坐标面的二维投影上。另外, 此向导也可以用来不绘制曲面而只显示等高线。

#### 2.3.8.10 图形名称

在图例中增加下一个图形对应的图例项。空名称将阻止为下一个图形生成图例项。

#### 2.3.8.11 线条颜色

为当前 `draw()` 命令所绘图形设置线条颜色。

#### 2.3.8.12 填充颜色

为当前 `draw()` 命令所绘图形设置填充颜色。

#### 2.3.8.13 网格

弹出设置网格线的向导。

### 2.3.8.14 精度

允许选择适当数目的点，以兼顾速度与精度（任何绘图程序都会权衡两者）。

## 2.4 嵌入图片

如果使用的是 .wxmx 文件格式，则可以通过拖放将图片文件嵌入 wxMaxima 文档中。但有时（例如某个图像的内容可能在稍后发生更改）最好指明在计算时加载图像：

```
show_image("man.png");
```

## 2.5 启动时读取的文件

在 wxMaxima 的设置对话框中可以编辑两个文件的命令，这些命令在启动时被执行：

- 文件 maxima-init.mac 中包含每次启动 Maxima 时被执行的命令；
- 文件 wxmaxima-init.mac 中包含每次 wxMaxima 调用 Maxima 时被执行的命令。

这些文件位于用户的主目录或配置文件目录中的 Maxima 用户目录（在 Windows 系统中通常为 maxima，在其他系统中为 .maxima）。使用命令 maxima\_userdir; 可以确定该目录。

## 2.6 以 wx 开头的特殊变量

- wxsubscripts：告诉 Maxima 是否应将包含下划线的变量名（例如 R\_150）转换为带下标的变量名。有关自动转换哪些变量名的详细信息，请参见 wxdeclare\_subscript。
- wxfilename：此变量包含当前在 wxMaxima 中打开的文件的名称。
- wxplot\_pngcairo：表明 wxMaxima 是否尝试使用 gnuplot 的 pngcairo 功能，以提供更多的线条样式和更好的整体图形质量。
- wxplot\_size：定义嵌入绘图的分辨率。
- wxchangedir：在大多数操作系统上，wxMaxima 自动将 Maxima 的工作目录设置为当前文件的目录。这允许文件 I/O（例如通过 read\_matrix）工作，而无需指定必须读或写的文件的整个路径。在 Windows 操作系统上，此功能有时会导致错误消息，因此可以在设置对话框中将其设置为 false。
- wxanimate\_framerate：下一个动图回放的帧率。
- wxanimate\_autoplay：是否在默认情况下自动播放动图。

## 2.7 输出美观的二维公式

函数 table\_form() 显示的二维列表比 Maxima 的默认输出更具可读性。该函数的输入是一个列表，其中含一个或多个列表。与 print 命令类似，此命令即使以符号 \$ 结尾也会显示输出。以分号 ; 结束命令时，除了得到列表，还会有“done”语句。

```
table_form(  
  [  
    [1,2],  
    [3,4]  
  ]  
)$
```

正如下一个例子所示，可以在执行 `table_form` 命令之前创建需要显示的列表。

```
(%i9)      titleList:["1st value", "2nd value", "3rd value"];
           xList : makelist(x,x,1,3);
           xsqList : makelist(x^2,x,1,3);
           table_form([ titleList,xList,xsqList ])$

(titleList) [1st value, 2nd value, 3rd value]
(titleList) [1, 2, 3]
(titleList) [1, 4, 9]
           1st value 2nd value 3rd value
(titleList) 1         2         3
           1         4         9
```

Figure 10: 一个三行表的例子

此外，由于矩阵是含列表的列表，它也可以用类似的方式转换为表格。

## 2.8 错误 (bug) 报告

wxMaxima 提供了一些函数，用于收集当前系统的错误报告信息：

- `wxbuild_info()` 收集当前运行的 wxMaxima 的版本信息；
- `wxbug_report()` 说明如何以及在何处提交错误报告。

## 2.9 让输出显示为红色

Maxima 的 `box()` 命令使 wxMaxima 以红色输出其参数。

---

# 3 疑难解答

## 3.1 无法连接到 Maxima

由于 Maxima (进行实际数学运算的程序) 和 wxMaxima (提供易于使用的用户界面的程序) 是通过本地网络连接进行通信的独立程序，因此，最有可能的原因是这种连接不起作用。例如，可能设置了防火墙不仅可以阻止来自互联网的未经授权的连接 (也许还会拦截一些通向互联网的连接)，而且可以阻止同一台计算机内部的进程间通信。注意，由于 Maxima 由 Lisp 语言处理程序运行，因此被阻止的通信进程不一定命名为“Maxima”。此时打开网络连接的程序名称可能是 `sbcl`、`gcl`、`ccl`、`lisp.exe` 或者类似的名字。

