

INGENIERÍA TÉCNICA INDUSTRIAL - Esp. MECÁNICAFUNDAMENTOS MATEMÁTICOS DE LA INGENIERÍAPráctica nº 1: Elementos básicos de MAPLE

RESUMEN: *En esta práctica presentamos los elementos fundamentales del software de cálculo simbólico MAPLE. Aprenderemos a introducir las órdenes y configurar un fichero de trabajo. Nos familiarizaremos con el entorno de trabajo y los menús desplegados. Las operaciones básicas, así como la definición de constantes, variables y funciones, son también estudiadas. Concluiremos con una breve introducción a los gráficos en dos dimensiones.*

COMANDOS MÁS IMPORTANTES:**evalf, plot**

* * *


MAPLE es un *software* de **cálculo simbólico** que permite realizar un enorme volumen de operaciones matemáticas donde comparezcan no sólo números sino expresiones que envuelvan variables.


- GENERALIDADES

- Los documentos de MAPLE se denominan hojas de trabajo o worksheets. Se archivan con la extensión .mws.
- Las líneas de órdenes de MAPLE deben empezar por el símbolo [**>**]. Una línea de órdenes se crea en la línea inmediata posterior pulsando el botón "**>**" de la barra de herramientas. La línea de órdenes la transformamos en una línea de texto pulsando el botón "**T**" de la barra de herramientas.
- Las órdenes o sentencias de MAPLE las escribe el programa en color rojo. El texto que podemos escribir en una hoja de cálculo puede configurarse como en cualquier procesador de texto en su tipo de letra, su tamaño, su alineación, etc.
- Cuando introducimos una orden, **no olvide colocar ";" o ":" al final**. En caso contrario no se ejecuta al pulsar Intro ya que supone que aún no ha terminado de completar la orden. El uso de ":" no muestra el resultado de la orden introducida y es muy útil cuando se encadenan varias órdenes y sólo interesa el resultado de la última de ellas. Para crear una nueva línea dentro de una línea de órdenes, y así separar las distintas órdenes que queremos que se ejecuten juntas, tenemos que usar **MAYÚSCULA+INTRO**.

- La ayuda sobre cualquier orden está disponible dentro del menú Help, con el submenú Topic Search.
- MAPLE distingue entre mayúsculas y minúsculas, por lo tanto, **a** y **A** son variables distintas.
- La orden **restart** borra toda la memoria de Maple. Es idóneo para aquellas sesiones donde hemos perdido la referencia de qué valores se han almacenado y está manejando el programa. Al ejecutarla debemos ejecutar todas las instrucciones introducidas de nuevo.
- El operador % permite recuperar el último resultado obtenido. El símbolo %% designa la penúltima salida.
- Las líneas de comentarios comienzan con #. Maple no lee después de este símbolo hasta una línea nueva.
- Se puede interrumpir la ejecución del programa pulsando el botón **STOP** de la barra de herramientas.
- Para borrar una línea de órdenes o un párrafo nos situamos en la región que queremos eliminar y pulsamos **CTRL+SUPR**.
- Siempre podemos deshacer la última operación hecha con las teclas deshacer estándar de Windows.
- Para salir de Maple, puede teclear la orden **QUIT** en la línea de comandos de Maple o bien hacer click en **EXIT** del menú **FILE**.

• Podemos hacer secciones

 esto es una sección

 esto es una subsección

Para hacer secciones o subsecciones, abrir el menú **Insert** y elegir la opción **Section** o **Subsection**.

CONCEPTOS BÁSICOS

Operaciones básicas

Esta es una lista de las operaciones aritméticas básicas disponibles en MAPLE

a + b	suma	a - b	resta
a * b	multiplicación	a / b	división
a ^ b	exponenciación	n !	factorial
abs(n)	valor absoluto	Pi	$\pi=3.14..$
infinity	∞		
min(a,b)	mínimo	max(a,b)	máximo

El formato que se utiliza para el número *e* lo veremos más adelante.

Muchas de las operaciones en MAPLE suelen dar en primera instancia su valor numérico exacto; esto implica que muchas veces nos presenta la representación racional o con radicales si dicho valor tiene infinitos decimales. Para obtener una representación decimal truncada debemos usar la orden

evalf (expresión, n° de dígitos)

donde el número de dígitos si no se indica otra cosa es 10.

```
[ > 1/3;  
  evalf(%);
```

```
[ > 22/79+34/23;  
  > 22.0/79+34/23;  
  > evalf(22/79+34/23);
```

¿Qué diferencias hay entre los últimos tres resultados?

```
[ > 100!;  
  > Pi;  
  > evalf(%);  
  > evalf(%%,40);
```

Debemos comentar que las operaciones tienen un rango de prioridad que es el siguiente:

1º: ^ 2º: *, / 3º: +, -

```
[ > 2*3^2;
```

no es lo mismo que

```
[ > (2*3)^2;
```

```
[ > 1+x/3;
```

no es lo mismo que

```
[ > (1+x)/3;
```

Funciones básicas

Raíz cuadrada: *sqrt(x)*

Computa la raíz cuadrada del valor x siendo este argumento un número (real o complejo) o una expresión algebraica

```
[ > sqrt(2);  
  evalf(%);
```

Función exponencial: *exp(x)*

Calcula el valor de $e = 2.718281828\dots$ elevado a la potencia x .

```
[ > exp(2);  
  evalf(%);
```

De esta forma, el número e se representa en MAPLE por *exp(1)*:

```
[ > exp(1);  
  evalf(%);
```

Función logaritmo: *ln(x)*, *log[b](x)*

El logaritmo natural **ln** es el logaritmo con base $e = 2.71828\dots$. Para $x > 0$:

$$\ln(x) = y \iff x = \exp(y).$$

De forma general, para $x > 0$ y $b > 0$ se tiene

$$\log[b](x) = y \iff x = b^y.$$

El valor por defecto de la base b es $\hat{e} = \exp(1)$, o sea, $\log(x) = \ln(x)$.

```

> log(1);
> ln(1);
> log[10](3);
> evalf(%,20);

```

- Funciones trigonométricas y sus inversas:

sin(x)	cos(x)	tan(x)	sec(x)	csc(x)	cot(x)
arcsin(x)	arccos(x)	arctan(x)	arcsec(x)	arccsc(x)	arccot(x)

Los argumentos de todas las funciones deben ser dados en radianes (1 radián = 180/π grados).

```

[ > sin(Pi/4);
  > tan(Pi/2);

```

CÁLCULOS SIMBÓLICOS

- Definición de variables

Podemos asignar un nombre a operaciones realizadas para utilizarlas posteriormente usando dicho nombre asignado.

Todo resultado obtenido puede ser llamado en la operación inmediata por medio del símbolo %. Esto permite usar un resultado antes de ser obtenido. Esta utilidad es la recomendada para un uso inmediato del valor obtenido, pero si lo que queremos es hacer un uso continuado a lo largo de la sesión de dicho valor, lo mejor es definirlo como una variable. Lo que hacemos es asignarle un nombre por medio del formato de definición de variables

nombre de variable := expresión de Maple

```

[ >a:=22/79 + 34/23;
  evalf(%);
[ > a^2;
  evalf(%);
[ > 1/a;
  evalf(%);
[ > sqrt(a);
  evalf(%);

```

Una posibilidad para hacer nuestras sesiones más claras es introducir comentarios en nuestras órdenes por medio del símbolo # a continuación de la orden introducida en sólo un renglón. MAPLE no lee después de este símbolo hasta el nuevo renglón.

```

[ > capital:=1000; #Valor total de ingresos
  comision:=capital*0.2; #20% del capital total

```

Veamos ahora cómo el orden en que se introducen las variables condiciona el cómo pueden usarse.

```

[ > Precio:=4.95;

```

```
[ > Costo:=2.80;
  > Beneficio:=Precio - Costo;
```

Lo que hacemos ahora es cambiar el valor de la variable *Costo* y ver el valor que toma *Beneficio* después de esta nueva asignación.

```
[ > Costo:=1.2; Beneficio;
```

Vemos que *Beneficio* no ha cambiado su valor porque la definición que hicimos de él fue numérica al haber introducido antes los valores que lo definen.

Hagamos el proceso al revés: primero limpiamos la memoria con **restart**, definimos *Beneficio* en función de las variables *Precio* y *Costo* y luego damos distintos valores a estas variables.

```
[ > restart; #Con esta orden borramos las asignaciones anteriores
  Beneficio:=Precio - Costo; #Definimos primero la variable que queremos
  calcular
  > Precio:=2.0;Costo:=1.5;Beneficio;
```

Ahora bien, en este caso, podemos cambiar el valor de las variables *Precio* y *Costo* y obtener un nuevo *Beneficio*

```
[ > Precio:=3.0;Costo:=1.5;Beneficio;
```

También se puede limpiar el valor de una variable sin cambiar la otra usando la sentencia siguiente:

```
[ > Precio:='Precio'; #son los apóstrofes rectos que están en la misma
  tecla que el símbolo ?
```

Así si subimos el *Precio* manteniendo el mismo *Costo*, obtenemos un nuevo *Beneficio*:

```
[ > Precio:=5.5; Beneficio;
```

Definición de funciones

La sección anterior nos ha dado pie a definir funciones donde vamos a dar sentido en su definición a la variable independiente y la variable dependiente.

Una función de una variable independiente x se define por la relación

$$f := x \rightarrow \text{expresión de } x \quad (1)$$

así podemos hacer referencia ahora a $f(x)$ y evaluarla directamente dando valores a x en esta expresión

```
[ > f:=x->x^2+2*x+1;
  f(x);
  f(1);
  f(-1);
```

Definamos ahora otra función a partir de esta $f(x)$ y veamos lo que significa el formato de definición de funciones.

