PRÁCTICA 1:
Primeros pasos con wxMaxima

Maxima es un programa que realiza cálculos matemáticos de forma tanto numérica como simbólica, esto es, sabe tanto manipular números como calcular la derivada de una función. Sus capacidades cubren sobradamente las necesidades de un alumno de un curso de Cálculo en unos estudios de Ingeniería. Se encuentra disponible bajo licencia GNU GPL tanto el programa como los manuales del programa.

Hay muchos programas que cumplen en mayor o menor medida los requisitos que se necesitan para enseñar y aprender Cálculo. Sólo por mencionar algunos, y sin ningún orden particular, casi todos conocemos *Mathematica* (©Wolfram Research) o *Maple* (©Maplesoft). También hay una larga lista de programas englobados en el mundo del software libre que se pueden adaptar a este trabajo.

Siempre hay que intentar escoger la herramienta que mejor se adapte al problema que se presenta y, en nuestro caso, *Maxima* cumple con creces las necesidades de un curso de Cálculo. Es evidente que *Mathematica* o *Maple* también pero creemos que el uso de programas de software libre permite al alumno y al profesor estudiar cómo está hecho, ayudar en su mejora y, si fuera necesario y posible, adaptarlo a sus propias necesidades.

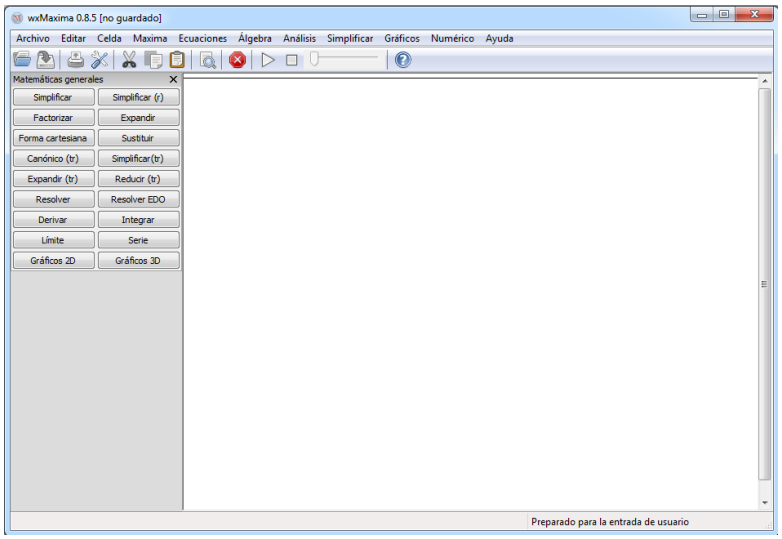
No es nuestra intención hacer una historia de *Maxima*, ni explicar cómo se puede conseguir o instalar, tampoco aquí encontrarás ayuda ni preguntas frecuentes ni nada parecido. Cualquiera de estas informaciones se encuentra respondida de manera detallada en la página web del programa:

`http://wxmaxima.sourceforge.net`

En esta página puedes descargarte el programa y encontrar abundante documentación sobre cómo instalarlo.

Vamos a comenzar familiarizándonos con *Maxima* y con el entorno de trabajo *wxMaxima*. Cuando iniciamos el programa se nos presenta una ventana como la de la Figura. En la parte superior tienes el menú con las opciones usuales (abrir, cerrar, guardar) y otras relacionadas con las posibilidades más "matemáticas" de *Maxima*. En segundo lugar aparecen algunos iconos que sirven de atajo a algunas operaciones y la ventana de trabajo.

El panel de comandos que aparece en la parte izquierda, lo abrimos yendo en el menú a Maxima —> Paneles —> Matemáticas generales. El panel es desplazable a lo largo de toda la pantalla mediante el ratón en la forma habitual de Windows.



