

# Elementos Básicos de Maple

José Luis Torres Rodríguez\*

Enero 2011

Una vez que hemos iniciado una hoja de trabajo podemos comenzar a insertar regiones de entrada y comandos para su ejecución. Existe un conjunto de elementos básicos de Maple, en base a los cuales se construyen las instrucciones que ejecuta. Entre ellos se encuentran los números, las constantes, cadenas y nombres asignados a expresiones. Así mismo, existen varias reglas sintácticas que deben tenerse en cuenta al teclear dichas instrucciones. Todo esto es importante ya que son los fundamentos para poder realizar operaciones en este sistema.

A continuación se presentan los principales elementos y reglas que se deben considerar al desarrollar una hoja de trabajo en Maple.

## 1. Mayúsculas y minúsculas

Antes de comenzar a introducir instrucciones cabe señalar que Maple hace distinción entre mayúsculas y minúsculas. Así, la palabra *“solve”* es diferente de *“Solve”* y de *“SOLVE”*. Se debe tener esto en cuenta al manipular regiones de entrada, pues es precisamente la información tecleada en estas regiones la que se interpreta.

## 2. Terminadores de instrucciones

Una instrucción en Maple está formada por una o más expresiones terminadas con un símbolo que indica su final. Este sistema utiliza dos símbolos como *“terminadores”* de instrucciones de entrada, el punto y coma “;”, y los dos puntos “:”. Toda instrucción debe ser terminada con uno de los dos símbolos; éstos también pueden ser utilizados para separar dos o más instrucciones que sean introducidas en una misma línea. Ambos terminadores tienen casi la misma función salvo por una diferencia muy importante:

- **Punto y coma (;)**. Al utilizar este terminador en una instrucción, una vez que la ejecutamos, Maple nos muestra inmediatamente el resultado de dicha instrucción.
- **Dos puntos (:)**. Al utilizar los dos puntos, Maple ejecuta la instrucción pero no nos muestra ningún resultado. Esto puede ser bastante útil cuando se realiza un cálculo cuyo resultado es demasiado grande o simplemente éste no se desea visualizar (por ejemplo, si solo se desea conservar su valor en una variable).

Es importante mencionar que siempre se debe colocar uno de estos caracteres como terminador, de lo contrario Maple nos mostrará un mensaje de error debido a que no pudo determinar el final de la instrucción. A continuación veremos algunos ejemplos de su uso:

```
> 3*5; # En esta instrucción, Maple nos muestra el resultado
```

15

---

\*Coordinación de Cómputo, Facultad de Ciencias, UNAM

```

> 3*5: # Se ejecuta la instrucción pero no se despliega el resultado
> 3*5 # Aparece un mensaje de error debido a la falta de terminador
Warning, premature end of input
> 3 + 2; 2 - 4; 4*5: # Estos símbolos también se usan como separadores
> de instrucciones

```

$$5$$

$$-2$$

### 3. Referencia a resultados anteriores

Podemos hacer referencia al resultado de la última instrucción ejecutada usando el símbolo “%”. De la misma forma se puede invocar el penúltimo y antepenúltimo resultado con los símbolos “%%” y “%%%”, respectivamente. Por ejemplo:

```

> a + b;

```

$$a + b$$

```

> % + 1; # el último resultado obtenido, más 1

```

$$a + b + 1$$

```

> %% + 2; # el penultimo resultado más 2

```

$$a + b + 2$$

```

> %%% + 3; # el antepenúltimo resultado más 3

```

$$a + b + 3$$

```

> %% + %%%; # el penúltimo resultado más el antepenúltimo

```

$$2a + 2b + 3$$

**Nota:** Al utilizar el operador “%” se debe tener en cuenta que Maple reevalúa la última expresión ejecutada. De la misma forma, los operadores “%%” y “%%%” reevalúan la penúltima y antepenúltima expresión, respectivamente. Esto debe tomarse en cuenta cuando se ejecutan expresiones complejas ya que al referenciarlas a través de los operadores mencionados se tendrán que volver a evaluar.

Una manera de evitar estos cálculos (muchas veces innecesarios), es haciendo uso de variables para guardar los resultados, ya que al referenciarlas no es necesario volver a calcular el valor que tienen asignado.

Maple no nos permite, mediante el uso de estos operadores, hacer referencia a un resultado antes del antepenúltimo. Esto solo puede hacerse volviendo a ejecutar la instrucción deseada o bien usando variables, las cuales se tratarán posteriormente.

