

# **Introducción a Maxima**

*Miguel Arsuaga Franco*

*Rosa Ramos Palanco*

*Lic. CC. Matemáticas*

*Profesores del Departamento de Matemáticas  
del I.E.S. Almunia de Jerez de la Frontera*



## Índice de materias

<b>Presentación</b>	<b>1</b>
A quién va dirigido este documento	1
Breve historia de Maxima	1
Otras consideraciones	1
<b>Normas básicas</b>	<b>3</b>
Descripción de la pantalla	3
El navegador NetMath	3
Órdenes	3
Modos de cálculo	4
<b>Aritmética</b>	<b>5</b>
Operadores aritméticos	5
<b>Funciones predefinidas</b>	<b>7</b>
Algunas constantes útiles	7
Exponenciales y logarítmicas	7
Trigonométricas	7
Otras funciones interesantes	9
<b>Álgebra</b>	<b>11</b>
Operaciones con polinomios y razones algebraicas.	11
Resolución de ecuaciones y sistemas de ecuaciones	12
Inecuaciones	14
<b>Matrices y determinantes</b>	<b>15</b>
Operaciones	15
<b>Análisis</b>	<b>17</b>
Sucesiones	17
Funciones. Composición de funciones	17
Funciones a trozos:	18
Límites de funciones	18
Derivadas	19
Integrales	19
<b>Gráficos</b>	<b>21</b>
¿Cómo verlos?	21
Gráficos bidimensionales	21
Gráficos tridimensionales	24
<b>Consideraciones finales</b>	<b>25</b>
Recapitulemos	25
Materiales	25
Derechos de copia	25



## Presentación

Maxima es un programa de cálculo simbólico bajo licencia GNU-GPL.

Es, por tanto, software libre en su doble acepción: está disponible el código fuente para que todo el que lo desee adapte el programa a sus propias necesidades y es gratuito por lo que es posible distribuirlo libremente a cuantos estuvieren interesados en el mismo.

Además puede ejecutarse sobre diferentes sistemas operativos, entre otros las diversas versiones de MS-Windows y de Linux.

### A quién va dirigido este documento

Esta introducción a Maxima tiene por destinatario a los alumnos de Matemáticas I y II de lo que queda del Bachillerato de Ciencias y, por extensión, a todos los alumnos y profesores a los que pueda servir de ayuda.

No cubre, por tanto, muchas de las capacidades del programa sino que se restringe a los contenidos básicos del bachillerato y, dentro de estos contenidos se ha procurado recurrir a ejemplos sencillos para que todos los alumnos puedan seguir este manual, aunque su nivel de competencia en la materia sea bajo.

Para la Estadística descriptiva y distribuciones de probabilidad, tanto para variables discretas como continuas, Mario Rodríguez Riotorto ha preparado dos paquetes, `descriptive.mac` y `distrib.mac` que se pueden descargar desde [www.biomates.net](http://www.biomates.net) junto a la documentación y archivos con datos muestrales. Un estudio de los mismos y algo de tiempo disponible nos permitirán ampliar este manual con aquellas funciones que hagan referencia a los contenidos en el Bachillerato.

### Breve historia de Maxima

El proyecto Matemáticas y Computación (MAC) se inicia en la década de los 60 del siglo pasado en el Instituto de Tecnología de Massachusetts (MIT), con el apoyo financiero, entre otros, de los departamentos de defensa (DOD), de Energía (DOE) y de la Agencia de Proyectos de Investigación Avanzada (ARPA) para atender a las necesidades de cálculo matemático de físicos, ingenieros y técnicos en general.

A finales de la década, el proyecto/programa Macsyma (MAC symbolic manipulator) se convierte en el primer programa de cálculo simbólico. Desarrollos posteriores dieron origen a múltiples variantes. Una de ellas es Maxima, heredera del DOE-Macsyma, que ha sido mantenida por el Dr. William Shelter de la Universidad Austin de Texas hasta su fallecimiento, fecha en la que un equipo de seguidores ha asumido su mantenimiento y desarrollo. En octubre de 1998 el DOE autorizó la distribución de Maxima bajo licencia GNU-GPL. Las actualizaciones del programa y de su documentación están disponibles en <http://maxima.sourceforge.net/>.

### Otras consideraciones

Hay ligeras diferencias entre la versión utilizada en este documento y la incluida en Guadalinux. Por ejemplo: las líneas de comando C1 y de respuesta D1 son ahora %i1 y %o1 (input y output). Más adelante veremos que son modificables por el usuario.

Está disponible la extensión wxMaxima que le da un aspecto al programa que recuerda a Derive y que permite introducir las órdenes más frecuentes mediante botones y opciones de menú. Es una ayuda que agradecemos los que no somos buenos mecanógrafos. Esperemos que sea incluida en Guadalinux.