Operaciones básicas

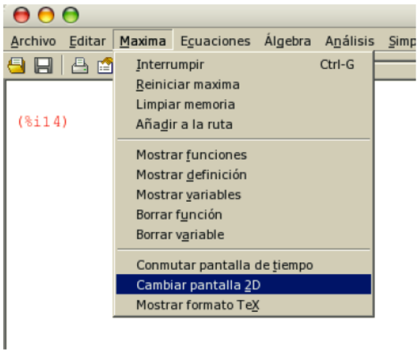
+	suma
∗	producto
/	división
^ o ∗∗	potencia
sqrt( )	raíz cuadrada

Constantes

Además de las funciones usuales, *Maxima* también conoce el valor de algunas de las constantes típicas.

%pi	el número π
%e	el número e
%i	la unidad imaginaria
%phi	la razón aúrea, $\frac{1+\sqrt{5}}{2}$

```
(%i7)   3^1000;
        13220708194808066368904552597
(%o7)   5[418 digits]6143661321731027
        68902855220001
```



Como puedes ver, *Maxima* realiza la operación pero no muestra el resultado completo. Nos dice que, en este caso, hay 418 dígitos que no está mostrando. ¿Se puede saber cuáles son? Sí. Nos vamos al menú **Maxima→Cambiar pantalla 2D** y escogemos **ascii**. Por último, repetimos la operación.

Cálculo simbólico

Cuando hablamos de que *Maxima* es un programa de cálculo simbólico, nos referimos a que no necesitamos trabajar con valores concretos. Fíjate en el siguiente ejemplo:

```
(%i14)  a/2+3*a/5
(%o14)  11a
        ---
        10
```

Como habíamos comentado, nos interesa sobre todo el cálculo simbólico. Pero imaginemos que queremos saber una aproximación decimal de alguna operación, por ejemplo $3\sqrt{2}+25$. Tenemos tres formas fundamentales para hacerlo:

float(*número*)	Expresión decimal de *número*
número,numer	Expresión decimal de *número*
bfloat(*número*)	Expresión decimal larga de *número*

```
(%i4)  3*sqrt(2)+25;
```

(%o4) $3\sqrt{2}+25$

```
(%i1)  float(3*sqrt(2)+25);
```

(%o1) 29.24264068711928

```
(%i2)  3*sqrt(2)+25,numer;
```

(%o2) 29.24264068711928

```
(%i3)  bfloat(3*sqrt(2)+25);
```

(%o3) $2.924264068711929b1$

La última expresión indica que lo que hay antes de la "b", hay que multiplicarlo por 10 elevado al número que hay después (en este caso,1). Se puede cambiar el nº de cifras decimales en **Numérico—>Establecer precisión** (por defecto son 16 cifras decimales).

Por ejemplo, podemos calcular cuánto valen los 100 primeros decimales de π:
Necesitamos la orden `bfloat` para que *Maxima* nos muestre todos los decimales pedidos (y cambiar la pantalla a `ascii`):

```
(%i30)  bfloat(%pi);
(%o30)  3.14159265358979323846264338327950288419716939937510582
        25342117068b0
(%i31)  set_display('ascii)$
(%i32)  bfloat(%pi);
(%o32)  3.14159265358979323846264338327950288419716939937510582
        0974944592307816406286208998628034825342117068b0
```

También podemos poner el programa en modo numérico. Para ello en el menú **Numérico—> Conmutar salida numérica**. Hay que acordarse de volver a cambiarlo si queremos seguir con el cálculo simbólico.

Funciones usuales

Además de las operaciones elementales que hemos visto, *Maxima* tiene definidas la mayor parte de las funciones elementales. Los nombres de estas funciones suelen ser su abreviatura en inglés, que algunas veces difiere bastante de su nombre en castellano.

