

wxMaxima

Contents

El manual del usuario de wxMaxima	3
1 Introducción de wxMaxima	4
1.1 <i>Maxima</i> y wxMaxima	4
1.1.1 <i>Maxima</i>	4
1.1.2 wxMaxima	5
1.2 Cuaderno básico	5
1.2.1 La aproximación del cuaderno	6
1.2.2 Celdas	7
1.2.3 Cursores horizontales y verticales	8
1.2.4 Envía celdas a “Maxima”	8
1.2.5 Auto-completado de instrucción	8
1.2.6 Paneles Laterales	11
1.2.7 Salida MathML	13
1.2.8 Admite Markdown	13
1.2.9 Teclas resaltadas	13
1.2.10 TeX crudo dentro de la exportación TeX	14
1.3 Formatos de Ficheros	14
1.3.1 .mac	14
1.3.2 .wxm	14
1.3.3 .wxmx	14
1.4 Opciones de configuración	16
1.4.1 Tasa predet. del marco de animación	16
1.4.2 Tamaño de trama predet. para sesiones <i>maxima</i> nuevas .	17
1.4.3 Coincide paréntesis en controles de texto	17
1.4.4 No guardar el cuaderno automáticamente	17
1.4.5 ¿Donde está guardada la configuración?	17
2 Extensiones para <i>Maxima</i>	18
2.1 Variables subscritas	18
2.2 Retroalimentación del usuario en la barra de estado	18
2.3 Tramado	19
2.3.1 Empotrando una trama dentro de la hoja de trabajo . . .	19
2.3.2 Creando tramas empotradas mayores o menores	19

2.3.3	Tramados de mayor calidad	20
2.3.4	Abriendo tramados empotrados en ventanas <i>gnuplot</i> interactivas	20
2.3.5	Abriendo consola para instrucciones de <i>gnuplot</i> en las ventanas plot	20
2.3.6	Empotrando animaciones dentro de la hoja de cálculo . .	20
2.3.7	Abriendo múltiples tramas en ventanas contemporáneas .	22
2.3.8	El panel lateral «Tramado utilizando dibujo»	23
2.4	Gráficos empotrados	25
2.5	Ficheros de inicio	25
2.6	Variables especiales <i>wx</i>	25
2.7	Salida de 2D con impresión bonita	26
2.8	Comunicar gazapo	27
2.9	Marcando que la salida será dibujada en rojo	27
3	Solución de problemas	27
3.1	No se puede conectar a <i>Maxima</i>	27
3.2	Cómo guardar datos desde un fichero estropeado <i>.wxmx</i>	28
3.3	Desea algo de información de depuración para ser desplegada sobre la pantalla antes que mi instrucción haya finalizado	28
3.4	El tramado solamente muestra una cobertura de sobre vacío cerrado con un mensaje de error	29
3.5	Tramando un resultado de animación en «error: variable no definida»	29
3.6	He perdido una celda de contesto y al deshacer no lo recuerda . .	30
3.7	El inicio de <i>wxMaxima</i> con el mensaje “Procesado de Maxima terminado.”	30
3.8	‘Maxima’ está calculando para siempre y no responde a la entrada	30
3.9	Mi <i>Maxima</i> basada en SBCL se ejecuta sin memoria	30
3.10	La entrada algunas veces el tecleo es vago/perezoso en Ubuntu .	31
3.11	<i>wxMaxima</i> se detiene cuando <i>Maxima</i> procesa caracteres griegos o umlauts	32
3.12	Tramado	32
3.12.1	¿Se puede crear la salida de <i>wxMaxima</i> por ambos fichero de imagen y tramas empotradas a la vez?	32
3.12.2	¿Puede establecer el radio de aspecto de una trama? . . .	33
4	P+F	33
4.1	¿Hay una manera de crear más texto que quepa en una página LaTeX?	33
4.2	¿Hay un modo oscuro?	33
4.3	<i>wxMaxima</i> algunas veces se cuelga durante algunos segundos una vez en el primer minuto	33
5	Argumentos de línea de instrucción	34

El manual del usuario de wxMaxima

wxMaxima es un interfaz gráfico del usuario (IGU) para el sistema algebraico de computación *Maxima* (CAS). wxMaxima permite uno para utilizar todas las funciones de *Maxima*. Además, proporciona asistentes convenientes para acceder a las características más comúnmente utilizadas. Este manual describe alguna de las características que hacen de wxMaxima una de las más popular IGU para *Maxima*.



Figure 1: Logotipo wxMaxima

Antes que aparezca en el contexto unos pocos puntos permaneciendo la navegación están en orden. Inmediatamente debajo hay una tabla pequeña de los contenidos. Pulsando cualquiera de los ítems listados mueve el cursor al techo de la sección indicada del manual. Pulsando en Índice mueve el cursor a un índice del contenido. Esta tabla extendida puede ser utilizada para explorar más directamente en partes específicas del manual. El enlace Contenidos aparece en los lugares a través del texto para propósitos de exploración.

- Introducción:

wxMaxima básicos

- Extensiones:

Las instrucciones wxMaxima agregan a *Maxima*

- Soluciones:

Qué hacer si wxMaxima no funciona como esperaba

- P+F:

Preguntas frecuentes

- Línea de instrucción:

Los argumentos de la línea de instrucción admite wxMaxima

1 Introducción de wxMaxima

1.1 *Maxima* y wxMaxima

En el dominio de código abierto, los sistemas grandes están divididos normalmente en proyectos más pequeños que son más fáciles para manipular en grupos pequeños de desarrolladores. Por ejemplo un programa para el quemado del DVD consistirá de una herramienta de línea de instrucciones que actualmente quema el CD un interfaz gráfico de usuario que permite a los usuarios implementarlo sin tener que aprenderse todas las opciones de línea de instrucciones como complemento “enviar-a-CD” para una aplicación de gestión de ficheros, para la función «quemar a CD» de un reproductor de música y como el grabador de CD para una herramienta de respaldo DVD. Otra ventaja es que dividiendo una tarea grande en partes más pequeñas permite a los desarrolladores proporcionar varios interfaces de usuarios para el mismo programa.

Un sistema de computación algebraico (CAS) como *Maxima* encaja dentro de este marco referencial. Un CAS puede proporcionar la lógica detrás de la aplicación de precisión de cálculo arbitrario o puede hacer transformaciones automáticas de fórmulas en el segundo plano de un sistema más grande (p.e., [Sage] (<https://www.sagemath.org/>)). Alternativamente, puede ser utilizado directamente como un sistema independiente. *Maxima* puede ser accedido vía una línea de instrucción. A menudo, sin embargo, un interfaz como *wxMaxima* proporciona una manera más eficiente para acceder al software, especialmente para los recién llegados.

1.1.1 *Maxima*

Maxima es un sistema algebraico computador completo (CAS) de características. Un CAS es un programa que puede resolver problemas matemáticos reconociendo fórmulas y encontrando una fórmula que resuelva el problema como difícil a ajustar la salida del valor numérico del resultado. En otras palabras, *Maxima* puede servir como una calculadora que proporciona representaciones numéricas de variables, y además puede proporcionar soluciones analíticas. Más aún, ofrece un rango de métodos numéricos para ecuaciones o sistemas de ecuaciones que no pueden ser analíticamente resueltos.

Pantallazo de Maxima, línea intrucción

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 es un interfaz del usuario gráfico que proporciona la completa funcionalidad y flexibilidad de *Maxima*. *wxMaxima* proporciona a los usuarios una pantalla gráfica y varias características que hacen trabajar más fácilmente con *Maxima*. Por ejemplo *wxMaxima* permite uno exportar cualquier contenido de celdas (o, si eso es necesario, cualquier parte de una fórmula, también) como texto, como especificación LaTeX o MathML solo con una pulsación secundaria. Por cierto, un cuaderno completo, puede ser exportado, o bien como un fichero HTML o como un fichero LaTeX. La documentación para *wxMaxima*, incluyendo cuadernos para ilustrar aspectos de su uso, está por conexión al sitio web de *wxMaxima*, así como a través del menú de ayuda.