### 4. Tipos de datos básicos

Maple trabaja con diferentes tipos de datos, entre los cuales se encuentran los siguientes:

## 4.1. Números

Cualquier dato formado por un valor numérico. Maple reconoce los siguientes tipos de números:

- Enteros y racionales

```
> 34; 123; 28;
```

$$\begin{array}{l} 34 \\ 123 \\ 28 \end{array}$$

```
> 1/4; 2/5; 6/3*3/8;
```

$$\begin{array}{l} \frac{1}{4} \\ \frac{2}{5} \\ \frac{3}{4} \end{array}$$

Acercas de los racionales, cabe mencionar que Maple ejecuta, de manera predeterminada, todas las operaciones usando aritmética racional. Esto es, si se le pide calcular la siguiente operación:

```
> 23/4*(12/5 - 24);
```

$$\frac{-621}{5}$$

Nótese que en ningún momento se realizan las divisiones, todo el cálculo se hace en forma racional. Esta característica de Maple evita los problemas que puedan surgir al hacer redondeo de los operandos. Por ejemplo, considerese la siguiente instrucción:

```
> 1/3*25!;
```

$$5170403347776995328000000$$

Este resultado se obtiene al operar usando racionales. Ahora introducimos la misma operación pero realizando la operación  $1/3$  y calculando la diferencia entre los resultados. Para ello usaremos la función **evalf**, la cual recibe una expresión aritmética y devuelve su valor como número de punto flotante (vease su hoja de ayuda ejecutando **?evalf**).

```
> evalf(1/3)*25!;
```

$$0,5170403346 \cdot 10^{25}$$

```
> %% - %;
```

$$0,2 \cdot 10^{16}$$

Nótese que la diferencia que existe entre la operación con aproximación ( $1/3 = .333333333\dots$ ) y la operación racional es considerablemente grande. Esta es la razón por la cual Maple no maneja los números como reales, a menos que se le obligue a hacerlo. Posteriormente se tratarán con más detalle este tipo de operaciones.

- Irracionales. Maple tiene la capacidad de trabajar también con este tipo de números, por ejemplo:

```
> sqrt(sqrt(2)*exp(1));
```

$$\sqrt{\sqrt{2}e}$$

En este caso tenemos una operación que incluye a dos números irracionales,  $\sqrt{2}$  y  $e$  los cuales son manejados de manera simbólica en las operaciones, a menos que se pida una aproximación de punto flotante con **evalf**.

- Números de punto flotante

```
> 18.345^4; 239.734*785.4;
113258,5153
188287,0836
```

Para este tipo de números (y en general para cualquier operación numérica), usando la función **evalf** es posible obtener una aproximación con una precisión arbitraria. Por ejemplo, podemos solicitar el cálculo siguiente con 500 dígitos de precisión:

```
> evalf(sqrt(2)*Pi-exp(1),500);
1,7246011096993210116555935187080312008573745956757306481184279792\
32358162745926316003944756239598809935943820753225731442542\
18492171658391947501206441381939254114649112793404235021983\
12205533268766340079036083607903732562279585178253140936050\
22470406362837069166459625837507891323849788988768162241759\
80200003499552716480762867419919927025826673675952134011885\
29558323211481988601723782556998659909661969755755798449514\
69197203997060003531791738757928968615973404616010629217618\
8652706870884120109392
```

La función **evalf** también puede ser usada para obligar a Maple a devolver el resultado de un cálculo en forma de punto flotante. Otra forma de hacer esto es incluyendo, como parte de la expresión, al menos un número de punto flotante.

```
> 39.0 - 234.0^2 + sqrt(23.8);
-54712,12148
```

- Números complejos e imaginarios. Para hacer uso de estos números, Maple utiliza la constante simbólica  $I$  que representa  $\sqrt{-1}$ . Esta constante se encuentra ya predefinida como parte del sistema.

```
> (2 + 3*I)*(4 - 2*I)*(I);
-8 + 14 I
```

En el caso de los números complejos, estos deben ser encerrados entre paréntesis para poder operarlos correctamente, además la parte imaginaria debe ser expresada como un producto explícitamente.

## 4.2. Constantes

Cualquier tipo de dato formado por un valor constante. Por ejemplo las constantes numéricas o las simbólicas predefinidas, como se muestra a continuación:

- Constantes numéricas. Están formadas por cualquier dato de tipo numérico:

```
> 23; 4.987; (3.5 + 4*I);
23
4,987
3,5 + 4. I
```

- Constantes simbólicas. Maple nos proporciona un conjunto de símbolos predefinidos que pueden ser invocados en cualquier expresión, entre los cuales se encuentran:
  - **Pi**. Representa el número  $\pi$ , cuyo valor es aproximadamente 3.1415926535...  
Este tipo de aproximación puede ser mejorada utilizando la instrucción **evalf**.
  - **I**. Representa la constante  $i = \sqrt{-1}$ , usada para expresar números complejos e imaginarios.
  - **infinity**. Representa el símbolo "*infinito*" o "*infinito positivo*" denotado en matemáticas por  $\infty$ .
  - **-infinity**. Representa el símbolo "*infinito*" o "*infinito negativo*", denotado por  $-\infty$
  - **gamma**. Representa a la **Constante de Euler**, comunmente denotada por  $\gamma$ . Su valor aproximado es: .5772156649.
  - **Catalan. Constante de Catalan**. Su valor aproximado es .9159655942.
  - **true, false**. Se utilizan para representar los valores booleanos de falso y verdadero.

Al invocar este tipo de datos debe recordarse que Maple hace distinción entre mayúsculas y minúsculas.