`sqrt(x)`	raíz cuadrada de x
`exp(x)`	exponencial de x
`log(x)`	logaritmo neperiano de x
`sin(x), cos(x), tan(x)`	seno, coseno y tangente *en radianes*
`csc(x), sec(x), cot(x)`	cosecante, secante y cotangente *en radianes*
`asin(x), acos(x), atan(x)`	arcoseno, arcocoseno y arcotangente
`sinh(x), cosh(x), tanh(x)`	seno, coseno y tangente hiperbólicos
`asinh(x), acosh(x), atanh(x)`	arcoseno, arcocoseno y arcotangente hiperbólicos

```
(%i45)   sin(%pi/4)
(%o45)   1/sqrt(2)
```

Por defecto, las funciones trigonométricas están expresadas en radianes.

```
(%i39)   log(20);
(%o39)   log(20)
```

y si lo que nos interesa es su expresión decimal

```
(%i40)   log(20),numer;
(%o40)   2.995732273553991
```

Observación Mucho cuidado con utilizar `ln` para calcular logaritmos neperianos: ⚠

Otras funciones

Además de las anteriores, hay muchas más funciones de las que *Maxima* conoce la definición. Podemos, por ejemplo, calcular factoriales

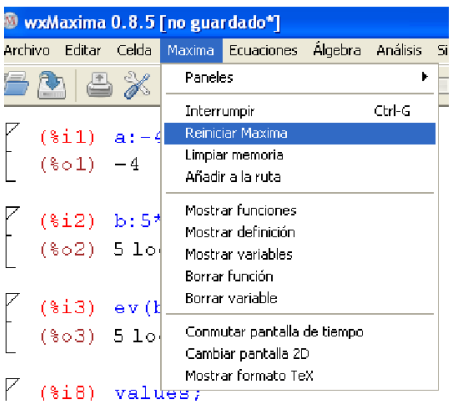
```
(%i48)   32!
(%o48)   263130836933693530167218012160000000
```

Insercción de texto

Podemos comentar resultados, explicaciones etc en Maxima. Para ello vamos a **Celda—> Nueva celda de texto** Nos inserta una celda con fondo verde-azulado donde podemos escribir. También podríamos elaborar un documento con secciones y subsecciones donde Maxima nos los numera automáticamente. Para una celda de sección, hay que ir a **Celda—> Nueva celda de sección**

Reinicio de Maxima

A medida que en una sesión de Maxima vamos definiendo variables, funciones, etc. no basta con borrar las celdas donde están definidas, pues continuan vigentes en memoria, pudiendo llegar a obtener resultados extraños debido a que, por ejemplo, a la variable x le habíamos dado un valor previo y no nos acordamos de vaciarla. Por eso, quizás sea conveniente hacer un reinicio de Maxima y se olvide de todo lo anterior. Para ello, vamos a **Maxima—> Reiniciar Maxima.** Luego conviene ir a **Celdas—> Evaluar todas las celdas.** También podemos limpiar memoria **Maxima—> Limpiar memoria** con parecidos resultados



Variables

El uso de variables es muy fácil y cómodo en *Maxima*. Uno de los motivos de esto es que no hay que declarar tipos previamente. Para asignar un valor a una variable utilizamos los dos puntos

```
(%i61)   a:2
(%o61)   2
(%i62)   a^2
(%o62)   4
```

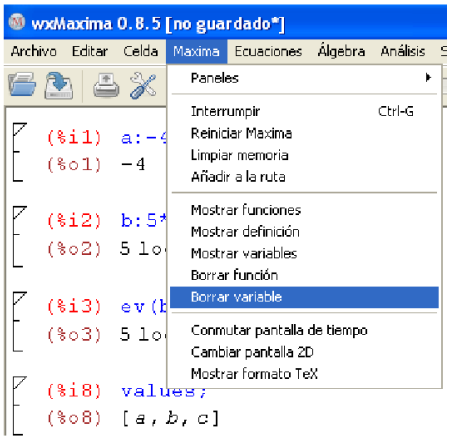
Evaluación de valores en expresiones

expr, [a_1=*valor1*, a_2=*valor2*···]	En *expr* sustituye a_1 por *valor1*, a_2 por *valor2*, ···

```
(%i6)  b:5*log(x)-a^2;
(%o6)  5 log(x)-a^2

(%i7)  b,x=3;
(%o7)  5 log(3)-a^2
```

Desde el menú de Maxima podemos borrar los valores de cualquier variable o incluso todas. Para ello, vamos a **Maxima** —> **Borrar variables** y en la ventana que nos sale escribimos los nombres de las variables a borrar, separadas por comas. Por defecto, las borra todas. También podes ver todas las variables que hay definidas, en el mismo menú en "Mostrar variables".