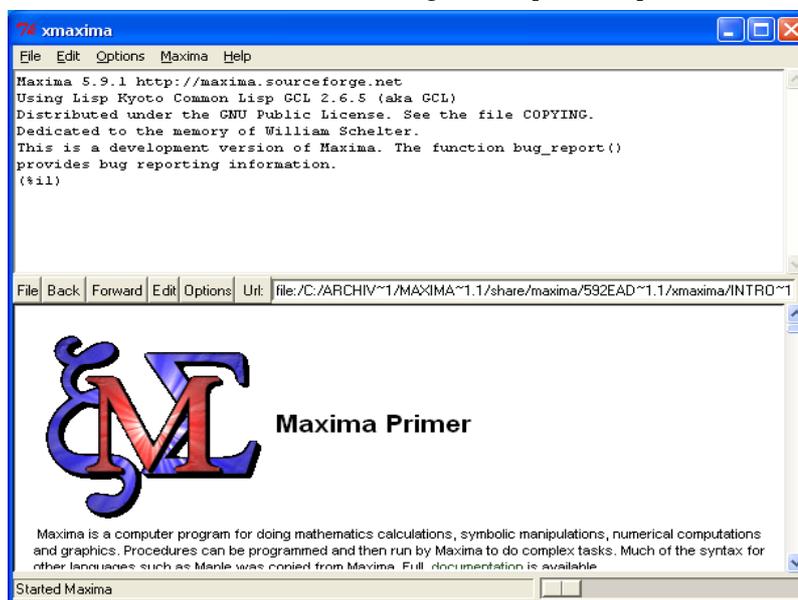


Ilustración 2: Ventana de inicio de xMaxima

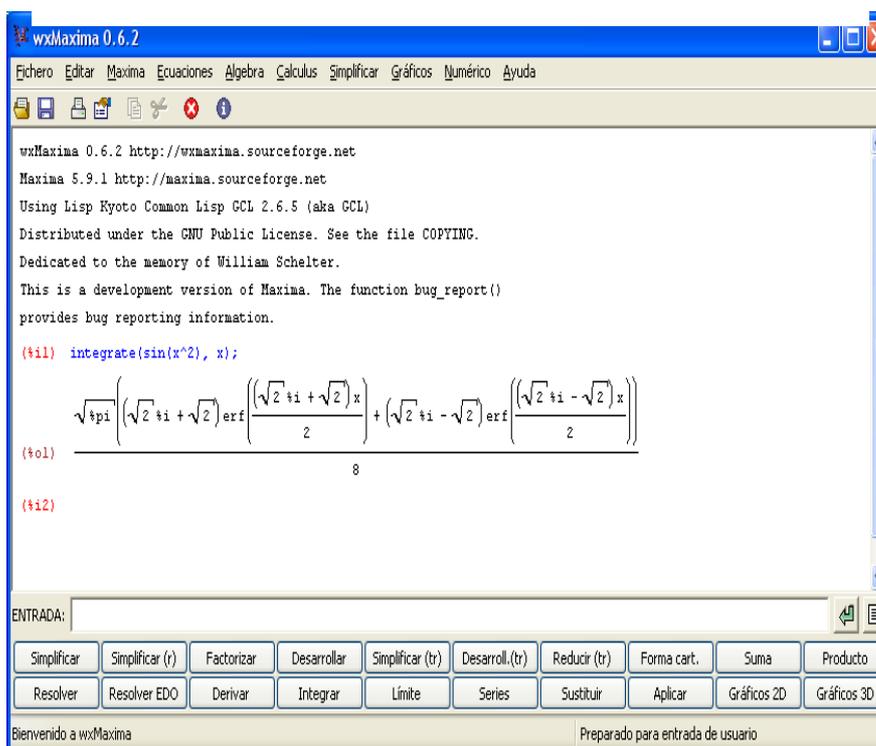


Ilustración 3: Ventana de Inicio de wxMaxima

## Normas básicas

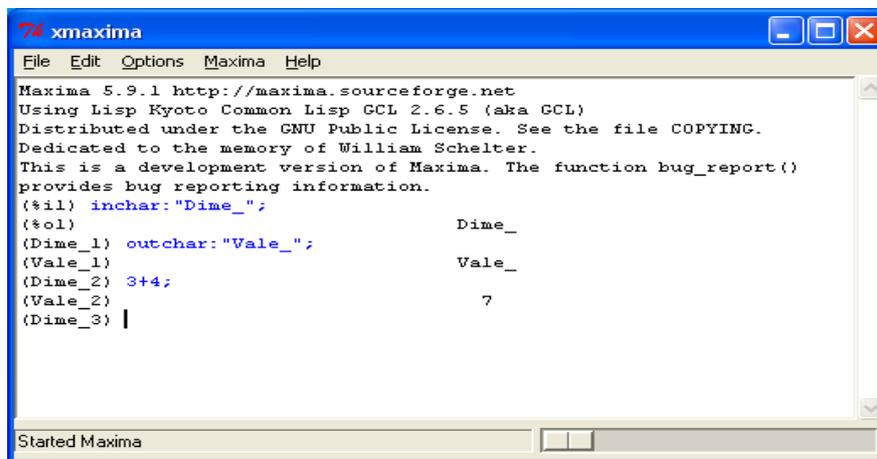
### Descripción de la pantalla

Como puede observarse en la ventana de inicio de xMaxima, ésta se divide en dos ventanas.

En la ventana superior es donde introducimos las órdenes que el programa procesará. Las órdenes se numeran de forma automática %i1, %i2, etc. y las respuestas correspondientes %o1, %o2, etc.

Podemos hacer referencia a resultados anteriores por su etiqueta. Si se trata del resultado inmediatamente anterior basta %.

Las variables inchar y outchar controlan las etiquetas de entrada y salida. Observe el siguiente ejemplo:



```

xmaxima
File Edit Options Maxima Help
Maxima 5.9.1 http://maxima.sourceforge.net
Using Lisp Kyoto Common Lisp GCL 2.6.5 (aka GCL)
Distributed under the GNU Public License. See the file COPYING.
Dedicated to the memory of William Schelter.
This is a development version of Maxima. The function bug_report()
provides bug reporting information.
(%i1) inchar: "Dime_";
(%o1)                                     Dime_
(Dime_1) outchar: "Vale_";
(Vale_1)                                 Vale_
(Dime_2) 3+4;
(Vale_2)                                 7
(Dime_3) |
  
```

Ilustración 4: Cambiando las etiquetas de entrada y salida

### El navegador NetMath

La ventana inferior es un navegador, NetMath, muy limitado pero eficiente. Es posible desactivarlo seleccionando [Options] → [Toggle Browser Visibility], de esta forma dispondremos de un espacio de trabajo con más líneas visibles. Para recuperarlo basta repetir la orden anterior.

Abriremos páginas web utilizando el habitual `http://dirección completa de internet`.

Abriremos archivos sustituyendo `http` por `file` y proporcionando la ruta completa y nombre del archivo.

En esta ventana es posible abrir documentos .htm preparados para ejecutar comandos. Los resultados pueden visualizarse en el propio documento o en la consola. La página de inicio es un ejemplo muy completo; contiene enlaces a la documentación de maxima, por la que podemos navegar, y ejemplos de comandos. Anímate a experimentar con ella.

### Órdenes

Las órdenes en Maxima terminan con ";".

Si no deseamos que el resultado de una orden aparezca terminaremos con \$.

Es posible introducir varias órdenes en una línea.

Todo lo que introduzcamos entre `/* y */` será ignorado por Maxima, lo que nos permite comentar las órdenes.

Para asignar valor a una variable utilizamos `:`. `variable:valor`.

Para definir una función usaremos `:=`. `función:=definición`.

```

xmaxima
File Edit Options Maxima Help
This is a development version of Maxima. The function bug_report()
provides bug reporting information.
(i1) 2+3; /* Suma de dos números */
(o1) 5
(i2) 2*3;2^3;2/3; /* Varias operaciones en la misma línea */
(o2) 6
(i3) 8
(o3) 8
(i4) 2
(o4) -
(i5) x:2$ /* x vale 2 */
(o5) 3
(i6) x+8;
(o6) 10
(i7) f(x):= x^2+2*x$ /* definimos f(x) */
(o7) 10
(i8) f(2); f(-3);
(o8) 8
(i9) 3
(o9) 3
(i10) |
Started Maxima
    
```

Ilustración 5: Órdenes en Maxima. Variables y funciones.

Cuando dejemos de usar una variable conviene “matarla” con la función `kill(nombre de la variable)` para no llevarnos sorpresas.