### 4.3. Expresiones

Las "*expresiones*" son uno de los elementos básicos utilizados por Maple. Todas las instrucciones están formadas por expresiones. Como definición podemos decir que "**todo conjunto de caracteres que no viole la sintaxis de Maple es considerado una expresión**". Veamos algunos ejemplos:

#### 4.3.1. Expresiones válidas

En general, las expresiones están formadas por combinaciones alfanuméricas sin espacios o parentesis sin cerrar (es recomendable no usar simbolos como #, &, %, @, !, \). Veamos los siguientes ejemplos:

- Palabras solas que no contienen signos de operaciones aritméticas. Nótese que es válido usar guiones bajos para unir palabras. También puede incluirse el caracter "?", siempre que no se encuentre en la primera posición.

```
> ab001drl3k879uiy_3?;
      ab001drl3k879uiy_3?
> hola_usuarios_de_Maple;
      hola_usuarios_de_Maple
```

En las expresiones también debe tomarse en cuenta que Maple distingue mayúsculas y minúsculas. Por ejemplo, realizemos las siguientes operaciones:

```
> A - A;
      0
> A - a;
      A - a
```

- Nombres con operaciones aritméticas; es decir, combinaciones de nombres alfanuméricos con signos de operación e instrucciones bien definidas (se debe tener cuidado con la correcta colocación de los parentesis asociativos).

```
> alpha - beta + delta - 32899;
      
$$\alpha - \beta + \delta - 32899$$

> sqrt(alpha - beta + delta) + 32777;
      
$$\sqrt{\alpha - \beta + \delta} + 32777$$

> (A + 1)^(A + 1);
      
$$(A + 1)^{(A+1)}$$

> (A + 1)^((A + 1)^(A + 1));
      
$$(A + 1)^{((A+1)^{(A+1)})}$$

> (a + hola_usuarios_de_Maple)/(a - hola_usuarios_de_Maple);
      
$$\frac{a + \text{hola\_usuarios\_de\_Maple}}{a - \text{hola\_usuarios\_de\_Maple}}$$

```

- Expresiones que contienen signos de relación.

```
> ecuacion = 3*a + b - gamma;
      
$$\text{ecuacion} = 3a + b - \gamma$$

> ddd + 1 > 33331;
      
$$0 < \text{ddd} - 33330$$

```

Todas las instrucciones dadas a, y producidas por Maple, están formadas por expresiones que caen dentro de alguno de los tipos descritos anteriormente.

#### 4.3.2. Expresiones NO válidas

A continuación veremos algunos ejemplos de expresiones no válidas en este sistema.

- Expresiones sintácticamente mal escritas, en las que se incluyen comas, comillas, parentesis sin cerrar, numeros y operaciones aritmeticas incompletas. Por ejemplo:

```
> 2,; # La coma no debería aparecer
Error, ';' unexpected
> )ae4n; # El parentesis está mal colocado
Error, ')' unexpected
> g%; # El símbolo% está mal colocado
Error, missing operator or ';'
> ae(11)bcv; # No se pueden colocar parentesis de esta forma
Error, missing operator or ';'

```

- Expresiones que usan incorrectamente símbolos de operaciones y relaciones.

```
> a-;
Error, ';' unexpected

> a^a^a; # Se deben colocar parentesis asociativos
Error, '^' unexpected

> a = b = c; # No se pueden expresar relaciones de esta forma
Error, '=' unexpected
```

- Expresiones formadas por símbolos o palabras reservadas. Por ejemplo, a los nombres de funciones predefinidas y las constantes simbólicas no se les puede asignar un valor diferente al que representan, tales como @, #, %, I, Pi, sqrt, etc. Por ejemplo:

```
> Pi := 23;
Error, attempting to assign to 'Pi' which is protected

> sqrt := 23;
Error, attempting to assign to 'sqrt' which is protected
```

**Nota:** el operador “:=” es utilizado para asignar a una expresión un valor determinado. Este operador sera tratado con detalle más adelante.

### 4.3.3. Expresiones que contienen espacios

Normalmente las expresiones que contienen espacios en blanco no pueden ser usadas como nombres, a menos que se encuentren delimitadas por acentos agudos. Compárese el resultado de las siguientes expresiones:

```
> la solucion general es;
Error, missing operator or ';'

> 'la solucion general es';
```

*la solucion general es*

En el primer ejemplo, Maple genera un mensaje de error pues tenemos una expresión que contiene espacios en blanco y por lo tanto cada palabra se considera como una expresión independiente. En cambio en la segunda expresión, al colocar los acentos agudos, Maple considera a todas las palabras con sus espacios en blanco como una sola expresión. A este tipo de datos delimitados por acentos agudos se les conoce con el nombre de “*cadena*” (“*strings*”), como se verá más adelante.

