



Práctica 0: INTRODUCCIÓN AL SOFTWARE WXMAXIMA.

En las prácticas de ordenador de la asignatura de Álgebra utilizaremos el programa *Maxima*. En los siguientes apartados pretendemos familiarizarnos con el software y con el entorno de trabajo *WxMaxima*. Se trata de un programa que realiza cálculos matemáticos de forma tanto numérica como simbólica, esto es, sabe tanto manipular números como calcular los autovalores de una matriz. Sus capacidades cubren sobradamente las necesidades de un alumno de un curso de Álgebra en unos estudios de Ingeniería. Es un programa basado en comandos y, al ser éstos fácilmente olvidables, es por lo que usaremos un intérprete del programa: el *WxMaxima* en el que tendremos acceso a la gran mayoría de comandos que necesitaremos mediante simples clics con el ratón.

Podemos encontrar *WxMaxima* en la siguiente dirección::

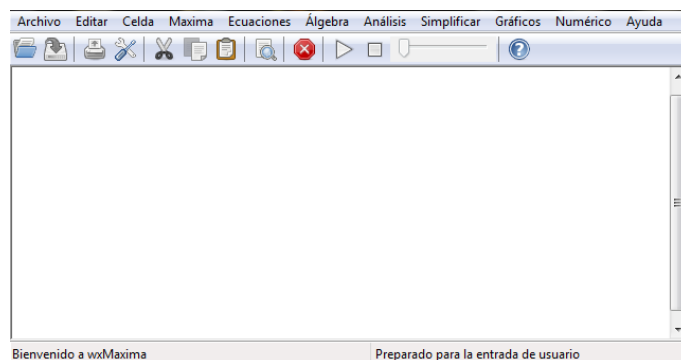
<http://sourceforge.net/projects/maxima/files/>

Sólo hay que pinchar en el enlace de nombre: [Download maxima-5.37.2-gcl.exe](#) (53.5 MB) o en otro similar correspondiente a una versión más reciente.

En esta práctica comenzaremos con el *WxMaxima* haciendo, al principio cosas muy simples que nos ayudarán a conocer los principales comandos y cómo se deben usar. Haremos una revisión general de las capacidades del programa, para posteriormente presentar las principales órdenes del cálculo diferencial de funciones de una variable.

1. Inicio del WxMaxima y Operaciones Básicas

Una vez instalado, al arrancar *WxMaxima*, se nos presenta una ventana como la figura. En la parte superior tenemos el menú con las opciones usuales (abrir, cerrar, guardar, etc) y otras relacionadas con las posibilidades más *matemáticas* del software. En segundo lugar aparecen algunos iconos que sirven de atajo a algunas operaciones y la ventana de trabajo, a la espera de que escribamos una instrucción.



Para ello, escribimos la orden correspondiente y a continuación pulsamos la tecla **Intro** del teclado numérico (o simultáneamente las teclas **Mayúscula** e **Intro**). *WxMaxima* procesará la orden y nos devolverá una respuesta, por ejemplo:

```
(%i1) 2+2;
(%o1) 4
```

Como se puede observar, el programa antepone a la orden que hemos escrito $(2 + 2)$ una etiqueta de la forma $(\%i1)$ (indicando la entrada o input número 1). Del mismo modo, se antepone a la respuesta 4 una etiqueta de la forma $(\%o1)$ (indicando la salida o output número 1). Las siguientes entradas que tecleemos y sus respectivas salidas se etiquetarán como $(\%i2)$, $(\%o2)$, etc. En cualquier momento, podemos hacer referencia a una entrada anterior o a su correspondiente salida, utilizando etiquetas del tipo $\%iN$ u $\%oN$, donde N es el número de entrada/salida al que deseemos acceder. Por ejemplo:

```
(%i2) 333+222;
(%o2) 555
```

```
(%i3) 333-222;
(%o3) 111
```

```
(%i4) %o2+%o3;
(%o4) 666
```

La etiqueta $\%$ siempre hace referencia a la última salida. Como fin de línea, se usa el símbolo $;$ (aunque si usamos wxMaxima no es necesario teclearlo, pues es insertado automáticamente). También podemos utilizar el símbolo $\$$ y en este caso no se presenta en pantalla el resultado, como se puede apreciar en ejemplos posteriores.