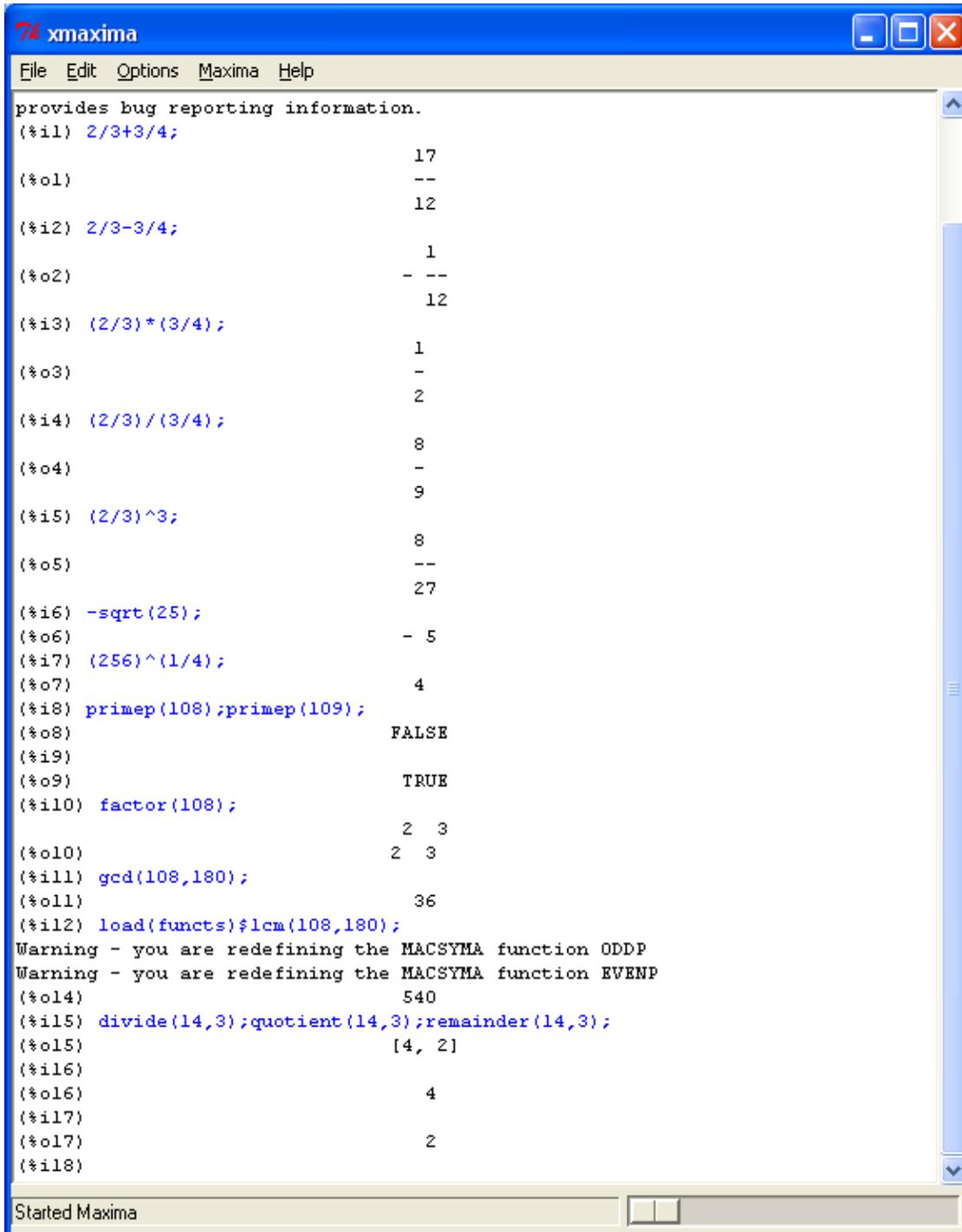
### Modos de cálculo

Por defecto, Maxima hace cálculos exactos. La orden `numer` nos da el valor aproximado. También podemos hacer esto con `float`.

Podemos fijar la precisión de los cálculos asignando el valor deseado a la variable `fpprec` y obtener el valor deseado en notación científica con `bfloat`. Observa el siguiente ejemplo:

## Aritmética

## Operadores aritméticos



```

xmaxima
File Edit Options Maxima Help
provides bug reporting information.
(%i1) 2/3+3/4;
              17
(%o1)      --
              12
(%i2) 2/3-3/4;
              1
(%o2)      - ---
              12
(%i3) (2/3)*(3/4);
              1
(%o3)      -
              2
(%i4) (2/3)/(3/4);
              8
(%o4)      -
              9
(%i5) (2/3)^3;
              8
(%o5)      --
              27
(%i6) -sqrt(25);
(%o6)      - 5
(%i7) (256)^(1/4);
(%o7)      4
(%i8) primep(108);primep(109);
(%o8)      FALSE
(%i9)
(%o9)      TRUE
(%i10) factor(108);
              2 3
(%o10)      2 3
(%i11) gcd(108,180);
(%o11)      36
(%i12) load(funcs)$lcm(108,180);
Warning - you are redefining the MACSYMA function ODDP
Warning - you are redefining the MACSYMA function EVENP
(%o14)      540
(%i15) divide(14,3);quotient(14,3);remainder(14,3);
(%o15)      [4, 2]
(%i16)
(%o16)      4
(%i17)
(%o17)      2
(%i18)
Started Maxima

```

Ilustración 6: Aritmética con Maxima

La suma (+) y la resta (-), el producto (\*), el cociente (/) y las potencias (^) usan los mismos operadores que la mayoría de los programas (por ejemplo: las hojas de cálculo). La raíz cuadrada puede calcularse con la función `sqrt` (square root), el resto de las raíces habrá de calcularse como exponentes fraccionarios.

Para efectuar la división entera disponemos de `divide(dividendo, divisor)`. Nos devuelve una lista con el cociente y el resto. Si únicamente necesitamos uno u otro usaremos `quotient` o `remainder` respectivamente.

Podemos averiguar si un número es primo, o no con la función `primep`.

Para descomponer en factores primos usaremos `factor`. Las funciones `gcd` (greatest common divisor) y `lcm` (least common multiple) permiten obtener el máximo común divisor y el mínimo común múltiplo, respectivamente. Para calcular éste último es preciso cargar previamente el paquete `functs`. Maxima avisa de que ello supone redefinir algunas funciones. No le hacemos mucho caso.

Todo lo anterior queda explicado con la Ilustración 6.

## Funciones predefinidas

### Algunas constantes útiles

Se utilizan:

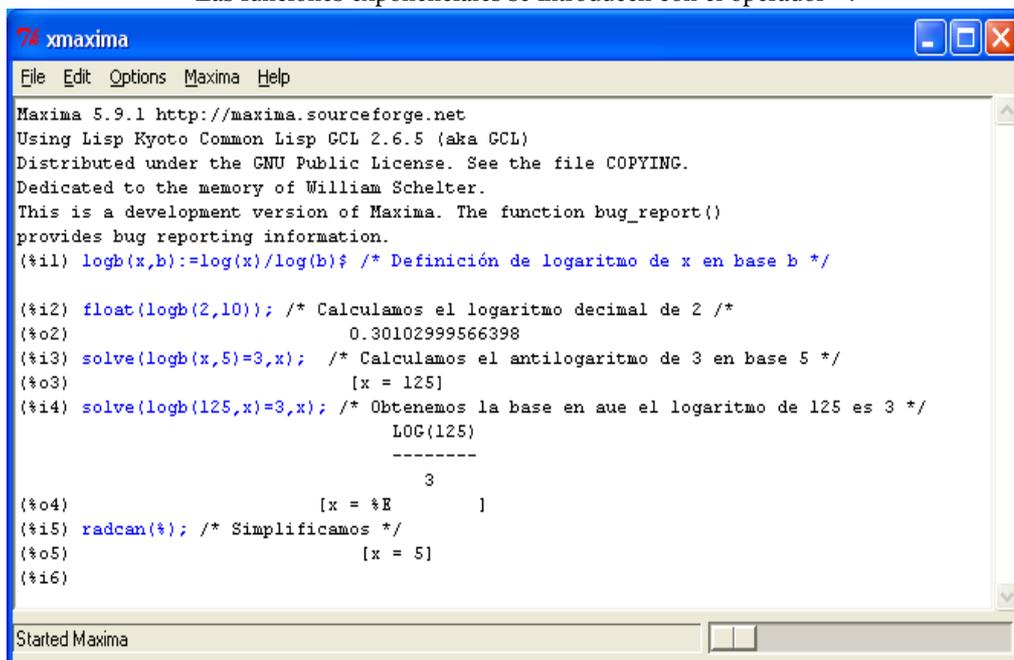
- %E para el número  $e$
- %pi para el número  $\pi$
- %i para  $i=\sqrt{-1}$