### 4.3.4. Parentesis asociativos en las expresiones

En todas las expresiones es importante colocar los parentesis necesarios para asociar de manera adecuada. Siempre se deben usar “( )”. Si no existen parentesis, Maple no hace la asociacion de manera automática. Por ejemplo:

```
> a^3/2 <> a^(3/2);
```

$$\frac{a^3}{2} \neq a^{(3/2)}$$

> a/3 + b <> a/(3 + b);

$$\frac{a}{3} + b \neq \frac{a}{3+b}$$

> A + b^3 <> (A + b)^3;

$$A + b^3 \neq (A + b)^3$$

#### 4.4. Cadenas

Las cadenas están formadas por expresiones que no tienen ningún significado especial para Maple. Por ejemplo cadenas de texto (siempre y cuando no sean comentarios). Son conjuntos de caracteres que generalmente se encuentran encerrados por acentos agudos.

```
> 'esta es una cadena'; 'Maple es un programa';
      esta es una cadena
      Maple es un programa
```

#### 4.5. Nombres de variables

Maple permite asignarle un nombre a cualquier elemento. Por ejemplo a una constante numérica, a una gráfica o al resultado de un cálculo. Este nombre debe estar formado por una expresión válida seguida de los caracteres ":=". Por ejemplo:

```
> nombre1 := 34;
      nombre1 := 34

> 'operación 1' := 23*124 - 75^2;
      operacin 1 := -2773

> grafica_1 := plot(x^2, x=-2..2):

> deriv := diff(sin(x^3)*cos(x^8), x);
      deriv := 3 cos(x^3) x^2 cos(x^8) - 8 sin(x^3) sin(x^8) x^7
```

**Nota:** `diff` calcula la derivada de una función. Vease su página de ayuda ejecutando `?diff`.

Este tipo de asignación de nombres se le conoce como definición de variables. Los nombres se consideran un tipo de dato básico, sin embargo las variables serán tratadas más adelante.

Además de los tipos de datos descritos anteriormente, existen algunas estructuras manejadas por Maple. Aunque éstas serán tratadas con detalle posteriormente, vale la pena mencionarlas ahora dado que también pueden ser consideradas como tipos básicos y además forman parte de la sintaxis de una gran cantidad de funciones.



## 4.6. Secuencias

Una secuencia está formada por varias expresiones separadas por comas. Por ejemplo:

```
> a, 3, lista, 0, 1;
```

*a, 3, lista, 0, 1*

Esta secuencia está formada por los elementos "a", "3", "lista", "0" y "1".

Estas secuencias pueden ser usadas, por ejemplo, para evaluar varias expresiones y obtener los resultados en forma de secuencia:

```
> evalf(sqrt(Pi)), 23*45, 4.3/8;
```

1,772453851, 1035, 0,5375000000

## 4.7. Listas

Este tipo de estructura esta formado por secuencias de datos encerrados entre los símbolos "[ ]". Cualquier expresión de cualquier tipo puede formar parte de una lista. Por ejemplo:

```
[1, 2, 3, 4, 5, 6]
```

```
[a, b, c, d, e, f, g]
```

```
['Maple es un sistema para hacer cálculos científicos', 'hola a todos', Pi, sqrt(34)]
```

Una lista puede estar formada incluso por otras listas:

```
[ [a, b, c], [1, Pi, exp(23)], ['cadena 1', 123, 'secuencia 3'] ]
```

## 4.8. Conjuntos

Están formados por secuencias de expresiones delimitadas por los símbolos "{ }". Al igual que las listas, los conjuntos pueden estar formados por datos de cualquier tipo. Por ejemplo:

```
{1, 5, 9, exp(34), Pi^2}
```

```
{abcd, dfer, 'Maple despliega gráficas', 98698769}
```

y, de la misma forma que las listas, los conjuntos pueden contener conjuntos:

```
{{a, g, t, 54325, I}, fkjdshlf, variable1, {grafica1, grafica2, grafica3}}
```

Todos estos tipos de datos pueden ser usados en combinación para crear, por ejemplo, listas de conjuntos, conjuntos de números, conjuntos de listas, listas formadas por conjuntos, números y cadenas, etc. Esto se tratará con más detalle posteriormente.

A continuación se describen otras estructuras manejadas por Maple.

## 4.9. Tablas

Las tablas están formadas por un grupo de datos, cada uno de los cuales tiene asignado un índice para referenciarlo. Estas estructuras son creadas con el comando **table** (vease su página de ayuda ejecutando **?table**). Por ejemplo:

```
> T := table([el1 = 3, el2 = 8, el3 = 6, palabra = hola, num = 987542]);
```

*T := table([el2 = 8, el3 = 6, num = 987542, palabra = hola, el1 = 3])*

Las expresiones colocadas a la izquierda de cada relación son los índices. Para acceder a un elemento debemos utilizar su índice, por ejemplo:

```
> T[num];
```

```
> T[e13];
```

6

Este tipo de datos también serán tratados posteriormente.

#### 4.10. Arreglos

Los arreglos, al igual que las tablas, están formados por datos indexados, pero sus índices solo pueden ser números, no cadenas. Tales estructuras son creadas mediante la instrucción **array** (consúltese su *página de ayuda* ejecutando `?array`). Veamos un ejemplo.

```
> A := array(3..5, [4, 8, 25]);
```

```
A := array(3..5, [
(3) = 4
(4) = 8
(5) = 25
])
```

Para hacer referencia a cada elemento utilizamos su índice:

```
> A[3];
```

4

```
> A[5];
```

25

Estas estructuras también pueden ser manejadas como vectores renglón, como veremos más adelante.