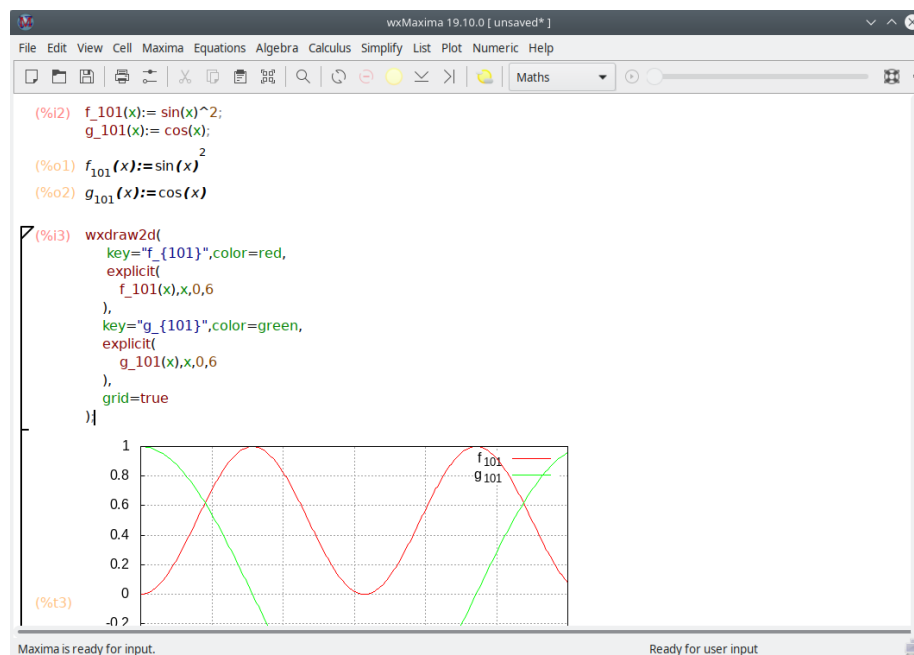


Figure 2: Ventana de wxMaxima

Los cálculos que son introducidos en *wxMaxima* son realizados por la herramienta de línea de instrucciones de *Maxima* en el segundo plano.

1.2 Cuaderno básico

Mucho de *wxMaxima* son auto-explicativos, pero algunos detalles requieren atención. Este sitio contiene un número de cuadernos que dirige varios aspectos de *wxMaxima*. Trabajar junto algunos de estos (particularmente el tutorial de «10 minutos (*wxMaxima*)») incrementará la familiaridad de uno con los ambos del contenido de *Maxima* y el uso de *wxMaxima* para interactuar con *maxima*. Este

manual se concentra en describir aspectos de *wxMaxima* que no son como los evidentes y que tal vez no está cubierto dentro del material por línea.

1.2.1 La aproximación del cuaderno

Una de las muy pocas cosas que no son normalizadas en *wxMaxima* es que se organiza los datos para *Maxima* dentro de celdas que son evaluadas (la cual indica: envía a *maxima*) solamente cuando solicita el usuario esto. Cuando una celda es evaluada, todas las instrucciones dentro de esa celda, y solamente esa celda, son evaluadas como un guión. (El enunciado precedente no es muy preciso: uno puede seleccionar un conjunto de celdas adyacentes y evaluarlas a la vez. Además, una puede instruir a *Maxima* que evalúe todas las celdas dentro de un cuaderno en una pasada). La aproximación de `_wxMaxima` para enviar instrucciones para ejecutar tal vez se siente poco familiar en el primer momento. Sin embargo, drásticamente trabaja fácilmente con documentos grandes (cuando el usuario no desea cada modificación para disparar automáticamente una re-evaluación completa del documento completo). Además, esta aproximación es muy útil para depurar.