在 Unix 系统上，另一个可能的原因是在同一台计算机上的两个程序之间提供网络连接的环回 (loopback) 网络配置不正确。

```
(%i17) M : matrix(titleList,xList,xsqList);
      table_form(M)$
```

$$(M) \begin{bmatrix} 1st\ value & 2nd\ value & 3rd\ value \\ 1 & 2 & 3 \\ 1 & 4 & 9 \end{bmatrix}$$

```
      1st value 2nd value 3rd value
(%t17)      1      2      3
           1      4      9
```

Figure 11: 另一个 table\_form 的例子

### 3.2 从损坏的.wmxm 文件中取回信息的方法

大多数现代的、基于 XML 格式的文件其实都是普通的压缩 (zip) 文件。wxMaxima 没有启用压缩，因此可以在任何文本编辑器中查看 .wmxm 文件的内容。

如果文件末尾的 zip 签名仍然完整，那么将损坏的 .wmxm 文件重命名为 .zip 文件后，大多数操作系统都提供了提取存储在其中的信息的方法。当需要从该文件中恢复原始图像文件时，可以执行此操作。当 zip 签名不完整时，也不是毫无办法：如果 wxMaxima 在保存文件时检测到错误，那么还会有一个 .wmxm~ 文件，其内容可能会有所帮助。

即使没有 .wmxm~ 文件，.wmxm 文件中 XML 部分是未压缩的，可以将该文件重命名为 .txt 文件，并使用文本编辑器恢复文件中的 XML 部分（以 <?xml version="1.0" encoding="UTF-8"?> 开头，并以 </wxMaximaDocument> 结尾（在 XML 部分的前后，可在文本编辑器中看到一些不可读的二进制内容）。

如果将只包含 XML 部分的文本文件（例如，将这一部分复制并粘贴到新文件中）保存为以 .xml 结尾的文件，则 wxMaxima 可以从中恢复原始文档的文本。

### 3.3 在命令完成前显示一些调试信息

通常 wxMaxima 会等待最后的公式形成后才开始输出，这节省了多次尝试输出部分公式的时间。不过，disp() 命令可以立即输出调试信息，而不必等待当前的 Maxima 命令完成：

```
for i:1 thru 10 do (
  disp(i),
  /* (sleep n) 是可接在字符"?" 后使用的 Lisp 函数 */
  /* 它推迟了程序的运行，此处是 3 秒 */
  ?sleep(3)
```

)\$

### 3.4 绘图结果是空的绘图框和错误信息

这意味着 wxMaxima 无法读取由 Maxima 指示 gnuplot 创建的文件。

这个错误的可能原因如下：

- 绘图命令是第三程序包（如 `implicit_plot`）的一部分，但在尝试绘图之前，没有调用 Maxima 的 `load()` 命令加载此包。
- Maxima 试图做一些当前版本的 gnuplot 无法理解的事情。此时，在由 Maxima 的变量 `maxima_userdir` 确定的目录中，以 `.gnuplot` 结尾的文件包含了 Maxima 给 gnuplot 的指令。在调试问题时，此文件的内容在大多数情况下都是有帮助的。
- Gnuplot 被设置为使用 `pngcairo` 库以提供抗锯齿功能和更多线条样式，但是在编译时没有开启此项功能支持。解决方案：取消选中设置对话框中的“使用 `cairo` 终端进行绘图”复选框，并且不要在 Maxima 中将 `wxplot_pngcairo` 设置为 `true`。
- Gnuplot 没有生成有效的 `.png` 文件。

### 3.5 绘制动图时得到“error: undefined variable（错误：未定义的变量）”

在默认情况下，滑块变量的值需要被替换到要绘制图形的表达式中，即该变量在表达式中可见。使用 `subst()` 命令将滑块变量替换到要绘制图形的方程中可以解决此问题。在工作簿中嵌入动图一节的末尾有一个例子。

### 3.6 删除了一个单元格，而撤销操作不起作用

对单元格的操作和单元格内部的更改均有单独的撤销函数，因此发生这种情况的可能性很低。如果确实是这样，有如下方法可以恢复数据：

- wxMaxima 实际上有两个撤销功能：全局撤销（如果未选择单元格）和单元格内撤销（如果光标在单元格内）。可以尝试使用这两个撤销选项，以查看是否可以恢复内容。
- 如果有办法确定 Maxima 分配给该单元格的标签，只需键入此标签，单元格的内容就会重新出现。
- 如果忘了单元格的标签，也无需惊慌。在“查看”菜单中，可以勾选显示“历史命令”侧边栏，其中显示了最近使用的所有 Maxima 命令。
- 如果以上都没有帮助，可以调用 Maxima 的回放功能：

```
autoplay();
```