### Exponenciales y logarítmicas

Maxima utiliza el logaritmo natural (neperiano) como función logarítmica predefinida.. Es posible definir una función alternativa, le hemos llamado  $\log_b(x, b)$ , para calcular  $\log_b(x)$ .

Conviene observar que tal función puede utilizarse para obtener el antilogaritmo o la base logarítmica. En el segundo caso, conviene simplificar con la función `radcan`.

Las funciones exponenciales se introducen con el operador  $\wedge$ .



```

xmaxima
File Edit Options Maxima Help
Maxima 5.9.1 http://maxima.sourceforge.net
Using Lisp Kyoto Common Lisp GCL 2.6.5 (aka GCL)
Distributed under the GNU Public License. See the file COPYING.
Dedicated to the memory of William Schelter.
This is a development version of Maxima. The function bug_report()
provides bug reporting information.
(%i1) logb(x,b):=log(x)/log(b)$ /* Definición de logaritmo de x en base b */

(%i2) float(logb(2,10)); /* Calculamos el logaritmo decimal de 2 */
(%o2) 0.30102999566398
(%i3) solve(logb(x,5)=3,x); /* Calculamos el antilogaritmo de 3 en base 5 */
(%o3) [x = 125]
(%i4) solve(logb(125,x)=3,x); /* Obtenemos la base en aue el logaritmo de 125 es 3 */
(%o4) LOG(125)
-----
3
[x = %E ]
(%i5) radcan(%); /* Simplificamos */
(%o5) [x = 5]
(%i6)
Started Maxima

```

Ilustración 7: Cálculo de logaritmos

### Trigonómicas

Las usuales funciones trigonométricas y sus respectivas inversas arco ... están definidas en Maxima pero, eso sí, sus argumentos son radianes. Si se desea calcular el seno, por ejemplo, de un ángulo expresado en grados sexagesimales u obtener los ángulos de las inversas en grados habrá que definir las correspondientes funciones.

```

xmaxima
File Edit Options Maxima Help
(%i6) rad(g):=(g*pi)/180$ /* Para convertir grados en radianes */
(%i7) deg(r):=(r*180)/%pi$ /* Para convertir radianes en grados */
(%i8) sen(g):=sin(rad(g))$ /* Para calcular el seno de un ángulo en grados */
(%i9) asen(x):=deg(asin(x))$ /* Para obtener el arco-seno en grados */

(%i10) sin(%pi/2);
(%o10) 1
(%i11) sen(90);
(%o11) 1
(%i12) asin(1);
(%o12) %PI
      ---
      2
(%i13) asen(1);
(%o13) 90
(%i14) |
Started Maxima
    
```

Las funciones trigonométricas más usuales predefinidas en Maxima se reseñan a continuación.

sin	Seno
cos	Coseno
tan	Tangente
sec	Secante
csc	Cosecante
cot	Cotangente
asin	Arco-seno
acos	Arco-coseno
atan	Arco-tangente
asec	Arco-secante
acsc	Arco-cosecante
cot	Cotangente

Las funciones `trigsimp`, `trigrat`, `trigreduce` y `trigexpand` realizan diferentes simplificaciones y/o transformaciones trigonométricas, experimenta con ellas.

```

xmaxima
File Edit Options Maxima Help

(%i14) (sin(x))^2+(cos(x))^2;
(%o14)          2      2
          SIN (x) + COS (x)
(%i15) trigreduce(% ,sin(x));
(%o15)          2
          1 - COS(2 x)
          ----- + COS (x)
          2
(%i16) trigreduce(%);
(%o16)          2      2
          COS(2 x) + 1  1 - COS(2 x)
          ----- + -----
          2              2
(%i17) trigexpand(%);
(%o17)          2      2      2      2
          SIN (x) - COS (x) + 1  - SIN (x) + COS (x) + 1
          ----- + -----
          2              2
(%i18) trigsimp(%);
(%o18)          1
(%i19) |

```

Started Maxima

### Otras funciones interesantes

<code>abs(x)</code>	Valor absoluto de x
<code>min(x1, x2, ...)</code>	Valor mínimo de una lista de valores
<code>max(x1, x2, ...)</code>	Valor máximo de una lista de valores
<code>signum(x)</code>	$\text{signum}(x) = \begin{cases} -1 & \text{si } x < 0 \\ 0 & \text{si } x = 0 \\ 1 & \text{si } x > 0 \end{cases}$
<code>x!</code>	Factorial de x
<code>binomial(m, n)</code>	$C_m^n = \binom{m}{n}$
<code>abs(x)</code>	Valor absoluto de x
<code>exp(x)</code>	$e^x$



## Álgebra

### Operaciones con polinomios y razones algebraicas.

Para las operaciones de suma, resta, multiplicación, división, factorización, máximo común divisor y mínimo común múltiplo de polinomios, hay que tener en cuenta lo explicado en el capítulo de aritmética.

Para desarrollar expresiones algebraicas usaremos la función `expand`.

The screenshot shows the xmaxima interface with the following content:

```

xmaxima
File Edit Options Maxima Help
(%i1) expand((x+2)*(3*x+1));
(%o1)          2
          3 x  + 7 x + 2
(%i2) factor(x^3-3*x^2+3*x-1);
(%o2)          3
          (x - 1)
(%i3) divide(3*x^2-5*x-4,4*x+2);
(%o3)          6 x - 13      3
          [-----, - -]
              8          4
(%i4) (2*x^3)^4;
(%o4)          12
          16 x
(%i5) gcd(x^2-2*x,x^2-3*x);
(%o5)          x
(%i6) load(funcs);
Warning - you are redefining the MACSYMA function ODDP
Warning - you are redefining the MACSYMA function EVENP
(%o6) C:/ARCHIV~1/MAXIMA~1.1/share/maxima/5.9.1/share/simplification#
/funcs.mac
(%i7) lcm(x^2-2*x,x^2-3*x);
(%o7)          (x - 3) (x - 2) x
(%i8)
Started Maxima

```

Para simplificar disponemos de las funciones `ratsimp`, `fullratsimp` y `radcan`.

En algunas operaciones Maxima separa una fracción en varias con el mismo denominador, la orden `combine` las convierte en una.

Podemos seleccionar el numerador o el denominador de una fracción algebraica con las funciones `num` y `denom` respectivamente. Ver los ejemplos.