Si está tecleado el texto en *wxMaxima* automáticamente crea una celda de hoja de trabajo nueva. El tipo de esta celda puede ser seleccionado en la barra de herramientas. Si se crea una celda de código, la celda puede ser enviada a *Maxima*, la cual causa que el resultado del cálculo es desplegado debajo del código. Un par de dichas instrucciones se muestra debajo.

```
(%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 3: Celda de entrada/salida

En la evaluación de los contenidos de celda de entrada la celda *Maxima* de entrada asigna una etiqueta a la entrada (por defecto muestra en rojo y reconocible por el «%i») por el cual puede ser referenciado posteriormente dentro de la sesión `_wxMaxima`. La salida que genera *Maxima* genera además obtener una etiqueta que comience con «%o» y por defecto está oculta, excepto si el usuario asigna la salida un nombre. En este caso por defecto el etiquetado definido por el usuario está desplegado. El estilo «%o» de etiquetado *Maxima* auto-genera además será accesible, sin embargo.

Al lado de las celdas de entradas *wxMaxima* permite celdas de texto para documentación, celdas de imagen, celdas de título, celdas de capítulo y celdas de

sección. Cada celda tiene su propio búfer para deshacer por lo que se depura a través de modificar los valores de varias celdas y después gradualmente es más fácil revertir las modificaciones no necesarias. Además el mismo cuaderno tiene un búfer global para deshacer que puede deshacer celdas editadas, agregar y borrar.

La figura inferior muestra tipos de celda diferentes (celdas título, sección, subsección, texto, E/S e imagen).

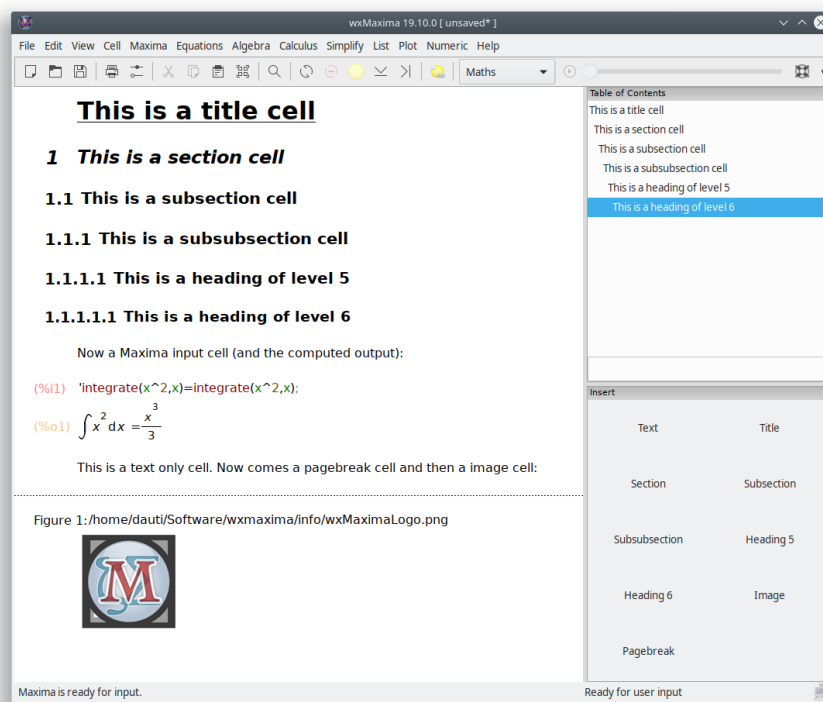


Figure 4: Ejemplo de celdas wxMaxima

1.2.2 Celdas

El cuaderno está organizado dentro de celdas. Cada celda puede contener otras celdas o los tipos de contenidos siguientes:

- una o más líneas *Maxima* de entrada
- una o más imágenes
- salida de, o una pregunta desde, *Maxima*
- un bloque de texto que puede por ejemplo ser utilizado para documentación
- un título, sección o una subsección.

El comportamiento predeterminado de *wxMaxima* cuando el texto es teclado para creación automática de una celda matemática. Las celdas de otros tipos puede ser creadas utilizando el menú Celda, utilizando las teclas resaltadas mostradas dentro del menú o utilizando el listado de arrastrar-bajar dentro de la barra de herramientas. Una vez que la celda no matemática es creada, lo que se teclee dentro del fichero es interpretado como texto.

El texto de los comentarios adicionales pueden introducirse dentro de una celda matemática si está encorchetado tal como sigue `_ /Este comentario no se enviará a Maxima para evaluación/`.

1.2.3 Cursores horizontales y verticales

Si el usuario intenta seccionar una oración completa, un procesador de palabras intentará extender la selección para iniciar y finalizar automáticamente con una palabra frontera. Tal como *WxMaxima* si es seleccionada más de una celda extenderá la selección a todas las celdas completamente.

Qué no es estándar es que *wxMaxima* proporciona flexibilidad para arrastrar-y-soltar definiendo dos tipos de cursores. *wxMaxima* aplicará entre estos automáticamente cuando lo necesite:

- El cursor es dibujado horizontalmente si es trasladado dentro del espacio entre dos celdas o pulsando allí.
- Un cursor vertical que funciona dentro de una celda. Este cursor está activado moviendo el cursor dentro de una celda utilizando el puntero del ratón o las teclas del cursor y funciona como el cursor en un editor de texto.

1.2.4 Envía celdas a “Maxima”

La instrucción dentro de una calda de código son ejecutadas una vez CTRL+INTRO, MAYÚS+INTRO o la tecla INTRO en el teclado numérico es presionado. La *wxMaxima* predeterminada está para introducir instrucciones cuando es introducido o bien CTRL+INTRO o bien MAYÚS+INTRO, pero *wxMaxima* puede ser configurado para ejecutar instrucciones como respuesta al INTRO.

1.2.5 Auto-completado de instrucción

wxMaxima contiene una característica de auto-completado que es disparada a través del menú (Celda/Palabra Completa) o alternativamente pulsando la combinación de teclas CTRL+ESPACIO. El auto-completado distingue MAYÚS-mínus. Por ejemplo si está activado dentro de una especificación unitaria para ezUnits ofrecerá un listado de unidades aplicables.

Al lado de completar un nombre de fichero, un nombre de unidad o las instrucciones actuales o el nombre de variable el auto-completado es capaz de mostrar

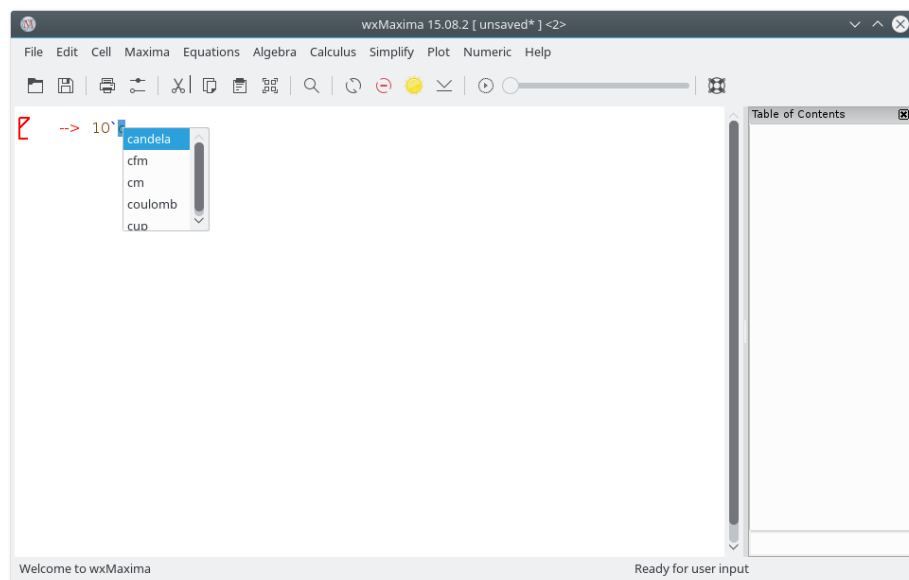


Figure 5: ezUnits

una plantilla para mucho de las instrucciones iniciando el tipo (y seleccionando) de los parámetros de este programa espera. Para activar esta característica presione MAYÚS+CTRL+ESPACIO o seleccione el ítem del menú respectivo (Celda/Mostrar Plantilla).

1.2.5.1 Caracteres griegos

Equipos tradicionalmente almacenados en valores 8-bit. Esto permite para un máximo de 256 caracteres diferentes. Todas las letras, números y símbolos de control (final de transmisión, final de cadena, líneas y bordes para dibujar rectángulos para menús :etc_.) de cerca cualquier idioma puede caber dentro de ese límite.

Para muchos países la página de código de 256 caracteres que han sido elegidos no incluyen cosas como letras griegas, por lo que, eso son frecuentemente utilizadas en matemáticas. Para solucionar este tipo de limitación [Unicode] (<http://home.unicode.org/>) ha sido inventado: Una codificación que hace que el texto inglés funcione como normal, pero para utilizar más de 256 caracteres.

Maxima permite Unicode, si fue compilado utilizando un compilador Lisp que o bien admite Unicode o que no tomen en cuenta acerca de la codificación tipográfica. Al menos una de este par de condiciones debe ser cierta. *wxMaxima* proporciona un método de introducción de caracteres griegos utilizando el teclado:

- Puede introducir una tecla griega pulsando la tecla ESC y después pulsando la tecla del nombre del carácter griego.
- Alternativamente puede ser introducido pulsando ESC, una letra (o dos para la tecla griega omicron) y ESC otra vez. En este caso son admitidas las letras siguientes:

tecla | Letra griega | tecla | letra griega | tecla | letra griega |

|:—:|:—:|:—:|:—:|:—:|:—:| 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 |

El mismo mecanismo también permite introducir algunos símbolos matemáticos adicionales:

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
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

keys to enter	mathematical symbol
subset	subset
notsubse ^t eq	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

Si un símbolo especial no está dentro del listado es posible introducir caracteres Unicode arbitrarios pulsando ESC [número del carácter (hexadecimal)] ESC.

ESC 61 ESC por lo tanto resulta en un α .

Note que muchos de estos símbolos (notablemente excepciones son los símbolos lógicos) no tener un significado especial dentro de *Maxima* y por lo tanto serán interpretados como caracteres ordinarios. Si *Maxima* está compilado utilizando un Lisp que no admite utilizar caracteres Unicode quizá en su lugar cause un mensaje de error.

1.2.6 Paneles Laterales

Enlaces para las instrucciones de *Maxima* más importantes de las cosas como un índice de contenidos, ventanas con mensajes de depuración o un historial de las últimas instrucciones emitidas pueden ser accedidas utilizando los paneles laterales. Puede ser habilitados utilizando el menú de “Vistas”. Todos ellos pueden ser trasladadas a otras localización dentro o fuera de la ventana de `_wxMaxima`. Otro de los paneles útiles es el que permite introducir letras griegas utilizando el ratón.

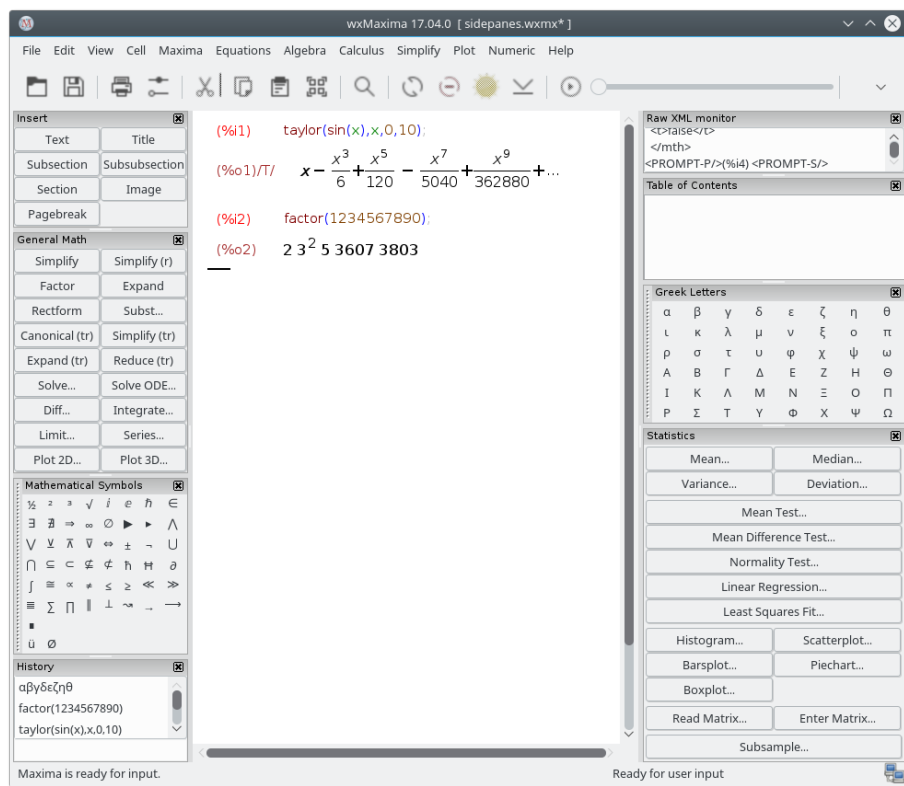


Figure 6: Ejemplo de diferentes paneles laterales

1.2.7 Salida MathML

Varios procesadores de palabras y programas similares o bien reconocen entradas de MathML y lo insertan automáticamente como una ecuación editable 2D - o (como LibreOffice 5.1) tiene un editor de ecuación que ofrece una característica “importar MathML desde portapapeles”. Otros admiten matemáticas RTF. *wxMaxima* por lo tanto ofrece varios apuntes dentro del menú a través de la pulsación secundaria del menú.

1.2.8 Admite Markdown

wxMaxima ofrece un conjunto de convenciones de Markdown Estándar que no colisione con la notación matemática. Una vez que este elemento es un listado de puntos.

Texto ordinario

- * Un ítem, nivel 1 de sangría
- * Otro ítem en nivel 1 de sangría
 - * Un ítem en un nivel de sangría 2
 - * Un segundo ítem en el nivel 2 de sangría
- * Un tercer ítem en el nivel 1 de la sangría

Texto ordinario

wxMaxima reconoce texto iniciando con los caracteres > como entrecomillado del bloque:

Texto ordinario