Simplificación de expresiones

`radcan(`*expr*`)`	simplifica expresiones con radicales
`ratsimp(`*expr*`)`	simplifica expresiones racionales
`fullratsimp(`*expr*`)`	simplifica expresiones racionales

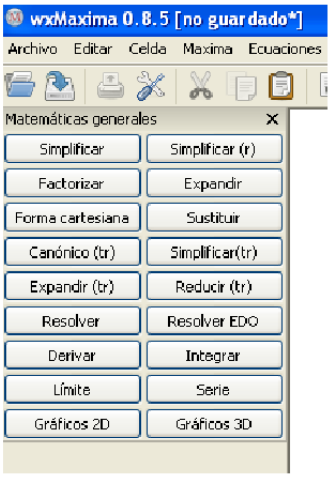
Para simplificar expresiones racionales, `ratsimp` funciona bastante bien aunque hay veces que es necesario aplicarlo más de una vez. La orden `fullratsimp` simplifica algo mejor a costa de algo más de tiempo y proceso.

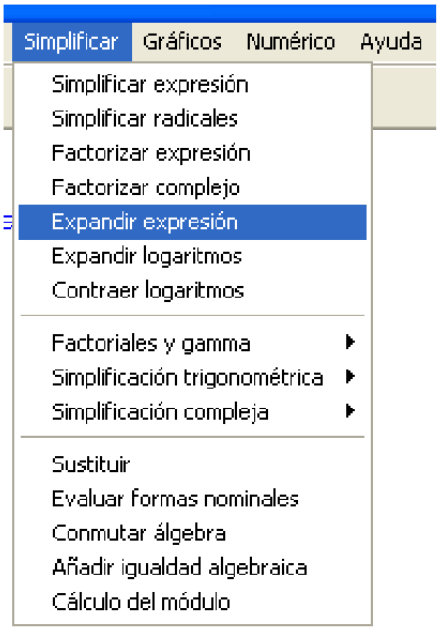
```
(%i30)   fullratsimp((x+a)*(x-b)^2*(x^2-a^2)/(x-a));
(%o30)   x^4+(2a-2b)x^3+(b^2-4ab+a^2)x^2+(2ab^2-2a^2b)x+a^2b^2
```

Para simplificar expresiones que contienen radicales, exponenciales o logaritmos es más útil la orden `radcan`

```
(%i31)   radcan((%e^(2*x)-1)/(%e^x+1));
(%o31)   %e^x-1
```

Todos estos comandos, son accesibles desde el menú de Maxima en **Maxima—> Paneles—> Matemátias generales.** También desde **Maxima—>Simplificar.**
Existen muchos otros comandos de expansión y muchos parámetros para los mismos. Consulte en la ayuda de Maxima si es necesario.





Listas

Maxima tiene una manera fácil de agrupar objetos, ya sean números, funciones, cadenas de texto, etc. y poder operar con ellos. Una lista se escribe agrupando entre corchetes los objetos que queramos separados por comas. Por ejemplo,

```
(%i1)   [0,1,-3];
(%o1)   [0,1,-3]
```

first, second, · · · , last	Primer, segundo,...., último elemento de una lista
lista[k]	k-ésimo elemento de la lista
sort	Ordena los elementos de una lista
length	longitud de la lista

```
(%i4)   lista:[[1,2],1,[3,a,1]]
(%o4)   [[1,2],1,[3,a,1]]
```

```
(%i5)   first(lista);
(%o5)   [1,2]
(%i6)   second(lista);
(%o6)   1
```

```
(%i9)   lista[3];
(%o9)   [3,a,1]
```