#### 4.11. Subíndices

Es posible usar subíndices en expresiones de Maple, éstos se denotan con los símbolos “[ ]”. Cualquier nombre (o secuencia de nombres) sintácticamente bien definido puede ser usado como subíndice.

```
> S[1];
```

 $S_1$ 

```
> Solucion['caso a=0'];
```

 $S_{\text{solucion}_{\text{caso } a=0}}$ 

```
> S['solucion general'];
```

 $S_{\text{solucion general}}$ 

```
> S[a, b, c];
```

 $S_{a,b,c}$ 

Es posible colocar subíndices a los subíndices en múltiples niveles. Por ejemplo:

```
> S[a][b][c][d]['caso x=0'];
```

 $S_{abcd_{\text{caso } x=0}}$ 

Como puede apreciarse en la instrucción anterior, se colocan los subíndices correspondientes, encerrados entre corchetes, secuencialmente. De esta forma se pueden usar expresiones con subíndices tan complejas como se desee.

## 5. Operadores Aritméticos

Los operadores aritméticos soportados por Maple son los siguientes:

- Suma:  $\mathbf{a + b}$ .
- Resta y números negativos:  $\mathbf{a - b, -d}$ .
- Producto:  $\mathbf{a*b}$ .
- Producto no conmutativo:  $\mathbf{a\&*b}$  (utilizado en operaciones de matrices y vectores).
- Potencia:  $\mathbf{a^b}$  (también:  $\mathbf{a**b}$ ).
- División:  $\mathbf{a/b}$  (también:  $\mathbf{a*b^{-1}}$ ).

Maple 8 permite usar el operador “/” en la forma:  $\mathbf{'/'(a,b)}$  o  $\mathbf{'/'(c)}$ , para expresar las operaciones  $\mathbf{a/b}$  y  $\mathbf{1/c}$ , respectivamente. Nótese que el operador debe estar delimitado por apostrofes.

- Factorial:  $\mathbf{!}$ .

También se puede usar la función **factorial**, vease su página de ayuda ejecutando: **?factorial**.

Con estos operadores es posible usar Maple para realizar cálculos aritméticos sencillos:

```
> 4 + 3; 3*24567; 264/4;
```

```
7
73701
66
```

Una característica importante (y muy poderosa) de Maple es que no solo realiza cálculos de tipo numérico, como las operaciones anteriores; también nos permite realizar operaciones de tipo simbólico. Por ejemplo, podemos ejecutar las mismas instrucciones anteriores pero usando datos simbólicos.

```
> a + b; 3*a; a/b;
```

```
a + b
3 a
a
b
```

Nótese que estas operaciones se llevan a cabo sin importar que uno o varios de los operandos involucrados no tengan un valor numérico asignado.

La precedencia de estos operadores es la usual, sin embargo, siempre que existan dudas o se crea necesario, se pueden usar los parentesis “( )” para asociar las operaciones. Un caso en el que en ocasiones es necesario usar parentesis para asociar las operaciones es en la exponenciación:

```
> a^a^a; # Hacen falta los parentesis
```

```
Error, ‘^’ unexpected
```

```
> a^(a^(a^a));
```

```
a(a(aa))
```

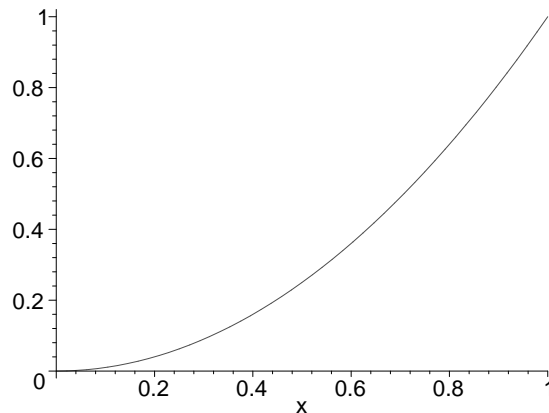
## 6. Parentesis de funciones, operadores e instrucciones

Los símbolos “( )” indican, ya sea de que variables depende una función, o bien, cuales son los argumentos que recibe. Todos los datos contenidos en estos parentesis deben ir separados por comas:

```
> f(a, b, c);
```

$$f(a, b, c)$$

```
> plot(x^2, x=0..1);
```



## 7. Evaluación de expresiones

### 7.1. La función restart

Durante el desarrollo de una hoja de trabajo, al realizar cálculos, definir variables o ejecutar funciones, los resultados generados permanecen en la memoria utilizada por Maple. En las instrucciones ejecutadas posteriormente, estos resultados guardados pueden llegar a causar ciertos problemas al interferir en los nuevos cálculos. Por ejemplo, considerense las siguientes instrucciones:

```
> b := sqrt(23) - 12*Pi;
```

$$b := \sqrt{23} - 12\pi$$

```
> evalf(b);
```

$$-32,90328033$$

¿Qué sucede si a continuación deseamos declarar una ecuación en términos de **b**?

```
> 32^2 + 4*b - b^2 = 0;
```

$$1024 + 4\sqrt{23} - 48\pi - (\sqrt{23} - 12\pi)^2 = 0$$


Nótese que, al tener **b** un valor asignado previamente, Maple considera la instrucción anterior como una ecuación en la que se debe sustituir el valor asignado a esta “**variable**”. De ahí el resultado.