```
> comilla comilla comilla comilla
> comilla comilla comilla comilla
> comilla comilla comilla comilla
```

Texto ordinario

Las salidas de TeX y HTML de *wxMaxima* también reconocerán => y lo reemplazarán por el signo correspondiente de Unicode:

cogito => sum.

Otros símbolos que el HTML y TeX exportan reconocerán son <= y >= para comparaciones, una flecha doble doble-apuntada (<=>), flecha con cabecera única (<->, -> y <-) y +/- como el signo respectivo. Para salida TeX además son reconocidos << y >>.

1.2.9 Teclas resaltadas

Muchas teclas resaltadas pueden ser encontradas dentro del texto de los respectivos menús. Desde que son tomados actualmente desde el texto del menú y por lo tanto puede ser adaptados por las traducciones de *wxMaxima* para coincidir con las necesidades de los usuarios del teclado local, no lo documentamos aquí. Unas pocas teclas especiales o alias, sin embargo, no están documentadas dentro de los menús:

- CTRL+MAYÚS+SUPR borra una celda completa.
- CTRL+TAB o CTRL+MAYÚS+TAB dispara el mecanismo de auto-completado.
- MAYÚS+ESPACIO introduce un espacio sin ruptura.

1.2.10 TeX crudo dentro de la exportación TeX

Si una celda de texto comienza con **TeX**: la exportación de TeX contiene el texto literal que continua el marcador **TeX**:. Utilizando esta característica permite al apunte del marcado TeX sin el cuaderno *wxMaxima*.

1.3 Formatos de Ficheros

El material desarrollado en una sesión *wxMaxima* puede ser almacenado para un posterior uso en cualquiera de estas tres maneras:

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.

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*.

Puede utilizar los ficheros '.mac' para escribir su propia biblioteca de macros. Pero desde que no contengan suficiente información estructural no puede releer como una sesión *wxMaxima*.

1.3.2 .wxm

Los ficheros .wxm contienen el cuaderno excepto la salida de *Maxima*. En las versiones >5.38 de *Maxima* pueden leerse utilizando la función 'load()' de *Maxima* tan solo como pueden ser ficheros .mac. Con este formato de texto simple algunas veces es inevitable que los cuadernos que utilicen características nuevas no sean compatibles con las versiones más antiguas de *wxMaxima*.

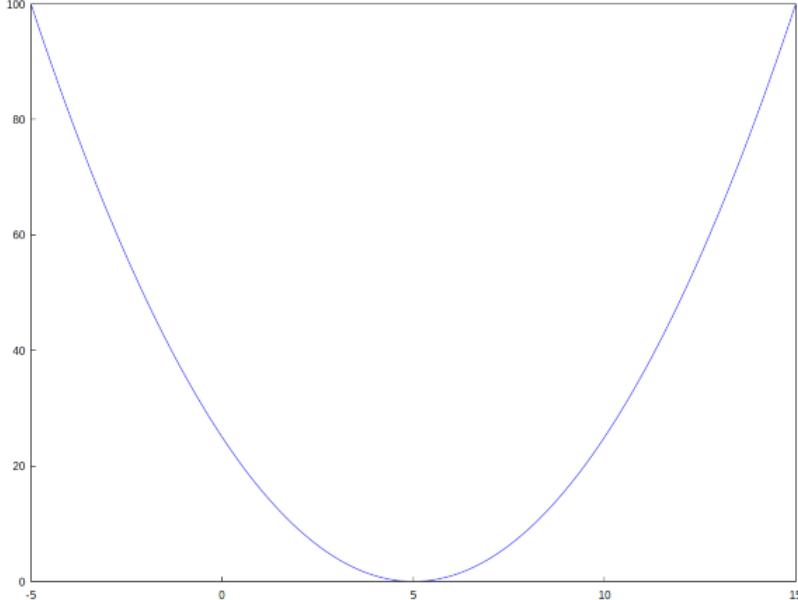
1.3.3 .wxmx

Este fichero cuyo formato está basado en XML guarda el cuaderno completo incluyendo cosas como el factor de zoom y el listado de vigía. Es el formato de fichero preferido.

```

(%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)

```



```

(%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 7: Loading a .mac file with batch()

1.4 Opciones de configuración

Para algunas configuraciones comunes de variables *wxMaxima* ofrecen dos maneras de configurarlas:

- La caja de diálogo de configuración de abajo le permite modificar sus valores predeterminados para las sesiones actuales y subsecuentes.
- Además, los valores para muchas de las variables de configuración pueden ser modificadas para la sesión actual solamente sobrescribiendo sus valores desde la hoja de cálculo, como se muestra debajo.

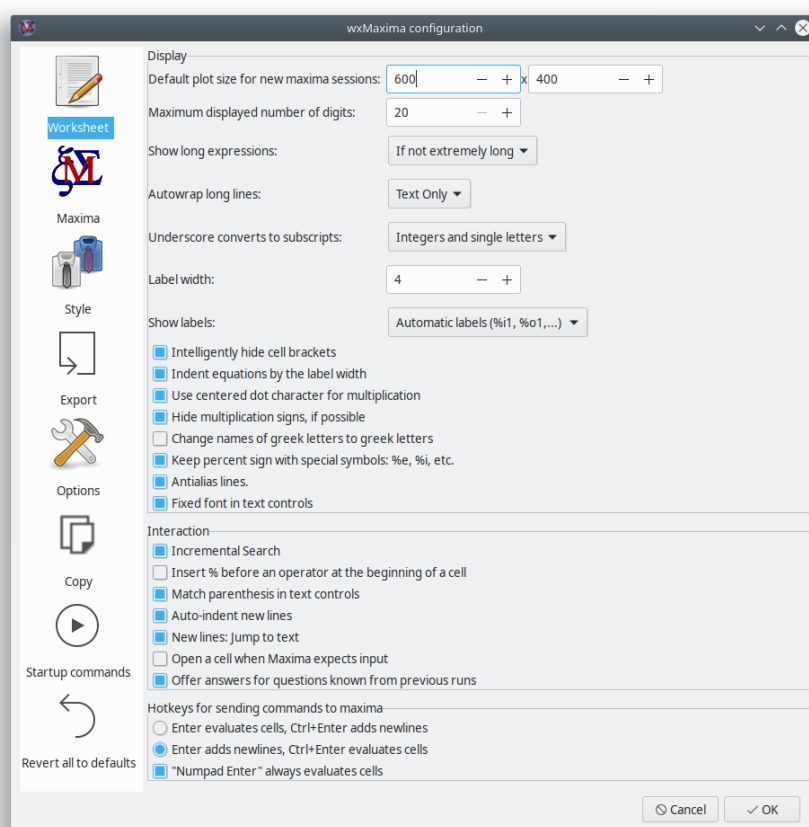


Figure 8: Configuración de wxMaxima 1

1.4.1 Tasa predet. del marco de animación

La tasa del marco de animación que se utiliza para animaciones nuevas se conserva dentro de la variable `wxanimate_framerate`. El valor inicial de esta vari-

able contiene dentro una hoja de trabajo nueva que puede ser modificada utilizando el diálogo de configuración.