## Resolución de ecuaciones y sistemas de ecuaciones

Usaremos la función `solve` para resolver ecuaciones y sistemas de ecuaciones. Veámoslo.

```

xmaxima
File Edit Options Maxima Help
(%i11) solve(x^2-3*x+2,x);
(%o11) [x = 1, x = 2]
(%i12) solve (a*x^3+b*x^2,x);
(%o12) [x = - , x = 0]
          b
          a
(%i13) 3/x=(2*x-1)/(x^2+3*x);
(%o13) 3 2 x - 1
      - = -----
      x 2
        x + 3 x
(%i14) solve(%);
(%o14) [x = - 10]
(%i15) %o13,x=-10; /* Comprobamos la solución */
(%o15) 3 3
      - - = - -
      10 10
(%i16)
Started Maxima

```

Ilustración 11: Resolviendo ecuaciones

Observa que si no hay un signo `=`, Maxima asume que buscamos los ceros de la ecuación. En la línea `%i15` se explica cómo sustituir la solución para comprobar la validez de la solución.

Resolviendo manualmente la ecuación de `%o13` obtenemos  $[x=0, x=-10]$ . Maxima elimina las pseudo soluciones.

La instrucción `algsys` permite obtener soluciones aproximadas cuando no se dispone de un método algebraico de resolución de la ecuación. Con la opción `realonly:true` obtenemos exclusivamente soluciones reales.

La función `solve` se puede usar también para despejar algún término en una ecuación. Observa cómo “enseñamos” a Maxima a resolver una ecuación con radicales.

Las dos ilustraciones siguientes ejemplifican lo que decimos.

```

xmaxima
File Edit Options Maxima Help
(%i24) x^5-3*x^4+2*x^3-2*x^2-x+4=0;
(%o24) x5 - 3 x4 + 2 x3 - 2 x2 - x + 4 = 0
(%i25) solve(%);
(%o25) [0 = x5 - 3 x4 + 2 x3 - 2 x2 - x + 4]
(%i26) algsys(%o24,[x]);
(%o26) [[x = 2.478283086356668],

[x = 0.11500575571173 - 1.27155810694299 %I],

[x = 1.27155810694299 %I + 0.11500575571173],

[x = - 0.85983966899405], [x = 1.151545166402536]]
(%i27) realonly:true;
(%o27) TRUE
(%i28) %%i26;
(%o28) [[x = 2.478283086356668],

[x = 1.151545166402536], [x = - 0.85983966899405]]
(%i29)
Started Maxima

```

Ilustración 12: Obteniendo soluciones reales aproximadas..

```

xmaxima
File Edit Options Maxima Help
(%i30) ec:sqrt(x+1)+sqrt(x+6)=3; /* Una ecuación irracional */
(%o30) Sqrt(x + 6) + Sqrt(x + 1) = 3
(%i31) solve(ec,sqrt(x+6)); /* Despejamos una de las raíces */
(%o31) [Sqrt(x + 6) = 3 - Sqrt(x + 1)]
(%i32) expand(%^2); /* Elevamos al cuadrado */
(%o32) [x + 6 = - 6 Sqrt(x + 1) + x + 10]
(%i33) solve(% ,sqrt(x+1)); /* Volvemos a despejar */
(%o33) [Sqrt(x + 1) = -]
(%i34) solve(% ,x); /* Y resolvemos */
(%o34) [x = - 5/9]
(%i35) solve(ec,x); /* Intentamos resolver directamente */
(%o35) [Sqrt(x + 6) = 3 - Sqrt(x + 1)]
(%i36)
Started Maxima

```

Ilustración 13: Despejando para resolver una ecuación irracional.

Igualmente podemos resolver sistemas de ecuaciones lineales. Cuando son indeterminados, obtiene las soluciones en función de uno o más parámetros. Los %R3 y %R4 serían como  $\alpha$  y  $\beta$ .

```

xmaxima
File Edit Options Maxima Help
(%i37) solve([2*x+3*y-z=7,x+4*y+2*z=11,-3*x+4*y+7*z=-2],[x,y,z]);
/* Compatible determinado */
(%o37)      [[x = 31/3, y = -8/3, z = 17/3]]
(%i38) solve(x+y+z=1,[x,y,z]); /* Compatible indeterminado */
(%o38)      [[x = - %R4 - %R3 + 1, y = %R4, z = %R3]]
(%i39) solve([x+y=1,x+y=2],[x,y]); /* Incompatible */
Inconsistent equations: (2)
-- an error. Quitting. To debug this try DEBUGMODE(TRUE);
(%i40)
Started Maxima

```

Ilustración 14: Resolución de sistemas de ecuaciones

## Inecuaciones

Maxima no resuelve inecuaciones, pero le podemos enseñar buscando los ceros y estudiando el signo en los intervalos que corresponda. De eso trata el siguiente ejercicio.

```

Maxima Manual - Table of Contents
File Edit Options Maxima Help
(%i44) ec: (x+1)/(x-1);
(%o44)      x + 1
            -----
            x - 1
(%i45) solve(ec<0,x); /* Intentamos resolver la inecuación */
Cannot solve inequalities. -SOLVE
-- an error. Quitting. To debug this try DEBUGMODE(TRUE);
(%i46) solve(num(ec),x);solve(denom(ec),x);
(%o46)      [x = - 1]
(%o47)      [x = 1]
(%i48) assume(x<-1)$is(ec<0);forget(x<-1); /* Signo si x<-1 */
(%o48)      FALSE
(%i49) assume(x>-1,x<1)$is(ec<0);forget(x>-1,x<1); /* Signo si -1<x<1 */
(%o49)      TRUE
(%i55) assume(x>1)$is(ec<0);forget(x>1); /* Signo si x>1 */
(%o55)      FALSE
(%i60)
(%i61)
(%i62)
(%i63)
Opening file:/C:/ARCHIV~1/MAXIMA~1.1/share/maxima/592EAD~1.

```

Ilustración 15: Resolviendo una inecuación estudiando el signo

Está claro que la solución es  $x \in [-1, 1]$ .

## Matrices y determinantes

### Operaciones

Para Maxima, una matriz es una lista de listas en la que cada elemento es una fila. Observa que el operador “\*” multiplica elemento a elemento dos matrices. Para el producto matricial usaremos “.”