Para las operaciones básicas, el programa *WxMaxima* funciona como una simple calculadora, con la salvedad de que con el programa puedes obtener una precisión que no tienen las calculadoras normales. Los operadores aritméticos son los usuales:

+	para la suma	-	para la diferencia
*	para el producto		
/	para la división	^	para la potencia

Por ejemplo:

```
(%i5) 2-5/3;
(%o5) 1/3
```

```
(%i6) 2^10;
(%o6) 1024
```

```
(%i7) 15^%o6;
(%o7) 225151/3
```

Como se puede observar, *Maxima* opera de forma simbólica, utilizando aritmética racional y no realizando aproximaciones numéricas, salvo que se especifique lo contrario. Por defecto, simplifica las fracciones y potencias

numéricas. Con la orden `numer` se pueden obtener expresiones numéricas con 16 cifras:

```
(%i8) 1/2, numer;
```

```
(%o8) 0,5
```

Por último, comentar que si introducimos una operación con números decimales, la respuesta también es dada en números decimales. Veamos la diferencia entre estas dos expresiones:

```
(%i9) 2/3; 2/3.0;
```

```
(%o9)  $\frac{2}{3}$ 
```

```
(%o10) 0,666666666666667
```

1.1. Cálculo simbólico

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

```
(%i11) a/2+3*a/5;
```

```
(%o11)  $\frac{11a}{10}$ 
```

Bueno, hasta ahora sabemos sumar, restar, multiplicar, dividir y poco más. Maxima tiene predefinidas la mayoría de las funciones usuales. Por ejemplo, para obtener la raíz de un número se usa el comando `sqrt`:

```
(%i12) sqrt(5);
```

```
(%o12)  $\sqrt{5}$ 
```

Lo cuál no parece muy buena respuesta. En realidad es la mejor posible: *WxMaxima* es un programa de cálculo simbólico y siempre intentará dar el resultado en la forma más exacta. Si queremos obtener la expresión decimal, utilizamos la orden `float`.

```
(%i13) float(sqrt(5));
```

```
(%o13) 2,23606797749979
```

1.2. Constantes y Funciones usuales

WxMaxima tiene una serie de constantes predefinidas, como π , e ó i (la unidad imaginaria, $i = (0,1)$). Para estas constantes se emplean, respectivamente, los símbolos `%pi`, `%e`, `%i`. Además de las operaciones elementales que hemos visto *WxMaxima* tiene definidas la mayor parte de las funciones elementales. Los nombres de estas funciones suelen ser abreviatura en inglés. Por ejemplo, ya hemos visto la raíz cuadrada y otras se muestran en la siguiente tabla:

<code>sqrt(x)</code>	Raíz cuadrada de x
<code>exp(x)</code>	Exponencial de x
<code>log(x)</code>	Logaritmo neperiano de x
<code>sin(x), cos(x), tan(x)</code>	Funciones trigonométricas
<code>%pi</code>	El número π
<code>%e</code>	El número e
<code>%i</code>	Unidad imaginaria

Ejercicio 1. Realice las siguientes operaciones con números reales y con números complejos:

a) $1 + 3,5$

c) $3 \left(1 + \frac{1}{3}\right)$

e) i^2

b) $2^{3,5}$

d) π^3

f) $(4 - i)/(2i)$

1.3. Insertar comentarios

No se deben escribir comentarios en una celda de entrada, pues, si ejecutamos dicha celda se producirán errores que pueden dar lugar al bloqueo del programa. Cuando queramos insertar alguna explicación o comentario debemos ir al menú *Celda* y escoger *Nueva celda de texto*. Esto crea una celda de comentario que no se ejecutará. Las celdas de comentario son necesarias, por ejemplo, si queremos ejecutar varias celdas a la vez.

2. Variables y Asignar valores

El uso de variables es muy fácil y cómodo en *WxMaxima*. Uno de los motivos de esto es que no hay que declarar tipos previamente. El operador `:` se utiliza para asignar a una variable el valor de una expresión (el signo `=` no se utiliza para asignación, sino para ecuaciones). Por ejemplo, si queremos guardar el valor 3 en la variable `a` escribimos:

```
(%i14) a:3;
```

```
(%o14) 3
```

```
(%i15) a^2;
```

```
(%o15) 9
```

Cuando una variable tenga asignado un valor concreto, a veces diremos que es una constante, para distinguir del caso en que no tiene ningún valor asignado. El nombre de una variable puede ser cualquier cosa que no empiece por un número. Puede ser una palabra, una letra o una mezcla de ambas cosas.

```
(%i16) largo:10;
```

```
(%o16) 10
```

```
(%i17) ancho:7;
```

```
(%o17) 7
```

```
(%i18) largo*ancho;
```

```
(%o18) 70
```