1.4.2 Tamaño de trama predet. para sesiones *maxima* nuevas

Tras el siguiente inicio de tramas empotradas a la hoja del cuaderno será creada con este tamaño si el valor de `wxplot_size` no es modificado por *maxima*.

Con la intención de establecer el tamaño de trama de un grafo único solamente utilice la notación siguiente puede ser utilizado que conjuntos de un valor de variable para una instrucción únicamente:

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

1.4.3 Coincide paréntesis en controles de texto

Esta opción habilita dos cosas:

- Si un paréntesis de apertura, corchete o entrecomillado doble es introducido `_wxMaxima` inserta un cierre tras éste.
- Si el texto está seleccionad si cualquiera de estas teclas es pulsada el texto seleccionado será puedes entre los signos coincidan.

1.4.4 No guardar el cuaderno automáticamente

Si esta opción está puesta, el fichero del cuaderno está dentro es sobrescrito solamente al solicitar por el usuario. En caso de un cuelgue/corte de luz/... una copia de respaldo reciente aún está disponible dentro del directorio temporal (/tmp).

Si esta opción no está puesta *wxMaxima* se comporta como un app telefónico moderno:

- Ficheros son guardados automáticamente al salir
- Y el fichero automáticamente será guardado cada 3 minutos.

1.4.5 ¿Donde está guardada la configuración?

Si está utilizando Unix/Linux, la información de la configuración será guardada dentro de un fichero `.wxMaxima` dentro de su directorio personal (si está utilizando `wxWidgets < 3.1.1`), o `.config/wxMaxima.conf` ((XDG-Estandarizado) si `wxWidgets >= 3.1.1` es utilizada). Puede obtener la versión de `wxWidgets` desde la instrucción `wxbuild_info()`; o para utilizar la opción del menú Ayuda > Acerca de. `wxWidgets` es la biblioteca IGU de plataforma cruzada el cual es la

base para *wxMaxima* (por lo tanto el **wx** dentro del nombre). (Cuando el nombre del fichero comience con un punto, **.wxmaxima** o **.config**, serán ocultos).

Si está utilizando Windows, la configuración será almacenada dentro del registro. Encontrará los apuntes para *wxMaxima* en la posición siguiente dentro del registro: HKEY_CURRENT_USER\Software\wxMaxima

2 Extensiones para *Maxima*

wxMaxima es primariamente un interfaz gráfico del usuario para *Maxima*. Por lo tanto su propósito principal es enviar instrucciones a *Maxima* y comunicar los resultados de ejecución de esas instrucciones. En algunos casos, sin embargo, *_wxMaxima* agrega funciones a *Maxima*. la capacidad de *_wxMaxima* para generar informes exportando unos contenidos del cuaderno a HTML y LaTeX han sido mencionados. Esta sección considera algunas maneras que *wxMaxima* mejora la inclusión de gráficos dentro de una sesión.

2.1 Variables subscritas

se se establece ‘wxsubscripts’ a variable cierta los nombres de formato **x_y** son desplegados utilizadn el subguion ‘if’

- ‘y’ es una letra única
- ‘y’ es un entero

Si el nombre de variable no coincide estos requerimientos aún puede ser declarado como “para ser subguionado” utilizando la instrucción `wxdeclare_subscript(variable_name);` o `wxdeclare_subscript([variable_nombre1,variable_nombre2,` declarando una variable como sub-guion puede ser revertido utilizando la instrucción siguiente: `wxdeclare_subscript(variable_nombre,false);`

2.2 Retroalimentación del usuario en la barra de estado

Las instrucciones de ejecución largas pueden proporcionar al usuario su estado dentro de la barra de estado. Esta alimentación está sustituida por cualquier alimentación devuelta que es colocada allí (permitiendo utilizarlo como un indicador de progreso) y es borrado tan pronto como la instrucción actual enviada a *Maxima* ha finalizado. Es seguro de utilizar `wxstatusbar()` incluso dentro de bibliotecas que tal vez son utilizadas con el simple *Maxima* (como opuesto a *wxMaxima*): Si *wxMaxima* no está presente la instrucción `wxstatusbar()` será tan solo dejado sin evaluar.

```
for i:1 thru 10 do (  
  /* Dice al usuario como obtuvo */  
  wxstatusbar(concat("Pass ",i)), disp(i),  
  /* (sleep n) es una función de Lisp, la cual puede utilizarse */
```

```

/* con el carácter "?" antes. Retrasa la ejecución del */
/* programa (aquí: por 3 segundos) */
?sleep(3)
)$

```

2.3 Tramado

Tramar (teniendo fundamentalmente hacerlo con gráficas) es un lugar donde un interfaz gráfico del usuario tendrá para proporcionar algunas extensiones para el programa original.

2.3.1 Empotrando una trama dentro de la hoja de trabajo

Maxima normalmente instrumenta el programa *gnuplot* externo para abrir una ventana separada para cada diagrama que crea. Debido a que muchas veces es conveniente empotrar a la hoja de trabajo en vez que *wxMaxima* proporcione su propio conjunto de funciones de trama que no difieran desde las funciones correspondientes a *maxima* para guardar dentro de su nombre: está todo prefijado por un “wx”. Por ejemplo `wxplot2d` se corresponde a `plot2d`, `wxplot3d` se corresponde a `plot3d`, `wxdraw` se corresponde a `draw` y `wxhistogram` se corresponde a `histogram`.

2.3.2 Creando tramas empotradas mayores o menores

Tal como notó antes, el dialogo de configuración proporciona una manera de modificar el tamaño de tramas predeterminada que son creadas con la cual establece el valor de inicialización de `wxplot_size`. Las rutinas de tramado de *wxMaxima* respeta esta variable que especifica el tamaño de una trama dentro de píxeles. Puede siempre ser solicitado o utilizado para establecer el tamaño de las tramas siguientes:

```

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

```

Si el tamaño de solamente una trama va a ser modificada, *Maxima* proporciona una manera canónica para modificar un atributo solamente para la celda actual. En este uso la especificación `wxplot_size = [valor1, valor2]` es agregada a la instrucción `wxdraw2d()`, y no es parte de la instrucción `wxdraw2d`.

```

wxdraw2d(
    explicit(
        sin(x),
        x,1,10
    )
)$

```

```
)
),wxplot_size=[1600,800]$
```

2.3.3 Tramados de mayor calidad

Gnuplot parece no proporcionar una manera portable para determinar si admite bitmap de salida de alta calidad que la biblioteca *cairo* proporciona. En los sistemas donde está compilado *gnuplot* para utilizar esta biblioteca la opción `pngcairo` desde el menú de configuración (que puede ser sobrescrito por la variable `wxplot_pngcairo`) habilita el mantenimiento para antialias y estilos de línea adicional. Si `wxplot_pngcairo` está establecido sin *gnuplot* admitiendo esto el resultado será mensajes de error en vez de gráficos.

2.3.4 Abriendo tramados empotrados en ventanas *gnuplot* interactivas

Si una trama fue generada utilizando las instrucciones de tipo `wxdraw` (`wxplot2d` y `wxplot3d` no son admitidas por esta característica) y el tamaño del fichero del proyecto *gnuplot* no es una forma *wxMaxima* buena para ofrecer un menú de pulsación secundaria que permite abrir el tramado dentro de una ventana interactiva de *gnuplot*.