Para calcular la inversa podemos utilizar la función `invert` o el operador “ $^{-1}$ ”. El determinante de una matriz lo calculamos con `determinant` y el rango con `rank`.

```

xmaxima
File Edit Options Maxima Help
(%i1) A:matrix([1,2],[3,4])$B:matrix([5,6],[7,8])$
(%i2)
(%i3) A+B;2*A;A*B;A.B;
(%o3)
      [ 6  8 ]
      [    ]
      [ 10 12 ]
(%i4)
      [ 2  4 ]
(%o4)
      [    ]
      [ 6  8 ]
(%i5)
      [ 5 12 ]
(%o5)
      [    ]
      [ 21 32 ]
(%i6)
      [ 19 22 ]
(%o6)
      [    ]
      [ 43 50 ]
(%i7) invert(A);B^^-1;
(%o7)
      [ - 2  1 ]
      [    ]
      [ 3   1 ]
      [ -  - ]
      [ 2   2 ]
(%i8)
      [ - 4  3 ]
      [    ]
(%o8)
      [ 7   5 ]
      [ -  - ]
      [ 2   2 ]
(%i9) rank(A);
(%o9)
      2
(%i12) determinant(B);
(%o12)
      - 2
(%i13) |
Started Maxima

```

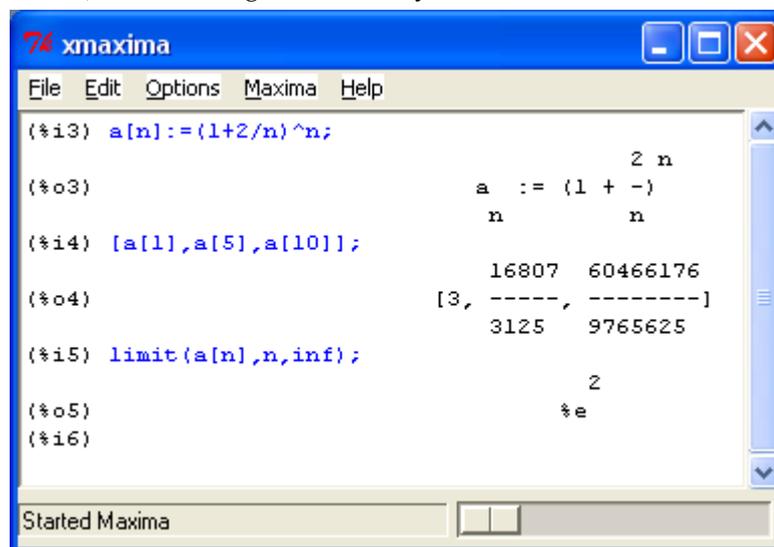
Ilustración 16: Operaciones con matrices.



## Análisis

### Sucesiones

Las sucesiones se definen de manera análoga a las funciones, salvo en el uso de corchetes para encerrar la variable. En el siguiente ejemplo definimos una sucesión, calculamos algunos términos y su límite.



```

7% xmaxima
File Edit Options Maxima Help

(%i3) a[n]:=(1+2/n)^n;

(%o3)
          2 n
      a := (1 + -)
          n   n

(%i4) [a[1],a[5],a[10]];

(%o4) [3, -----, -----]
      16807 60466176
      3125  9765625

(%i5) limit(a[n],n,inf);

(%o5)
          2
      %e

(%i6)

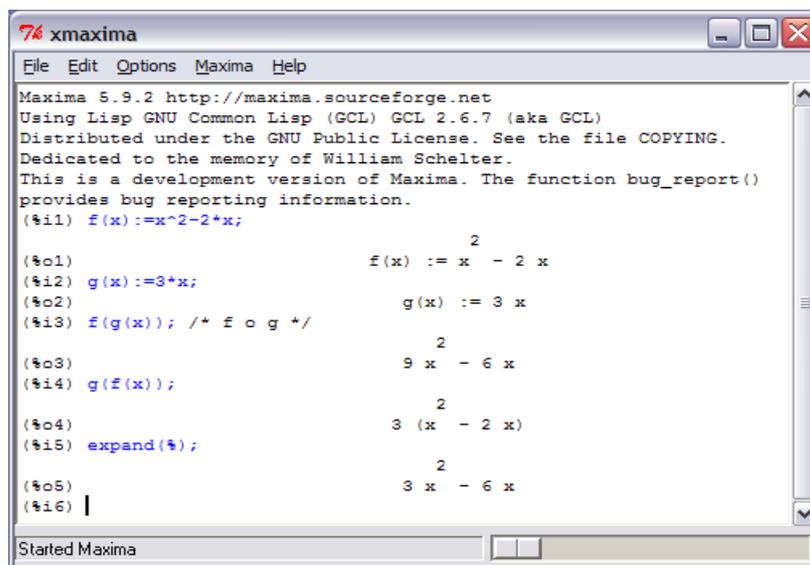
```

Started Maxima

Ilustración 17: Definiendo una sucesión

### Funciones. Composición de funciones

Definimos un par de funciones, por el método ya conocido y las componemos.



```

7% xmaxima
File Edit Options Maxima Help

Maxima 5.9.2 http://maxima.sourceforge.net
Using Lisp GNU Common Lisp (GCL) GCL 2.6.7 (aka GCL)
Distributed under the GNU Public License. See the file COPYING.
Dedicated to the memory of William Schelter.
This is a development version of Maxima. The function bug_report()
provides bug reporting information.

(%i1) f(x):=x^2-2*x;

(%o1)
          2
      f(x) := x  - 2 x

(%i2) g(x):=3*x;

(%o2)
      g(x) := 3 x

(%i3) f(g(x)); /* f o g */

(%o3)
          2
      9 x  - 6 x

(%i4) g(f(x));

(%o4)
          2
      3 (x  - 2 x)

(%i5) expand(%);

(%o5)
          2
      3 x  - 6 x

(%i6) |

```

Started Maxima

Ilustración 18: Componiendo funciones

**Funciones a trozos:**

$$\text{Vamos a definir } f(x) = \begin{cases} x^2 & \text{si } x < 0 \\ 2x - 1 & \text{si } 0 \leq x < 4 \\ 1 - x & \text{si } x \geq 4 \end{cases}$$

```

wxMaxima 0.6.4 http://wxmaxima.sourceforge.net
Maxima 5.9.2 http://maxima.sourceforge.net
Using Lisp GNU Common Lisp (GCL) GCL 2.6.7 (aka GCL)
Distributed under the GNU Public License. See the file COPYING.
Dedicated to the memory of William Schelter.
This is a development version of Maxima. The function bug_report()
provides bug reporting information.

(%i1) f(x):= if(x<0) then (x^2) else (if (x<4) then (2*x - 1) else (1 - x));

(%o1) f(x) := if x < 0 then x^2 else if x < 4 then 2 x - 1 else 1 - x

(%i2) [f(-2), f(2), f(5)];

(%o2) [ 4 , 3 , - 4 ]

```

Ilustración 19: Definiendo una función a trozos

**Límites de funciones**

Para calcular límites de funciones usaremos `limit`.