### 3.7 wxMaxima 启动时显示信息“Maxima process terminated（Maxima 进程已终止）。”

一个可能的原因是，在设置对话框的“Maxima”选项卡中设置的位置找不到 Maxima 程序，因此根本无法运行 Maxima。将路径设置为有效的 Maxima 二进制文件所在文件夹应该可以解决此问题。

### 3.8 Maxima 永远在计算中而不响应输入

理论上讲，有可能是 wxMaxima 没有意识到 Maxima 已经完成了计算，因此没有得到通知，以告知它可以向 Maxima 发送新的数据。如果是这种情况，重新求值可能会使两个程序同步。

### 3.9 基于 SBCL 的 Maxima 耗尽内存

默认情况下，Lisp 编译器 SBCL 有一个内存限制，以使得它能在低端计算机上运行。当编译一个大型软件包（如 Lapack）或处理特别大的列表和方程时，该限制值可能太低。为了扩展内存限制，可以向 SBCL 提供命令行参数 `--dynamic-space-size`，该参数告诉 SBCL 应该使用多少兆字节的内存。32 位 Windows 系统版本的 SBCL 最多可使用 999 兆字节，64 位 Windows 系统版本的 SBCL 可以被设置为使用更大的内存（例如超过编译 Lapack 所需的约 1280 兆字节）。

为 Maxima（因此也为 SBCL）提供命令行参数的一种方法是利用 wxMaxima 设置对话框的“额外的 Maxima 参数（Additional parameters for Maxima）”字段。

### 3.10 在 Ubuntu 系统上，有时输入无响应或响应迟缓

安装软件包 `ibus-gtk` 应该可以解决此问题。详情见 <https://bugs.launchpad.net/ubuntu/+source/wxwidgets3.0/+bug/1421558>。

### 3.11 wxMaxima 在处理希腊字符或元音变音符时停止

如果所使用的 Maxima 基于 SBCL，则必须将以下行添加到 `.sbclrc` 文件中：

```
(setf sb-impl::*default-external-format* :utf-8)
```

`.sbclrc` 文件应放入特定的文件夹。任何基于 SBCL 的 Maxima 如果已经在当前会话中计算了一个单元格，那么在获得如下命令后，都会显示该文件的位置：

```
:lisp (sb-impl::userinit-pathname)
```

### 3.12 绘图

#### 3.12.1 是否可以同时输出和嵌入绘图文件？

工作簿可嵌入 `.png` 文件。wxMaxima 允许用户指定它们的生成位置：

```
wxdraw2d(  
  file_name="test",  
  explicit(sin(x),x,1,10)  
);
```

如果要使用不同的格式，则可以先生成图像，然后再将其导入工作簿：

```
load("draw");  
pngdraw(name,[contents]):=  
(  
  draw(  
    append(  