2.3.5 Abriendo consola para instrucciones de *gnuplot* en las ventanas *plot*

En MS Windows, si dentro de la variable de *Maxima* `gnuplot_command` “gnuplot” es reemplazado por “wgnuplot”, *gnuplot* ofrece la posibilidad de abrir una ventana de la consola, donde las instrucciones de *gnuplot* pueden ser introducidas. Desafortunadamente, habilitando esta característica causa que *gnuplot* “roba” el foco del teclado por un breve tiempo mientras se prepara un tramado.

2.3.6 Empotrando animaciones dentro de la hoja de cálculo

Diagramas 3D tienden a ser difíciles de leer para datos cuantitativos. Una alternativa viable tal vez es asignar el 3er parámetro a la rueda del ratón. La instrucción `with_slider_draw` es una versión de `wxdraw2d` que prepara múltiples tramados y permite cambiar entre ellos moviendo el arrastre en la cima de la pantalla. *wxMaxima* permite exportar esta animación como un gif animado.

Los primeros dos argumentos para `with_slider_draw` son el nombre de la variable que está pasada entre las tramas y un listado de los valores de estas variables. Los argumentos que siguen son los argumentos ordinarios para `wxdraw2d`:

```
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
);

```

La misma funcionalidad para tramas de 3D es accesible como ‘with_slider:draw3d’, el cual permite rotación en tramas de 3D:

```

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

```

Si el formato general de la trama es que pasa quizá sufra mover el tramado tan solo un pequeño bit con el fin de crear su naturaleza 3D disponible a la intuición:

```

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

```

Para aquello más familiar con **trama** en vez de **dibujo** hay un conjunto secundario de funciones:

- ‘con_arrastre’ y
- wxanimate.

Normalmente las animaciones son retro-reproducidas o exportadas con la parte de marco elegida dentro de la configuración de *wxMaxima*. Para establecer la

velocidad se reproduce una animación individual de vuelta a la variable «wxanimate_framerate» puede utilizarse:

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

Las funciones de animación utilizan las instrucciones `makelist` de *Maxima* y por lo tanto comparte el escollo que el valor de variable del arrastre se sustituye dentro de la expresión solamente si la variable es visible directamente dentro de la expresión. Por lo tanto el siguiente ejemplo fallará:

```
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)
)$
```

Si *Maxima* se solicita explícitamente para sustituir el valor del tramado deslizante funciona bien en su lugar:

```
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 Abriendo múltiples tramas en ventanas contemporáneas

Mientras no sean proporcionados por *wxMaxima* esta característica de *Maxima* (sobre configuraciones que la admitan) algunas veces vienen manualmente. Los siguientes ejemplos vienen desde una carta desde Mario Rodríguez al listado de correo de *Maxima*:

```
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));
```

Es posible tramar múltiples tramas dentro de la misma ventana, también:

```
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 El panel lateral «Tramado utilizando dibujo»

La barra lateral «Tramado utilizando dibujo» oculta un generador de código simple que permite generar escenas que hagan uso de alguno de la flexibilidad del paquete *draw* con el que *maxima* viene.

2.3.8.1 2D

Genera el esqueleto de una instrucción `draw()` que dibuja una escena 2D. Esta escena posteriormente tiene que ser filtrada con instrucciones que generan el contenido de la escena, por ejemplo utilizando los botones dentro de las filas debajo del botón “2D”.

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 3D

Genera el esqueleto de una instrucción `draw()` que dibuja una escena 3D. Si ni una escena 2D o 3D está configurada todo de la configuración de los otros botones en una escena 2D que contiene el botón que genera.

2.3.8.3 Expresión

Adjunta una trama común de una expresión como `sin(x)`, `x*sin(x)` o `x^2+2*x-4` para la instrucción `draw()` el cursor actualmente está dentro. Si no hay ninguna instrucción de dibujo en una escena 2D con la trama es generada. Cada escena puede ser rellenada con cualquier número de tramas.

2.3.8.4 Tramado implícito

Intenta encontrar todos los puntos como una expresión `y=sin(x)`, `y*sin(x)=3` o `x^2+y^2=4` es verdadero y trama de la curva resultante dentro de la instrucción

`draw()` el cursor actualmente está dentro. Si no hay instrucción de dibujo en escena de 2D con el trama es generado.

2.3.8.5 Trama paramétrica

Para un paso de una variable desde un límite inferior a un límite superior y utiliza dos expresiones como `t*sin(t)` y `t*cos(t)` para generar coordenadas x, y (y en tramas 3D también z) de una curva que es puesta dentro de la instrucción de actual del dibujo.

2.3.8.6 Puntos

Dibuja muchos puntos que pueden ser unidos opcionalmente. Las coordenadas de los puntos son tomados desde un listado de listas, una unimatriz 2D o un listado o unimatriz por cada eje.

2.3.8.7 Título del diagrama

Dibuja un título sobre el final superior del diagrama,

2.3.8.8 Eje

Establece los ejes.

2.3.8.9 Contorno

(Únicamente para tramas 3D): Agrega líneas de contorna similar a los que puedan definir dentro de una asociación de una montaña a la trama de instrucciones que continúe dentro de la instrucción actual de `'draw()'` y/o el plano de tierra del diagrama. Alternativamente este asistente permite descartar el dibujo de las curvas completamente solamente mostrando la trama del contorno.

2.3.8.10 Nombre de trama

Agrega una leyenda de apunte mostrando el nombre del siguiente tramado de la leyenda del diagrama. Un nombre vacío deshabilita generando apuntes de leyenda para las tramas siguientes.

2.3.8.11 Color de línea

Establece el color de línea para las tramas seguidas que contiene la instrucción de dibujo actual.

2.3.8.12 Color de relleno

Establece un color de relleno para tramas seguidas que contiene la instrucción de dibujo actual.

2.3.8.13 Rejilla

Aparece un asistente que permite establecer las líneas de rejilla.

2.3.8.14 Precisión

Permite seleccionar un punto adecuado dentro de la velocidad versus exactitud compensada que es parte de cualquier programa de trama.

2.4 Gráficos empotrados

Si el formato de fichero `.wxmx` está siendo utilizado empotrando dichos dentro de un proyecto *wxMaxima* puede ser hecho como fácilmente como por arrastrar-y-soltar. Sin embargo algunas veces (por ejemplo si el contenido de la imagen es modificado posteriormente sobre una sesión) es mejor decirle al fichero que cargue la imagen al evaluar:

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

2.5 Ficheros de inicio

El diálogo de configuración de *wxMaxima* ofrece editar dos ficheros con instrucciones que son ejecutadas al inicializar:

- Un fichero que contenga instrucciones que estén ejecutadas al iniciar *Maxima*: «maxima-init.mac»
- un fichero de instrucciones adicionales que son ejecutadas si *wxMaxima* está iniciando *Maxima*: «wxmaxima-init.mac»

Estos ficheros están dentro del directorio del usuario de Maxima (usualmente *maxima* en Windows, *.maxima* en otros sitios) dentro del directorio inicial del usuario / perfil. La ubicación puede ser encontrada fuera con la instrucción: `maxima_userdir`;

2.6 Variables especiales wx...

- `wxsubscripts` le dice a *Maxima* si debería convertir los nombres de variables que contengan un guión bajo (`'R_150'` o lo similar) en variables sub-guion. Vea `wxdeclare_subscript` para detalles los cuales nombres de variables son convertidos automáticamente.
- `wxfilename`: esta variable contiene el nombre del fichero actualmente abierto en *wxMaxima*.
- `wxplot_pngcairo` indica si *wxMaxima* intenta utilizar el terminal de png-cairo *gnuplot* que proporciona más estilos de línea y un mejor calidad gráfica sobre todo.
- `wxplot_size` define la resolución de tramas empotradas.
- `wxchangedir`: en la mayoría de los sistemas *wxMaxima* establece automáticamente el directorio de trabajo de *Maxima* al directorio del fichero actual.

Esto permite la E/S del fichero (p.ej. a través de `read_matrix`) para trabajar sin especificar la ruta completa del fichero que tiene que ser leído o escrito. En Windows algunas veces esta característica causa mensajes de error y por lo tanto puede ser puesta a `false` desde el dialogo de configuración.