La estructura de la instrucción es:

`limit(función, variable, punto)`. Donde `punto` puede ser un valor, `inf` ( $\infty$ ) o `minf`, ( $-\infty$ ). En el caso primero admite un modificador `minus` o `plus` para indicar por la izquierda o por la derecha.

```

74 xmaxima
File Edit Options Maxima Help
Maxima 5.9.2 http://maxima.sourceforge.net
Using Lisp GNU Common Lisp (GCL) GCL 2.6.7 (aka GCL)
Distributed under the GNU Public License. See the file COPYING.
Dedicated to the memory of William Schelter.
This is a development version of Maxima. The function bug_report()
provides bug reporting information.
(%i1) f(x):=(x^3-1)/x^2;
(%o1)
          3
          x  - 1
f(x) := ----
          2
          x

(%i2) limit(f(x),x,inf);
(%o2)
      inf

(%i3) limit(f(x),x,minf);
(%o3)
      minf

(%i7) limit(f(x),x,0);
(%o7)
      minf

(%i8) limit(f(x),x,1,minus);
(%o8)
      0

(%i9) limit(f(x),x,1,plus);
(%o9)
      0
Started Maxima

```

Ilustración 20: Calculando límites

## Derivadas

La instrucción `diff` nos permite obtener la función derivada de una función, indicando opcionalmente el orden de la derivada. Por sustitución podemos obtener la derivada de una función en un punto.

```

74 xmaxima
File Edit Options Maxima Help
(%i34) diff(x^2*cos(3*x),x); /* La derivada */
(%o34)
      2
      x COS(3 x) - 3 x SIN(3 x)
(%i35) diff(x^2*cos(3*x),x,3); /* La derivada tercera */
(%o35)
      2
      27 x SIN(3 x) - 18 SIN(3 x) - 54 x COS(3 x)
(%i36) %o34,x=%pi/2; /*La derivada en %pi/2 */
(%o36)
      2
      3 %PI
      ----
      4
(%i37)
Started Maxima

```

Ilustración 21: Derivadas.

## Integrales

Las integrales indefinidas (es decir, el cálculo de la antiderivada o primitiva de una función) y la integral definida (el cálculo de la medida del área bajo una curva, por ejemplo) hacen uso de la misma instrucción `integrate`, que Maxima interpreta de una forma u otra según omitamos o no los límites inferior o superior del intervalo.

```

xmaxima
File Edit Options Maxima Help
(%i144) integrate(sqrt(x^2-1),x);
          2          2
      x Sqrt(x  - 1) LOG(2 Sqrt(x  - 1) + 2 x)
(%o44)  -----
          2          2
(%i146) integrate(sqrt(x^2-1),x,1,2);
          2          2
      LOG(2) LOG(2 Sqrt(3) + 4) - 2 Sqrt(3)
(%o46)  -----
          2          2
(%i147) integrate(%e^(-x^2),x,minf,inf);
(%o47)  Sqrt(%PI)
(%i148)
Started Maxima
  
```

Ilustración 22: Cálculo de integrales.

## Gráficos

### ¿Cómo verlos?

Conviene indicar que Maxima no tiene capacidades gráficas. Sino que actúa de interfaz de un programa de generación de gráficos – GNUplot –. Este programa admite gran número de opciones que precisarían de otro manual. Nos limitaremos a lo más elemental con algunos ejemplos sencillos.

Los gráficos pueden verse incrustados en la consola de trabajo, en una ventana independiente cuyo contenido se actualiza según vamos creando nuevos gráficos, o en múltiples ventanas; en este último caso cada nuevo gráfico fuerza la apertura de una ventana nueva.

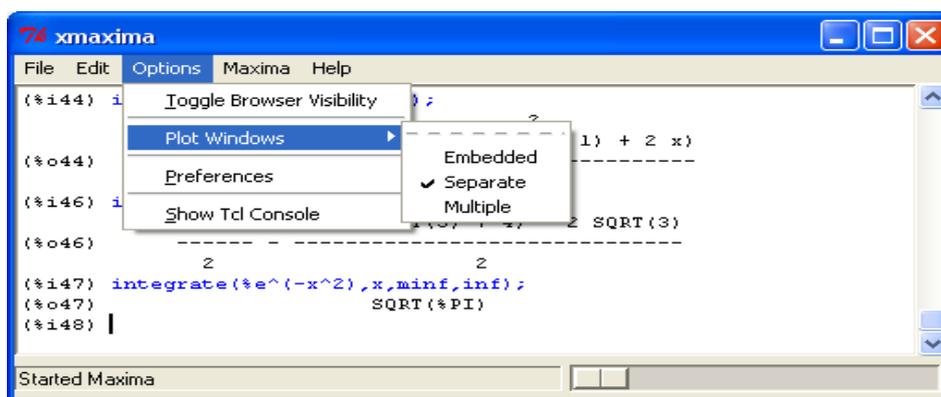


Ilustración 23: Opciones de la ventana de gráficos.

### Gráficos bidimensionales

La función `plot2d` permite múltiples variantes, como incluir una lista de funciones en el mismo gráfico, fijar el rango de valores de  $y$ , ... Lo mínimo que necesita es una función y el rango de valores de  $x$ .

Las  $s$  gráficas siguientes se han producido con las órdenes:

```
plot2d(sin(x), [x, -2*%pi, 2*%pi]);
```

y

```
plot2d([x^2-4*x, -2*x-1, 2*x-9], [x, -3, 7], [y, -5, 20],
[gnuplot_preamble, "set zeroaxis"], [nticks, 20]);
```

En el caso de la segunda gráfica no se ha capturado la pantalla sino que se ha hecho uso de las opciones de exportación de gráficos de gnuplot.

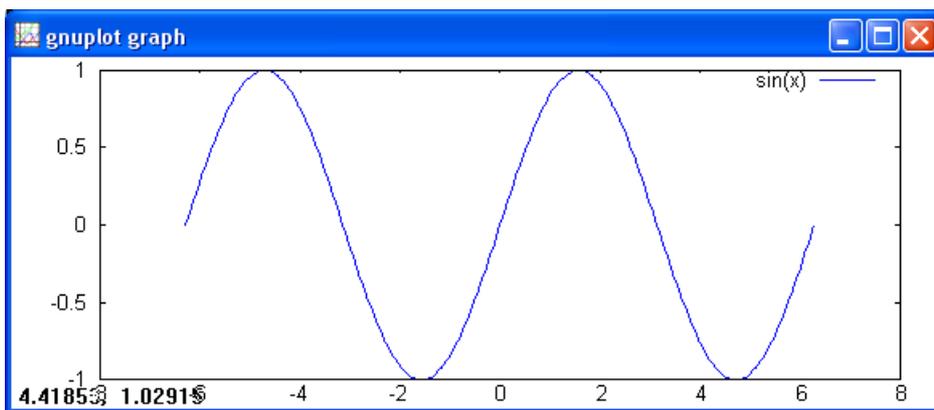


Ilustración 24: Gráfica de la función seno.

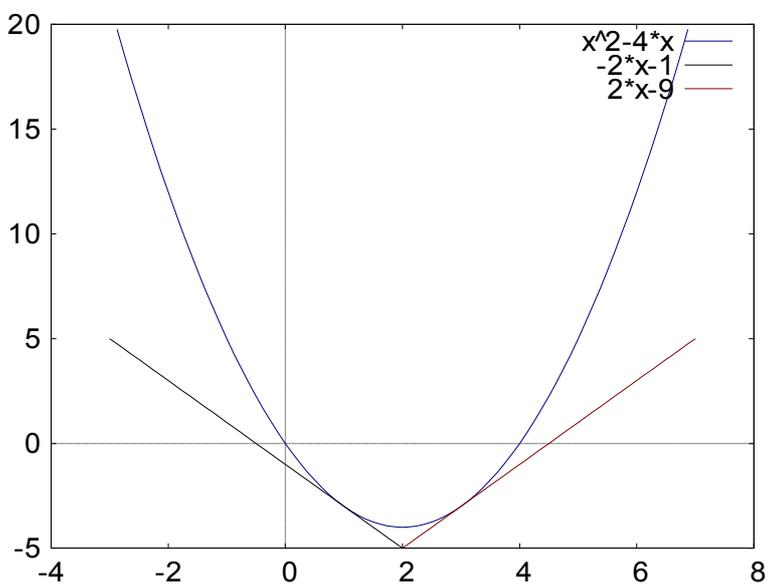


Ilustración 25: Gráfica de  $x^2 - 4x$  y sus tangentes en  $x=1$  y  $x=3$

¿Os acordáis de la función a trozos anterior? Vamos a representarla. Observa que las discontinuidades vienen marcadas por dos segmentos verticales que no deberían aparecer pero...