Existe una manera de “*desasignar*” una variable, de tal forma que se elimine todo valor previamente asignado (esto se tratará posteriormente en la sección de variables). Otra forma de hacerlo es indicándole a Maple que debe eliminar todo resultado anterior de la memoria, antes de realizar el siguiente cálculo. Esto es posible a través de la función **restart**. Por ejemplo, si ejecutamos esta instrucción antes de definir la ecuación anterior obtenemos lo siguiente:

```
> restart;
> 32^2 + 4*b - b^2 = 0;
```

$$1024 + 4b - b^2 = 0$$

Nótese que ahora Maple considera a esta expresión como una ecuación que depende de la variable  $b$ , la cual, por cierto, es una variable simbólica, sin ningún valor asignado. Esto se debe a que **restart** eliminó todo valor almacenado en la memoria, en particular toda referencia anterior a una variable con este nombre.

La instrucción **restart** también puede ser invocada por medio del botón , de la barra de herramientas.

## 7.2. Evaluación por sustitución

Existen diferentes formas en las que podemos solicitar a Maple que lleve a cabo la evaluación de una expresión, dependiendo del tipo de ésta y del tipo de resultado que deseamos. Una de estas formas de evaluación consiste en solicitar a Maple que sustituya las variables de la expresión por valores particulares; al hacer esto automáticamente se evalúa la expresión como si ésta fuera de tipo aritmético. Para ilustrar este tipo de evaluación, consideremos el siguiente caso:

La siguiente expresión es conocida como “*ecuación general de una línea recta*”:

```
> y := m*x + b;
```

$$y := mx + b$$

Las variables simbólicas,  $m$  y  $b$ , pueden tomar cualquier valor. La cuestión es como evaluamos la expresión “ $y$ ” en valores particulares de  $m$ ,  $b$ ,  $x$  (por ejemplo  $m = 2$ ,  $b = 3$ ,  $x = 5$ ). La manera de hacer esto es sustituyendo, mediante la instrucción **subs**:

```
subs(m = 2, b = 3, x = 5, y);
```

En general, esta forma de sustituir es válida para cualquier expresión. Si tenemos una expresión  $M$  cualquiera y  $v1$ ,  $v2$ ,  $v3$ , ...,  $vn$  las variables de las cuales depende; las sustituciones de valores particulares de cada una de ellas pueden hacerse de la siguiente forma:

```
subs(v1 = valor1, v2 = valor2, . . ., vn = valorn, M);
```

Es posible sustituir tantas variables como tenga la expresión. Por ejemplo:

```
> y; subs(m = 2, y);
```

$$\begin{aligned} mx + b \\ 2x + b \end{aligned}$$

```
> y; subs(m = 2, b = 3, y);
```

$$\begin{aligned} mx + b \\ 2x + 3 \end{aligned}$$

```
> y; subs(m = 2, b = 3, x = 0.3, y);
```

$$\begin{aligned} mx + b \\ 3,6 \end{aligned}$$

### 7.2.1. Sustitución de variables por otras variables

No solo es posible hacer sustitución de variables simbólicas por números, también se pueden sustituir variables simbólicas por otras variables simbólicas.

Consideremos la siguiente expresión:

```
> M:=(1 + a*x - b*x^2)/((1 + alpha)*(1 + beta));
```

$$M := \frac{1 + ax - bx^2}{(1 + \alpha)(1 + \beta)}$$

Utilizando la función **subs**, podemos sustituir, por ejemplo **alpha = 1 + x**.

> **subs(alpha = 1 + x, M);**

$$\frac{1 + ax - bx^2}{(2 + x)(1 + \beta)}$$

o bien: **beta = 1 + a + x**.

> **subs(beta = 1 + a + x, M);**

$$\frac{1 + ax - bx^2}{(1 + \alpha)(2 + a + x)}$$

También es posible, en una misma instrucción, sustituir todas las variables de las cuales depende una expresión algebraica. De esta forma, podemos hacer que dicha expresión dependa de cualquier variable. Por ejemplo, podemos hacer que **M** dependa solo de **x**:

> **subs(a = x, b = x, alpha = x, beta = x, M);**

$$\frac{1 + x^2 - x^3}{(1 + x)^2}$$

### 7.3. Evaluación simbólica

A veces es necesario hacer que Maple evalúe una expresión, además de hacer una sustitución. Por ejemplo, consideremos la expresión “**y**” definida previamente:

> **y := m\*x + b;**

$$y := mx + b$$

hagamos las siguientes sustituciones:

> **subs(b = cos(u), y);**

$$mx + \cos(u)$$

> **subs(u = 0, %);**

$$mx + \cos(0)$$

Pero sabemos que **cos(0) = 1**, sin embargo Maple no realiza en este caso la evaluación de la expresión tomando en cuenta este valor. Para que Maple evalúe esta expresión para  $u = 0$  es necesario usar la instrucción **eval**.