Los valores que asignamos a una variable no se borran por sí solos. Siguen en activo mientras no los cambiemos o comencemos una nueva sesión de Maxima. Es posible que dar un valor a una variable haga que una operación posterior nos de un resultado inesperado o un error. Para borrar el contenido de una variable se usa el comando `kill`:

```
(%i19) kill(a);
```

```
(%o19) done
```

Se pueden borrar varias variables a la vez separándolas por comas: `kill(x,y)`. También se pueden borrar todas las variables de una vez con `kill(all)`.

Ejercicio 2. Utilice las variables `a1,a2,a3,a4,a5,a6` para almacenar, respectivamente, los valores de la expresión $\frac{1}{0!} + \frac{1}{1!} + \frac{1}{2!} + \dots + \frac{1}{n!}$ para $n = 1, \dots, 6$.

3. Expresiones simbólicas

Hasta ahora sólo hemos usado el *WxMaxima* como una calculadora muy potente, pero prácticamente todo lo que hemos aprendido puede hacerse sin dificultad con una calculadora convencional. Entonces, ¿qué puede hacer *WxMaxima* que sea imposible con una calculadora? Bueno, entre otras muchas cosas que veremos posteriormente, la principal utilidad es el cálculo simbólico, es decir, el trabajar con expresiones algebraicas (expresiones donde intervienen variables, constantes... y no tienen por qué tener un valor numérico concreto) en vez de con números.

La capacidad de *WxMaxima* para trabajar con expresiones es notable. Comencemos con funciones sencillas. Consideremos el polinomio

```
(%i20) p:(x+2)*(x-1);
```

```
(%o20) (x-1)(x+2)
```

lo único que hace *WxMaxima* es reescribirlo. ¿Y las potencias?

```
(%i21) q:(x-3)^2;
```

```
(%o21) (x-3)^2
```

Vale, tampoco desarrolla el cuadrado. Probemos ahora a restar las dos expresiones:

```
(%i22) p-q;
```

```
(%o22) (x-1)(x+2)-(x-3)^2
```

El software no factoriza ni desarrolla automáticamente: debemos decirle que lo haga. Para ello se utiliza la orden `expand`.

```
(%i23) expand(p);
```

```
(%o23) x^2 + x - 2
```

```
(%i24) expand(p/q);
```

```
(%o24) x^2/(x^2 - 6x + 9) + x/(x^2 - 6x + 9) - 2/(x^2 - 6x + 9)
```

La orden `factor` realiza la operación inversa a `expand`. La podemos utilizar tanto en números como con expresiones polinómicas.

```
(%i25) factor(x^2-1);
```

```
(%o25) (x - 1)(x + 1)
```

Ahora que hemos estado trabajando con expresiones polinómicas, para evaluar en un punto podemos utilizar la orden `ev`. En su versión más simple, esta orden nos permite dar un valor en una expresión:

```
(%i26) x^2+ev(2*x,x=3);
```

```
(%o26) x^2 + 6
```

4. Representación gráfica

La representación gráfica de una función en *Wx-Maxima* depende de si la función es de variable real o de varias variables. Si es una función real de variable real, $y = f(x)$, se utilizará la orden `plot2d`, con argumentos la propia función, la variable y el intervalo donde la queremos representar. Por ejemplo, si queremos representar la función $f(x) = x^2 + 2 \cos x$, tendremos:

```
(%i27) f(x):=x^2+2*cos(x);
```

```
(%o27) x^2 + 2 cos x
```

```
(%i28) plot2d(f(x),[x,-1,1]);
```

5. Ayuda de WxMaxima

El entorno *WxMaxima* permite acceder a la amplia ayuda incluida con *Maxima* de una manera sencilla y que puede ser de gran utilidad para ver qué es lo que hace un comando. Hay dos formas de inspeccionar la ayuda:

- Escribiendo en una celda de entrada ?? comando.
- En el menú *Ayuda* y seleccionando *Ayuda de Maxima* e *Índice*, veremos todas las ordenes del software.

Muchas veces es mejor un ejemplo sobre cómo se utiliza una orden que una explicación teórica. Esto lo podemos conseguir con la orden **example**.

```
(%i29) example(matrix);
```