- `wxanimate_framerate`: El número de marcos por segundo de las siguientes animaciones con las que tienen que ser representadas.
- `wxanimate_autoplay`: Reproduce automáticamente animaciones predefinidamente?

2.7 Salida de 2D con impresión bonita

La función `table_form()` despliega un listado 2D dentro de un formato que es más legible que la salida desde la rutina de salida predeterminada de *Maxima*, esta instrucción despliega la salida incluso cuando finaliza con un signo de dólar. Finalizando la instrucción con un resultado punto y como dentro de la misma tabla a lo largo con un enunciado “done”.

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

Como se muestra en el ejemplo siguiente, los listados que son ensamblados por la instrucción ‘table_form’ pueden ser creados antes que la instrucción sea ejecutada.

```
(%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 9: Un tercer ejemplo de tabla

Además, porque una matriz es un listado de listados, las matrices pueden convertirse a tablas de una aparición similar.

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

	<i>1st value</i>	<i>2nd value</i>	<i>3rd value</i>
(M)	1	2	3
	1	4	9

```
(%t17)
```

	<i>1st value</i>	<i>2nd value</i>	<i>3rd value</i>
	1	2	3
	1	4	9

Figure 10: Otro ejemplo de `tabla_formulario`

2.8 Comunicar gazapo

wxMaxima proporciona unas pocas funciones que obtenga información de comunicados de gazapos acerca del sistema actual:

- `wxbuild_info()` obtiene información sobre la versión actualmente en ejecución de *wxMaxima*
- `wxbug_report()` informa cómo y donde archivar los gazapos

2.9 Marcando que la salida será dibujada en rojo

La instrucción de *Maxima's* `box()` causa que *wxMaxima* escriba su argumento con un fondo rojo.

3 Solución de problemas

3.1 No se puede conectar a *Maxima*

Dese *Maxima* (el programa que hace las matemáticas actuales) y *wxMaxima* (proporcionando el interfaz de usuario fácil-de-uso) con programas separados que se comunican por los medios de una conexión de red local. Por lo tanto la causa más probable es que esta conexión es algo que no funciona. Por ejemplo

un cortafuegos pude establecer de una manera que no impida conexiones no autorizadas desde el Internet (y quizá interceptar algunas conexiones para Internet, también), pero también para bloques de comunicación de inter-procesos dentro del mismo computador. Nota que desde *Maxima* es siendo ejecutado por un procesador Lisp la comunicación de procesos que están bloqueados desde unos no necesariamente tienen que ser nombrados “maxima”. Los nombres comunes del programa que abre la conexión de red pudo ser sbcl, gcl, ccl, lisp.exe o nombres similares.

En los equipos Unix otra razón posible podría ser que la res del loopback que proporciona conexiones de la red entre dos programas dentro del mismo equipo no está configurado apropiadamente.

3.2 Cómo guardar datos desde un fichero estropeado .wxmx

Internamente muchos formatos modernos basados en XML están normalmente en ficheros zip. *wxMaxima* no activa la compresión, por lo que los contenidos de los ficheros .wxmx pueden verse en cualquier editor de texto.

Si la firma zip en el final del fichero es aún intacta tras renombrar un fichero .wxmx estropeado para .zip muchos sistemas operativos proporcionarán una manera de extraer cualquier porción de información que está almacenada dentro de esta. Esto puede ser hecho cuando hay la necesidad de recuperar la imagen original de ficheros desde un documento de procesamiento de texto. Si la firma zip no está intacta que do necesita estar el final del mundo: si *wxMaxima* al guardar ha detectado que algo es erróneo habrá un fichero wxmx~ cuyo contenido quizá ayude.

E incluso si no hay un fichero: el fichero .wxmx es un formato de contenedor y la porción es almacenada sin comprimir. Es posible renombrar el fichero .wxmx a un fichero .tct y utilizar un editor de texto para recuperar la porción XML del contenido del fichero (se inicia con `<?xml version="1.0" encoding="UTF-8"?>` y termina con `</wxMaximaDocument>`. Antes y después de ese texto verá algún contenido binario no legible dentro del editor del texto).

Si un fichero de texto conteniendo solamente estos contenidos (p.ej. copiar y pegar este texto dentro de un fichero nuevo) es guardado como un fichero terminando en .xml, *wxMaxima* conocerá cómo recuperar el texto del documento desde éste.

3.3 Desea algo de información de depuración para ser desplegada sobre la pantalla antes que mi instrucción haya finalizado

Normalmente *wxMaxima* espera a la fórmula 2D completa para ser transferida antes que comenzar a configurar el conjunto del tipo. Esto guarda tiempo

para crear muchos intentos para tipos de teclas para solo ecuación parcialmente completada. Hay una instrucción `disp`, por lo que, eso proporcionará salida de depuración inmediatamente y sin esperar a que la instrucción de *Maxima* finalice:

```
for i:1 thru 10 do (
  disp(i),
  /* (sleep n) es una función de Lisp, la cual puede utilizarse */
  /* con el carácter "?" antes. Retrasa la ejecución del */
  /* programa (aquí: por 3 segundos) */
  ?sleep(3)
)$
```

3.4 El tramado solamente muestra una cobertura de sobre vacío cerrado con un mensaje de error

Esto significa que *wxMaxima* no pudo leer el fichero *Maxima* que fue admitido a la instrucción *gnuplot* para crear.

Las posibles razones de este error son:

- La instrucción de tramado es parte de un paquete de tercera parte como 'implicit_plot' pero este paquete no fue cargado por la instrucción 'load()' de *Maxima* porque está intentando puntear.
- *Maxima* intentó hacer algo que la versión actualmente instalada de *gnuplot* no es capaz de entender. En este caso, un fichero terminando en *.gnuplot* localizado en el directorio, el cual la variable `maxima_userdir` está apuntando, contiene las instrucciones desde *Maxima* para *gnuplot*. La mayoría del tiempo, estos contenidos del fichero contiene por lo tanto ayuda cuando se depura el problema.
- Gnuplot fue instruido para utilizar la biblioteca `pngcairo` que proporciona antialias y estilos de líneas adicionales, pero no fue compilado para admitir esta posibilidad. Solución: Desmarque la casilla «Utilizar el terminal cairo para tramas» dentro del diálogo de configuración y no establezca `wxplot_pngcairo` a cierto desde *Maxima*.
- Gnuplot no saca un fichero «.png» válido.

3.5 Tramando un resultado de animación en «error: variable no definida»

El valor de la variable desplazada por defecto está únicamente sustituido dentro de la expresión que está para ser tramado si es visible ahí. Utilizando una instrucción `subst` que sustituya el arrastre variable dentro de la ecuación o para trama de soluciones que resuelvan este problema. Al final de la sección Animaciones empotradas dentro de la hoja de cálculo puede ver un ejemplo.

3.6 He perdido una celda de contesto y al deshacer no lo recuerda

Hay funciones de deshacer separadas para operaciones de celdas y para modificaciones dentro de celdas tales que modifiquen son bajos que este a veces ocurran. Si no hay varios métodos para cubrir datos:

- *wxMaxima* actualmente tiene dos características de deshacer: el búfer de deshacer global que está activo si ninguna celda está seleccionada y un deshacer por celda que está activo si el cursor está dentro de una celda. Es peor intentar utilizar ambas opciones con el fin de ver si un valor anterior aún puede ser accedido.
- Si todavía tiene una manera de encontrar cual etiqueta de *Maxima* ha asignado a la celda tan solo teclee dentro de la etiqueta de la celda y sus contenidos aparecerán.
- Si no: no se asuste. En el menú “Vista” hay una manera de ver un panel histórico que muestra todas las instrucciones de *Maxima* que han sido emitidas recientemente.
- Si nada más ayuda *Maxima* contiene una característica de reproducción:

`playback();`

3.7 El inicio de *wxMaxima* con el mensaje “Procesado de Maxima terminado.”

Una razón posible es que *Maxima* no puede ser encontrada dentro de la localización que está establecida dentro de la etiqueta “Maxima” del diálogo de configuración de *wxMaxima* y por lo tanto no ejecutará nada. Estableciendo la ruta a un *Maxima* binario para el trabajo debería solucionar este problema.

3.8 ‘Maxima’ está calculando para siempre y no responde a la entrada

Es posible teóricamente que *wxMaxima* no realiza que *Maxima* ha finalizado el cálculo y por lo tanto nunca está informado que puede enviar datos nuevos a *Maxima*. Si esto es el caso “Disparar evaluación” tal vez resincronice los dos programas.

3.9 Mi *Maxima* basada en SBCL se ejecuta sin memoria

El compilador Lisp SBCL por defecto viene con un límite de memoria que permite ejecutar incluso en computadoras de bajo nivel. Cuando compile un paquete de software grande como Lapack u ocupándose con listados extremadamente grandes o ecuaciones este límite tal vez sea demasiado bajo. Con el fin de extender los límites SBCL puede ser previsto con la línea del parámetro de instrucción `--dynamic-space-size` que indica a SBCL cuantos megabytes debería reservar. Una ventana SBCL de 32-bit puede reservar hasta 999 megabytes,

Una versión SBCL de 64-bit ejecutándose en Windows puede ser instruido para utilizar más que la cantidad de 1280 megabytes compilando las necesidades Lapack.

Una manera de proporcionar a *Maxima* (y por lo tanto SBCL) con parámetros de línea de instrucción es el campo «Parámetros adicionales para Maxima» del diálogo de configuración de *wxMaxima*.

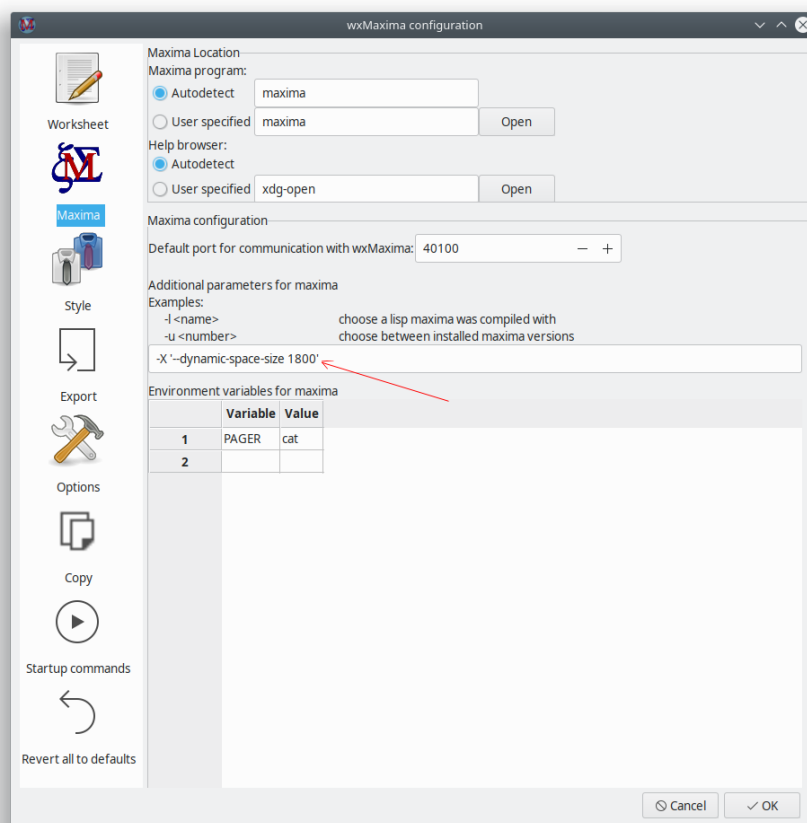


Figure 11: sbcl memory

3.10 La entrada algunas veces el tecleo es vago/perezoso en Ubuntu

La instalación del paquete «ibus-gtk» debería resolver este problema. Vea <https://bugs.launchpad.net/ubuntu/+source/wxwidgets3.0/+bug/1421558> para los detalles.

3.11 *wxMaxima* se detiene cuando *Maxima* procesa caracteres griegos o umlauts

Si su *Maxima* está basado en SBCL las siguientes líneas tienen que ser agregadas a su `.sbclrc`:

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

La carpeta de este fichero tiene que colocarse dentro del sistema- y la instalación específica. Pero cualquier *Maxima* basada en SBCL que ya tiene evaluada una celda dentro de la sesión actual felizmente dirá donde puede ser encontrada tras obtener la instrucción siguiente:

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

3.12 Tramado

3.12.1 ¿Se puede crear la salida de *wxMaxima* por ambos fichero de imagen y tramas empotradas a la vez?

El cuaderno empotra ficheros .png. *wxMaxima* permite al usuario especificar donde deberían ser generados:

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

Si un formato distinto está para ser utilizado es más fácil generar las imágenes y después importarlas dentro de la hoja del cuaderno de nuevo:

```
load("draw");  
pngdraw(name,[contents]):=  
(  
  draw(  
    append(  
      [  
        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 ¿Puede establecer el radio de aspecto de una trama?

Utilizando *Maxima* no directamente. Pero hay instrucciones gnuplot para ésta:

```

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

```

4 P+F

4.1 ¿Hay una manera de crear más texto que quepa en una página LaTeX?

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[izq=1cm,der=1cm,cim=1cm,fon=1cm]{geometría}
```

4.2 ¿Hay un modo oscuro?

Si wxWidgets es suficientemente nuevo *wxMaxima* automáticamente estará en modo oscuro si el resto del sistema operativo también lo está. La misma hoja de trabajo por defecto está equivocada con un fondo brillante. Pero puede ser configurado de otras maneras. Alternativamente hay un apunte del menú para ‘Ver/Invertir brillo del cuaderno’ para convertir rápidamente la hoja del cuaderno desde oscuro a claro y viceversa.

4.3 *wxMaxima* algunas veces se cuelga durante algunos segundos una vez en el primer minuto

wxMaxima delega algunas tareas grandes como interpretando el manual de página 1000 de *Maxima* para tareas del fondo, la cual normalmente va totalmente no comunicado. En el momento el resultado de dicha tarea es necesario, es posible que *wxMaxima* necesite esperar un par de segundos antes de poder continuar su trabajo.

5 Argumentos de línea de instrucción

Muchos sistemas operativos proporcionan menos formas complicadas de iniciar programas que la línea de instrucciones por lo que esta posibilidad es únicamente raramente utilizada. *wxMaxima* aún proporciona algunos indicadores por línea de instrucciones, de todas formas.

- ‘-v’ o ‘-version’: extrae la información de la versión
- ‘-h’ o ‘-help’: extrae un texto breve de ayuda
- ‘-o’ o ‘-open=’: abre el nombre del fichero dado como argumento a este interruptor de línea-transaccional
- ‘-e’ o ‘-eval’: evalúa el fichero tras abrirlo.
- ‘-b’ o ‘-batch’: si la línea de instrucción abre un fichero todas las celdas dentro de este fichero son evaluadas y el fichero es guardado después. Esto es para ejemplo útil si la sesión descrita dentro del fichero hace que *Maxima* genere ficheros de salida. Los procesamientos batch serán detenidos si *wxMaxima* detecta que *Maxima* tiene un error de salida y lo detendrá si *Maxima* tiene una cuestión: las matemáticas es algo interactivo por naturaliza por lo que un proceso de batch libre de interacción no puede ser completamente garantizado.
- --logtostdout: registra todos los mensajes del lateral “mensajes de depuración” también a la ‘stderr’.
- --pipe: mensajes entubados desde Maxima a ‘stdout’.
- --exit-on-error: cierra el programa en cualquier error de maxima.
- -f or --ini=<cdn>: utilice el fichero de init que fue proporcionado como argumento a esta contorno de línea de instrucciones.
- -u, --use-version=<cdn>: utilice la versión de ‘maxima’ <cdn>.
- -l, --lisp=<cdn>: utilice un compilador de Maxima compilado con Lisp <cdn>.
- -X, --extra-args=<cdn>: permite especificar argumentos adicionales de “Maxima”
- -m or --maxima=<cdn>: permite especificar la ubicación del binario de *maxima*
- --enableipc: permite que “Maxima” controle “wxMaxima” a través de comunicaciones de interprocesos. Utilice esa opción con cuidado.

En vez de un menos algunos sistemas operativos tal vez utilicen un guión al frente de las opciones de línea de instrucciones.