```

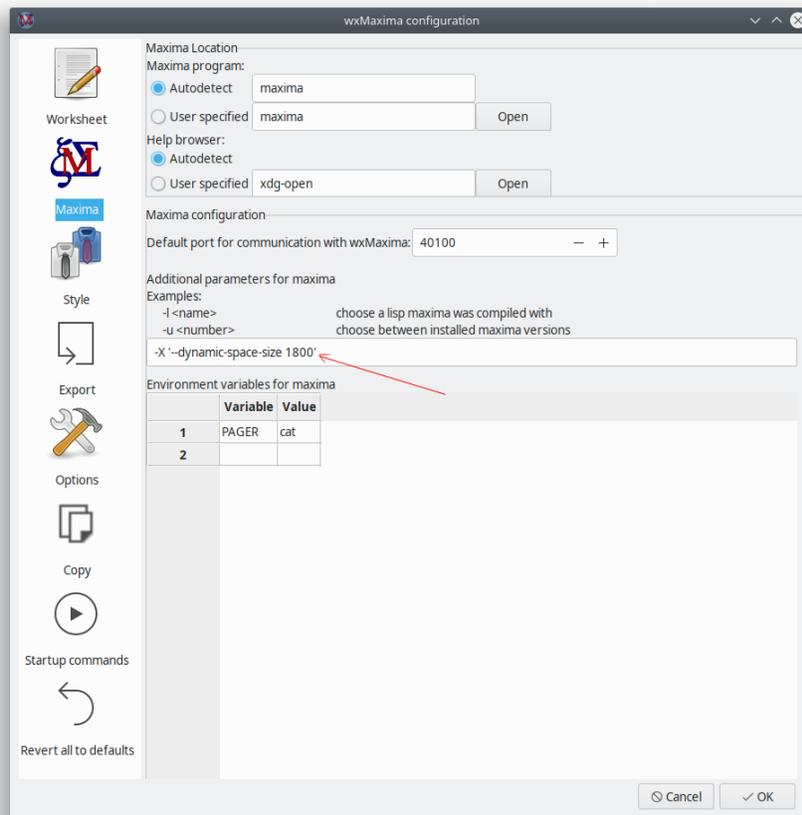


Figure 12: SBCL 内存

```

        [
            terminal=pngcairo,
            dimensions=wxplot_size,
            file_name=name
        ],
        contents
    )
),
show_image(sprintf(false, "~a.png", name))
);
pngdraw2d(name, [contents]) :=
    pngdraw(name, gr2d(contents));

pngdraw2d("Test",
    explicit(sin(x), x, 1, 10)
);

```

### 3.12.2 是否可以指定绘制图形的相对比例？

直接使用 Maxima 无法完成此功能，但是可以利用 `gnuplot` 命令：

```

wxdraw2d(
    proportional_axis=xy,
    explicit(sin(x), x, 1, 10)
), wxplot_size=[1000, 1000];

```

## 4 常见提问 (FAQ)

### 4.1 有没有办法在 LaTeX 页面中显示更多的文字？

Yes. Use the LaTeX package “geometry” to specify the size of the borders.

You can add the following line to the LaTeX preamble (for example by using the respective field in the config dialogue (“Export”->“Additional lines for the TeX preamble”), to set borders of 1cm):

```
\usepackage[left=1cm,right=1cm,top=1cm,bottom=1cm]{geometry}
```

### 4.2 是否有深色模式？

如果 `wxWidgets` 版本足够新，那么当系统设置了深色主题时，`wxMaxima` 将自动处于深色模式。在默认情况下，工作簿本身具有浅色背景。但也可以进行其他配置：利用菜单“查看->反转工作簿亮度”，可以快速将工作簿在浅色模式和深色模式中转换。

### 4.3 wxMaxima 有时在启动后的一段时间内无响应

wxMaxima 将一些大型任务（如解析 Maxima 的超过 1000 页的手册）委托给后台任务，而这些任务通常不会被执行。然而，当需要执行这样一个任务时，可能需要等待几秒钟才能继续工作。

---

## 5 命令行参数

在大多数操作系统中，wxMaxima 都有比运行命令行更简单的启动方式，所以用到命令行参数的可能性很小。不过，wxMaxima 仍然提供了一些命令行选项。

- `-v` 或 `--version`：输出版本信息。
- `-h` 或 `--help`：输出简短的帮助文本。
- `-o` 或 `--open=<str>`：打开作为此命令行参数的文件名。
- `-e` 或 `--eval`：打开文件后对公式单元格求值。
- `-b` 或 `--batch`：如果用命令行打开一个文件，则计算该文件中的所有单元格，然后保存文件。例如，如果文件中描述的会话使 Maxima 生成输出文件，则这一参数会很有用。如果 wxMaxima 检测到 Maxima 输出了错误信息，则批处理将停止；如果检测到 Maxima 有疑问，则批处理将暂停。Maxima 的计算过程有一定的交互性，因此无法保证完全无交互的批处理。
- `--logtostdout`：将所有“调试信息”侧边栏里的信息也输出到标准错误文件（`stderr`）。
- `--pipe`：将 Maxima 的信息通过管道（`pipe`）输出到标准输出文件（`stdout`）。
- `--exit-on-error`：在遇到任何 Maxima 错误时关闭程序。
- `-f` 或 `--ini=<str>`：指定要使用的初始设定文件。
- `-u` 或 `--use-version=<str>`：指定要使用的 Maxima 的版本。
- `-l` 或 `--lisp=<str>`：指定 Maxima 使用的 Lisp 解释器。
- `-X` 或 `--extra-args=<str>`：指定额外的 Maxima 参数。
- `-m` 或 `--maxima=<str>`：指定 Maxima 程序的位置。
- `--enableipc`：使 Maxima 通过内部进程通讯来控制 wxMaxima（应谨慎使用此选项）。

一些操作系统可能在命令行选项前面使用连字符（`-`，dash），而不是减号。