También podemos construir una lista a partir de una fórmula:

makelist(*expr, var, n, m*)	Construye una lista variando la variable *var* desde *n* hasta *m* con la expresión *expr*

```
(%i1)   makelist(k^3,k,1,10);
```

(%o1) $[1, 8, 27, 64, 125, 216, 343, 512, 729, 1000]$

La primera lista representa los cubos de los 10 primeros números. Como casi todo, se puede construir una lista desde el menú del Maxima, yendo a **Algebra—>Construir lista**. Nos aparece una ventana donde introducimos los datos

Vectores

Una lista, también podemos considerar que es un vector. En tal caso, podemos efectuar las operaciones habituales: suma, producto por un escalar y producto escalar.

```
(%i15)   v1:[1,0,-1];v2:[-2,1,3];
(%o15)   [1,0,-1]
(%o16)   [-2,1,3]
(%i17)   v1+v2;
(%o17)   [-1,1,2]
```

```
(%i19)   v1.v2;
(%o19)   -5
```

NOTA: Para el producto escalar, debemos utilizar "." Si utilizamos " * " nos multiplica término a término (y no lo suma)

```
(%i18)  v1*v2;
(%o18)  [-2,0,-3]
```

Matrices

Para definir una matriz, lo hacemos con el comando *matrix()* cuyo argumento es una serie de listas, cada una de ellas representa una fila de la matriz.

matrix(*fila1, fila2,* $\cdots$)	Definir la matriz
rank(*matriz*)	Calcula el rango de la matriz
determinant(*matriz*)	Calcula el determinante de la matriz
invert(*matriz*)	Calcula la matriz inversa.

```
(%i33)  A:matrix([1,2,3],[-1,0,3],[2,1,-1]);
        B:matrix([-1,1,1],[1,0,0],[-3,7,2]);
```

Podemos efectuar todas las operaciones habituales sobre matrices: sumas, producto por escalares y producto (usando "."):

```
(%i42)  A+B;
```

$$\begin{bmatrix} 0 & 3 & 4 \\ 0 & 0 & 3 \\ -1 & 8 & 1 \end{bmatrix}$$

```
(%i43)  A-B;
```

$$\begin{bmatrix} 2 & 1 & 2 \\ -2 & 0 & 3 \\ 5 & -6 & -3 \end{bmatrix}$$

El operador * multiplica los elementos de la matriz entrada a entrada.

```
(%i44)  A.B;
```

$$\begin{bmatrix} -8 & 22 & 7 \\ -8 & 20 & 5 \\ 2 & -5 & 0 \end{bmatrix}$$

```
(%i45)  A*B;
```

$$\begin{bmatrix} -1 & 2 & 3 \\ -1 & 0 & 0 \\ -6 & 7 & -2 \end{bmatrix}$$

^^n eleva toda la matriz a *n*, esto es, multiplica la matriz consigo misma *n* veces

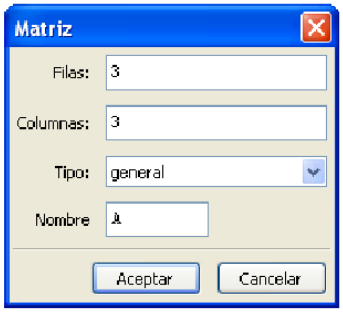
```
(%i46)  A^^2
```

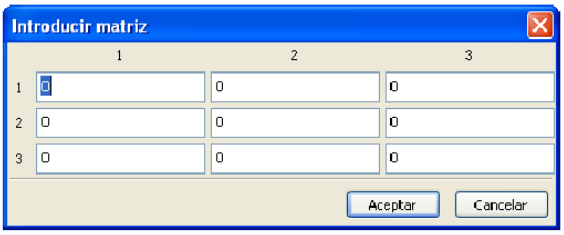
$$\begin{bmatrix} 5 & 5 & 6 \\ 5 & 1 & -6 \\ -1 & 3 & 10 \end{bmatrix}$$

y ^n eleva cada entrada de la matriz a *n*.

```
(%i47)  A^2
                                [ 1  4  9 ]
(%o47)                          [ 1  0  9 ]
                                [ 4  1  1 ]
```