> **eval(%);**

$$mx + 1$$

En muchos casos, es suficiente usar **subs**, pero en general, es conveniente usar **subs** y **eval** juntos, de la siguiente forma:

> **subs(b = cos(u), y);**

$$mx + \cos(u)$$

> **subs(u = 0, %);**

$$mx + \cos(0)$$

> **eval(%);**

$$mx + 1$$

También es posible combinar las instrucciones **eval** y **subs** en una sola instrucción:

```
> eval(subs(b = cos(0), y));
```

$$m x + 1$$

Este tipo de combinación de funciones (**eval** y **subs**, por ejemplo) son válidas en Maple, siempre y cuando haya compatibilidad en cuanto a los datos que manejan ambas (es obvio que no se le puede pedir a **eval** que realice alguna operación sobre una gráfica, por ejemplo).

La instrucción **eval** se puede utilizar para realizar evaluaciones explícitas en expresiones simbólicas en general, como en el ejemplo anterior.

## 7.4. Evaluación de expresiones pasivas o inertes

Existen varias funciones en Maple que tienen la capacidad de generar, a partir de instrucciones de Maple, expresiones en notación matemática. Este modo de operar se le conoce como forma "*pasiva*" o "*inerte*" de la función. Al ejecutar las instrucciones en este modo, en lugar de obtener un resultado, lo que se obtiene es una expresión en simbología matemática que representa la operación solicitada. Por ejemplo, una de las funciones que pueden trabajar en forma inerte es **limit**, que calcula el límite de una función alrededor de un punto dado. Su sintaxis es:

**limit(función(x) o expresión(x), x=punto);**

Por ejemplo, calculemos el límite de  $\frac{\sin(h)}{h}$ , alrededor de  $h = 0$ .

```
> limit(sin(h)/h, h = 0);
```

1

En este caso, estamos usando la función **limit** en su forma normal, lo cual produce como resultado la evaluación de dicha función y por lo tanto el valor del límite estimado. La forma inerte se obtiene al ejecutar una instrucción (que cuente con una forma inerte en Maple), pero escribiendo el nombre con la primera letra mayúscula. En este caso, la forma inerte de esta función es:

**Limit(función(x) o expresión(x), x=punto);**

Solicitemos nuevamente el mismo cálculo, pero en forma inerte:

```
> Limit(sin(x)/x, x = 0);
```

$$\lim_{x \rightarrow 0} \frac{\sin(x)}{x}$$

Como puede apreciarse, en este caso obtenemos una expresión en simbología matemática que representa la instrucción ejecutada. Este tipo de expresiones pueden ser bastante útiles como se verá más adelante.

Cuando obtenemos un resultado a partir de la forma inerte de una función, es posible hacer que Maple evalúe la expresión generada para obtener el valor de dicha expresión. Esto puede hacerse por medio de la función **value**. Así, podemos obtener el valor numérico del límite simbólico anterior con la instrucción:

```
> value(%);
```

1

En general, **value** nos permite evaluar expresiones que fueron obtenidas en forma "*inerte*".

## 7.5. Evaluación numérica

Este sistema evalúa de manera diferente las expresiones aritméticas, dependiendo del tipo de números que las componen. En general, se realizan evaluaciones de este tipo en dos formas diferentes.

### 7.5.1. Numerós enteros y racionales

Siempre que se introduce una expresión que involucra datos numéricos, Maple automáticamente trata de evaluarla; dicha evaluación, por cierto, generalmente simplifica la expresión. Sin embargo existen ciertas reglas que se siguen al hacer tales evaluaciones.

Si una expresión esta formada solo por números enteros y/o racionales, el resultado será dado con números enteros o racionales. Por ejemplo:

```
> 5!*46 - 23^3*8/5;
```

$$\frac{-69736}{5}$$

```
> (25*x + 1/4*x^2) + (2*x - 5*x^2);
```

$$27x - \frac{19}{4}x^2$$

```
> 23*Pi^2 - 18^3*4^2/(3^2 + 6^4) + 4!;
```

$$23\pi^2 - \frac{6888}{145}$$

Nótese que Maple realiza todas las evaluaciones posibles de tal forma que, en el resultado, no aparezcan números de punto flotante. Por ejemplo, en la última expresión no se evalúa  $\pi^2$ , ni la división  $\frac{6888}{145}$ . En general, esta es la forma en la cual se realizan las evaluaciones con enteros y/o racionales. Este modo de evaluar es diferente al de una calculadora, en la cual los números se convierten automáticamente a punto flotante, pero existe el problema de que en cálculos que involucran números grandes, la conversión a punto flotante puede provocar errores de precisión considerables. En este sentido la forma de evaluar de Maple es más eficiente que la de los programas que trabajan siempre con punto flotante.