$$\text{Esta era la función: } f(x) = \begin{cases} x^2 & \text{si } x < 0 \\ 2x - 1 & \text{si } 0 \leq x < 4 \\ 1 - x & \text{si } x \geq 4 \end{cases}$$

```
wxMaxima 0.6.4 http://wxmaxima.sourceforge.net
Maxima 5.9.2 http://maxima.sourceforge.net
Using Lisp GNU Common Lisp (GCL) GCL 2.6.7 (aka GCL)
Distributed under the GNU Public License. See the file COPYING.
Dedicated to the memory of William Schelter.
This is a development version of Maxima. The function bug_report()
provides bug reporting information.

(%i1) f(x):= if(x<0) then (x^2) else (if (x<4) then (2*x - 1) else (1 -x));
(%o1) f(x) := if x < 0 then x^2 else if x < 4 then 2x - 1 else 1 - x
(%i2) [f(-2), f(2), f(5)];
(%o2) [ 4 , 3 , - 4 ]
(%i3) plot2d('f(x)), [x, -5, 10]);
```

ENTRADA:

Simplificar    Simplificar (r)    Factorizar    Expandir    Simplificar (tr)    Expandir (tr)    Reducir (tr)    F

Resolver ...    Resolver EDO ...    Derivar ...    Integrar ...    Límite ...    Serie ...    Sustituir ...    A

Maxima está calculando

Y ésta es su gráfica.

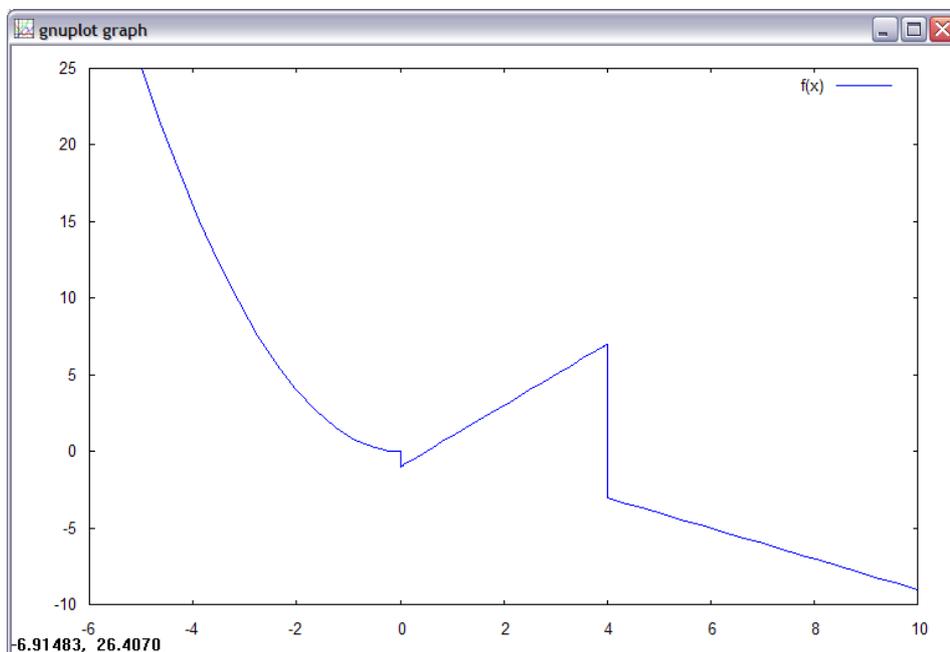


Ilustración 26: Gráfica de una función a trozos

### Gráficos tridimensionales

Para gráficos tridimensionales tenemos la instrucción `plot3d` en la que hay que precisar, al menos, los rangos de  $x$  y de  $y$ .

El siguiente ejemplo se ha hecho con la orden `plot3d(1+x^2-y^2, [x,-5,5], [y,-5,5], [gnuplot_preamble, "set pm3d at s; unset surf; unset colorbox"])`§

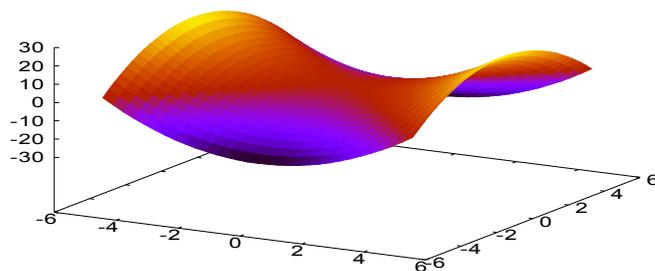


Ilustración 27: Un gráfico tridimensional

## Consideraciones finales

### Recapitulemos

Maxima y otras herramientas CAS son una potente herramienta de aprendizaje. Si, como es el caso de Maxima o Yacas o... son libres y están disponibles para Linux y Windows, entre otros sistemas operativos, mejor.

Podemos proporcionarles a nuestros alumnos estas herramientas para que puedan usarlas en casa sin tener que cambiar su sistema operativo o hacer problemáticas particiones del disco duro. Podemos trabajar en clase con Guadalinex y en casa con Windows.

El propósito de este manual es acercar al alumno al uso de estas herramientas no muy intuitivas pero que, una vez aprendido lo que se precisa, se convierte en un instrumento importante de aprendizaje. Y aumenta la autonomía, y por tanto, la seguridad en las propias capacidades, al poder contrastar rápidamente la validez de los resultados obtenidos.

Proporcionar a los alumnos en un CD OpenOffice, Maxima, wxMaxima y algunos manuales adaptados a sus necesidades les abren puertas que algunos cruzan.

### Materiales

Para redactar este manual hemos empleado OpenOffice 2.0, xMaxima y wxMaxima. Maxima reference manual y Primeros pasos en Maxima ha sido la documentación que básicamente hemos utilizado.

Conseguir estos programas y su documentación vía internet es muy simple. En sourceforge.net están alojadas un número casi inabarcable de herramientas matemáticas, entre ellas Maxima y wxMaxima y, en cualquier caso, cualquier buscador nos llevará inmediatamente a múltiples páginas de ayuda y documentación adicional. Es cuestión de ponerse.

### Derechos de copia

Copyright © 2004 Miguel Arsuaga Franco y Rosa Ramos Palanco. Este documento es libre. Se otorga permiso para copiarlo, distribuirlo y/o modificarlo en los términos de la Licencia de Documentación Libre GNU, versión 1.2 o posterior publicada por la Fundación de Software Libre, <http://es.gnu.org/licencias/fdles.html>.