En primer lugar hacemos una definición como si fuese una variable y vemos que la evaluación no funciona como queremos:

```
[ > g:=3*f(x)+f(x)^2+sin(x);
  g(2);
```

Hacemos ahora una definición de la misma operación con $f(x)$ pero con el formato (1)

```
[ > G:=x->3*f(x)+f(x)^2;
  G(2);
```

Podemos definir cualquier tipo de función a partir de las operaciones y funciones básicas incorporadas en MAPLE

```
[ > f:=x->sin(3*x^5-7*x^3+6)+(ln(x^2+3))^3;
```

```
f(0);
> f(0.0);
> evalf(f(0));
```

Podemos definir igualmente funciones de varias variables

```
> g:=(x,y)->sin(x^2+y^2);
g(1,0);
g(0.5,0.5);
```

- GRÁFICOS

Presentamos en esta sección una breve introducción a los gráficos con MAPLE. En sucesivas prácticas iremos incorporando una mayor profundidad en este apartado, que es de gran utilidad. MAPLE permite realizar gráficos bidimensionales y tridimensionales. Además, podemos representar ecuaciones explícitas, implícitas, paramétricas y conjuntos de datos. También podemos representar funciones con discontinuidades. En esta práctica nos centraremos en los gráficos bidimensionales (2D).

- Comando y opciones

Para generar gráficos bidimensionales (2D) de funciones, esto es, la gráfica de funciones de una variable independiente de la forma $y=f(x)$, usaremos la orden:

```
plot({función_1,función_2,...,función_n}, x=x0..x1, y=y0..y1,
title=`título de la gráfica`, color=[color 1,...,color n],
thickness=[grosor 1,...,grosor n], linestyle=[tipo1,...,tipo n],
discont=true)
```

donde

{función_1,función_2,...,función_n}	son las funciones a dibujar
x=x0..x1	es el rango de la variable indep. que
queremos representar (obligatorio)	
y=y0..y1	es el rango de la variable dependiente que
queremos representar (opcional)	
title=`título de la gráfica` el tipo de comillas ``)	es el título que queremos del gráfico (observar
color=[color 1,...,color n] ejemplo: red, blue, green, yellow,...)	son los colores de los gráficos (en inglés, por
thickness=[grosor 1,...,grosor n] por defecto es 0)	es el grosor de las distintas gráficas 0,1,2,3 (
linestyle=[tipo1,...,tipo n] 3=segmentos, 4=seg.+puntos	tipo de línea: 0=continua, 1=puntos, 2=trazos,
discont=true de haberlas	opción para reconocer discontinuidades en caso

Se pueden quitar opciones menos el rango de la variable.

- Ejemplos

```
[ > plot(sin(x), x=-Pi..Pi);
```

```
[ > plot({sin(x), cos(x)}, x=-Pi..Pi, title=`Funciones y=sen(x) e  
y=cos(x)`, color=[red, blue], thickness=[1,3], linestyle=[0,4]);
```

Todo gráfico puede ser ampliado o reducido en tamaño sin más que pincharlo con el ratón y rescalarlo con el marco que lo rodea pinchando las marcas de las esquinas y los lados. Si pinchamos un punto de cualquiera de las gráficas se señala en la ventana superior izquierda, justo debajo de la barra de herramientas, las coordenadas del punto marcado. Al seleccionar el gráfico aparece la barra de herramientas de gráficos que permite cambiar: el formato de las líneas de gráfico, la presencia o no de los ejes y su ubicación y la elección de la misma escala para ambos ejes (1:1).

Veamos ahora qué hace la orden **plot** sin la opción `discont=true` ante una discontinuidad de salto infinito

```
[ > plot(1/(x-1), x=0..2, y=-20..20, title=`y=1/(x-1)`, color=black );
```

veamos ahora qué representa con la opción `discont=true`

```
[ > plot(1/(x-1), x=0..2, y=-20..20, title=`y=1/(x-1)`, discont=true,
  color=blue);
```

Veamos otra forma de dibujar varias funciones en un mismo gráfico. Para ello almacenamos los plots de cada una de las funciones en variables. Después usamos el comando `display` contenido en el paquete `plots` para dibujarlas. Por ejemplo se van a dibujar la función $\sin(x)$ en rojo y $\cos(x)$ en azul, en el intervalo $[-\pi, \pi]$.

```
[ > with(plots) :
  > f1:=plot(sin(x), x=-Pi..Pi, color=red) :
  > f2:=plot(cos(x), x=-Pi..Pi, color=blue) :
  > display({f1, f2}, title=`Gráfica de funciones trigonométricas`);
  > F:=plot(cos(x), x=-Pi..Pi, y=-Pi..Pi, style=line, color=black) :
  G:=plot(tan(x), x=-Pi..Pi, y=-Pi..Pi, style=point, color=orange) :
  display({F, G}, axes=boxed, scaling=constrained, title=`Coseno y
  tangente`);
```