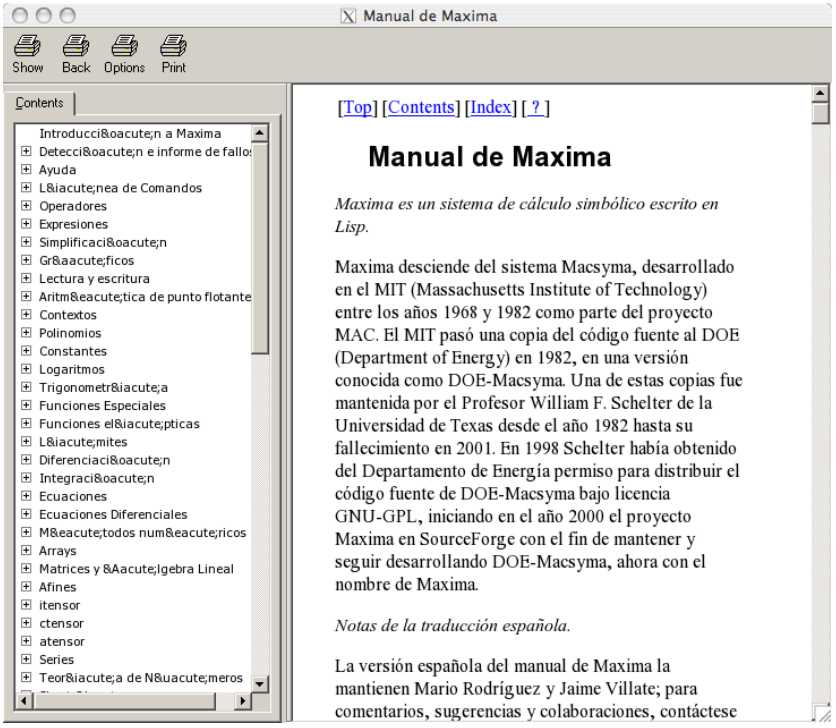
Existen gran cantidad de comandos para matrices, además de los expuesto anteriormente. Consulte la ayuda de Maxima si fuera necesario. Por supuesto, se puede definir una matriz desde el menú de Maxima, en **Algebra—>Introducir matriz**





La ayuda de *Maxima*

El entorno *wxMaxima* permite acceder a la amplia ayuda incluida con *Maxima* de una manera gráfica.



EJERCICIOS PROPUESTOS

1) Calcular $1+\frac{1}{9}+3^{2+4}$.

2) Calcular $\sqrt{4+\sqrt{144}}+\sqrt[3]{27}$.

3) Calcular $\sqrt{1+\sqrt{4}}+\sqrt{2}$.

4) Cacular $e^{\pi i}$.

5) Dar una expresión aproximada de $\sqrt[5]{\pi}$.

6) Calcular $\operatorname{sen}\frac{\pi}{4}+\cos\frac{\pi}{2}+\ln e^4$.

7) Calcular $arctg1+ar\cos(-1)$.

8) Asignar al símbolo *pepe* el valor de 100! y calcular $\frac{pepe}{95!}$.

9) Sustituir en la expresión $pepe+\cos^2 x+sen^2x^2$ la variable x por 0.

10) Sustituir en la expresión $(x+y)^2-x^3$ la variable x por 1 y la variable y por -1.

11) Simplificar la expresión $(x+y)(x-y)-x^2$.

12) Construir con *makelist* una lista con los 30 primeros números impares y de manera que vayan alternando de signo.

13) Dada la matriz

$$A=\begin{pmatrix}1&1&0\\0&1&1\\0&0&1\end{pmatrix}$$

Calcular $(I+A)^3$ y la transpuesta de $(I+A)^3$, siendo I la matriz identidad.