### 7.5.2. Numeros de punto flotante

Como ya se mencionó, Maple por omision no realiza las operaciones numéricas en punto flotante. Sin embargo, existen varias formas en las cuales podemos obligar a que los resultados se presenten así. Una de ellas es (como se vio anteriormente) incluyendo como operando al menos un número de punto flotante. Por ejemplo:

```
> 4.2/3^4 - 54^2*4^3;
```

$$-186623,9481$$

Otra forma de obtener resultados de esta forma es usando la instrucción **evalf**, la cual recibe como argumento una expresión aritmética y la evalúa devolviendo un resultado de punto flotante. Su sintaxis es:

**evalf(N, numero de decimales);**

También puede ser invocada en la forma:

**evalf[número de decimales](N);**

Esta función nos permite además obtener aproximaciones de numeros irracionales, tales como  $\sqrt{2}$ , Pi, e; así como realizar cálculos de precisión arbitraria. Por ejemplo:

```
> sqrt(2); # No se obtiene un número de punto flotante
```

$$\sqrt{2}$$

```
> evalf(%); # Esto nos da un número de punto flotante
```

$$1,414213562$$



Por omisión, al realizar operaciones de punto flotante, Maple solo despliega un máximo de diez decimales. **evalf** nos permite obtener una precisión mayor, solicitando una cantidad arbitraria de decimales. Por ejemplo, evaluemos la expresión anterior con 100 decimales.

```
> evalf(%%, 100);
```

```
1,4142135623730950488016887242096980785696718753769480731766797379\  
90732478462107038850387534327641573
```

Esta última instrucción es equivalente a: **evalf[100](% %)**;

Veamos otro ejemplo.

```
> 1.1*sqrt(3) + .1;
```

$$1,1\sqrt{3} + 0,1$$

Este resultado es un número irracional (de hecho es un número exacto pero no nos da una idea clara de su valor). Podemos obtener una aproximación con **evalf**:

```
> evalf(%);
```

$$2,005255889$$

## 8. Operadores relacionales

Los símbolos relacionales que soporta Maple son los siguientes:

- Igual: =
- Menor que: <
- Mayor que: >
- Menor o igual: <=
- Mayor o igual: >=
- Desigual o diferente: <>

Con ellos podemos expresar igualdades (ecuaciones) o desigualdades (inecuaciones). Por ejemplo:

```
> a = b;
```

$$a = b$$

```
> 3 < 4; 3 <> 4;
```

$$3 < 4$$

$$3 \neq 4$$

```
> a <= 0; a >= 0;
```

$$a \leq 0$$

$$0 \leq a$$

Es posible introducir en las regiones de entrada expresiones con símbolos de relación (un solo símbolo de relación por cada instrucción).

```
> a + b^2 + f - (a + b)^2 = c*d*(1 - a*b);
```

$$a + b^2 + f - (a + b)^2 = cd(1 - ab)$$

```
> (3*b - 2*s)/(a - b)*f >= 0;
```

$$0 \leq \frac{(3b - 2s)f}{a - b}$$

## 9. Operadores lógicos

Los operadores lógicos soportados por Maple son:

- Conjunción: **and**
- Disyunción: **or** y **xor** (**or** exclusivo)
- Negación: **not**
- Implicación: **implies**

Posteriormente veremos algunos ejemplos del uso de estos operadores.

## 10. Manipulación de los miembros de una relación

En la sección anterior vimos como expresar relaciones. Una relación está formada por dos expresiones unidas por un operador relacional. En ocasiones es necesario poder referenciar cada uno de sus componentes. Las instrucciones **lhs** y **rhs** permiten extraer los miembros izquierdo y derecho, respectivamente, de instrucciones en las cuales aparece un signo de relación.

```
> a + b = c + 1/d;
```

$$a + b = c + \frac{1}{d}$$

```
> lhs(%);
```

$$a + b$$

```
> rhs(%);
```

$$c + \frac{1}{d}$$

La instrucción **lhs** nos permite extraer el miembro izquierdo de la relación  $a + b = c + \frac{1}{d}$ ; es decir,  $a + b$ . Mientras que el operador **rhs** nos permite extraer el componente derecho,  $c + \frac{1}{d}$ .

## 11. Utilización de letras griegas

Maple nos permite utilizar letras del alfabeto griego en las instrucciones de entrada. Para poder usar una de estas letras basta con escribir su nombre completo. Por ejemplo, para obtener la letra “ $\alpha$ ”, debemos escribir la palabra “*alpha*”. Estas palabras (nombres de letras griegas), automáticamente se convierten en el símbolo correspondiente. Así, podemos introducir expresiones como la siguiente:

```
> alpha + beta^2 + epsilon - (a + b)^2 = c*d*(1 - alpha*beta);
```

$$\alpha + \beta^2 + \varepsilon - (a + b)^2 = cd(1 - \alpha\beta)$$

```
> (3*beta - 2*gamma)/(a - b)*Phi >= 0;
```

$$0 \leq \frac{(3\beta - 2\gamma)\Phi}{a - b}$$

Para poder utilizar letras mayúsculas del alfabeto griego solo debemos colocar su nombre con la primera letra en mayúscula. Por ejemplo, para obtener la letra “ $\Gamma$ ”, debemos escribir la palabra “**Gamma**”, colocando la letra “**G**” mayúscula. Maple interpreta los nombres de letras griegas siempre y cuando éstas sean

introducidas en una región de entrada o en regiones de texto a través de la opción **Standard Math** del menú **Insert**.

Recuérdese que esta versión de Maple nos proporciona varias paletas a través de las cuales podemos introducir expresiones, una de ellas es la “**paleta de símbolos**”, precisamente por medio de ésta podemos introducir letras del alfabeto griego. Para visualizar la paleta debemos solicitar la opción **Palettes - Symbol Palette**, del menú **View**.