

Material de estudio

OCW 2018: Utilizando Mathematica como apoyo al cálculo algebraico en los grados de Ingeniería

Tema 1. Comenzando a trabajar

Equipo docente del curso

*Martín Yagüe, Luis
Unzueta Inchaurre, Aitziber
Arrospide Zabala, Eneko
García Ramiro, María Begoña
Soto Merino, Juan Carlos
Alonso González, Erik*

TEMA 1. COMENZANDO A TRABAJAR CON MATHEMATICA.

Introducción

¿Qué es Mathematica?

Se trata de un *Sistema de Computación Algebraica* (siglas C.A.S. del inglés *Computer Algebraic System*) desarrollado por *Wolfram Inc.* Por lo tanto, es un programa de ordenador que facilita las operaciones matemáticas trabajando con ecuaciones y fórmulas de forma simbólica, en lugar de numéricamente. Por ejemplo, una expresión como $x+y$ es interpretada siempre como "la suma de dos variables" y no como "la suma de dos números". Además de *Mathematica*, otros C.A.S. destacados son *Maple*, *Derive*, *MatLab*, *Maxima* y *MuPad*.

Este tipo de programas suelen incluir manipulaciones tanto con expresiones simbólicas como con expresiones numéricas y diversas representaciones de los resultados.

Debe entenderse el uso del programa *Mathematica* como una herramienta de apoyo en la resolución de los ejercicios clásicos de los diferentes temas que componen el programa del curso.

En este curso se ha utilizado tanto la versión *Wolfram Mathematica 11.2* como la *11.3*, última disponible.

Principales prestaciones

De forma general, se presentan las principales prestaciones del programa. Se irá profundizando a medida que avance el curso.

- Cálculo numérico
 - obtención de radicales

$$\sqrt{10233.45}$$

$$101.161$$

- resolución de una integral definida con una precisión de 6 cifras significativas

$$N\left[\int_1^{\sqrt{13}} \frac{\text{Log}[x]^2}{x^3} dx, 6\right]$$

$$0.118184$$

- Cálculo simbólico
 - simplificación

$$\text{Simplify}\left[\frac{(2 + 2a + b + ab)(a - 1)^2}{(a^3 - a^2 - a + 1)(b^2 + 3b + 2)}\right]$$

$$\frac{1}{1 + b}$$

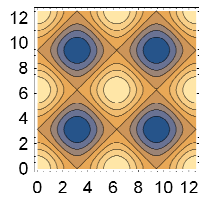
- método de integración por partes

$$\int x^4 \text{ArcSin}[x] dx$$

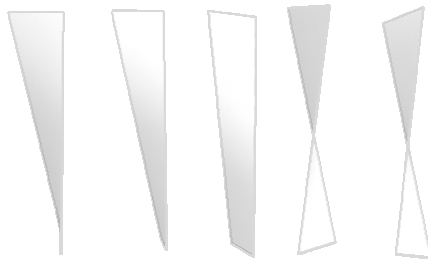
$$\frac{1}{75} \sqrt{1 - x^2} (8 + 4x^2 + 3x^4) + \frac{1}{5} x^5 \text{ArcSin}[x]$$

- Potente representación gráfica en 2 D y 3 D
 - ejemplo obtenido en la *Ayuda* del programa

`ContourPlot[Cos[x] + Cos[y], {x, 0, 4 Pi}, {y, 0, 4 Pi}]`



- ejemplo (distintas perspectivas de una lama de la cubierta exterior del estadio San Mamés en Bilbao)



Conceptos básicos

Estructuración

El programa *Mathematica* se estructura en 2 partes:

- Front End. Interfaz que permite la comunicación con el usuario, presenta las entradas y salidas y permite crear y editar ficheros que contienen texto, cálculos, gráficas, etc. Estos ficheros tienen la extensión **.nb** y se denominan *notebooks* (cuadernos de notas).
- Kernel (núcleo). Constituye la estructura interna del programa y se encarga de la realización de los cálculos y operaciones matemáticas que solicita el usuario.

Celdas

El área de trabajo se divide en casillas llamadas **celdas** (*cells*) que quedan delimitadas por un corchete situado en la parte derecha. Pueden ser, básicamente, de los siguientes tipos:

- *Input* (`In[núm]:=`). Asignación por defecto. Contienen las órdenes (comandos, instrucciones, operaciones, etc.) que el usuario quiere realizar. Para que el programa ejecute la instrucción contenida en una celda *Input* hay que pulsar la tecla <Intro> (o la combinación de teclas: <Shift>+<Enter>). En una celda de este tipo puede insertarse texto, a modo de comentario, siempre que se haga con el siguiente formato: `(* texto del comentario *)`
- *Output* (`Out[núm]=`). Asociada a cada celda *Input* (entrada) el programa genera una celda *Output* (salida) donde muestra el resultado de la operación ejecutada. El número *núm* se genera automáticamente y de manera secuencial a lo largo de una sesión de trabajo, desde que se inició el programa hasta que se sale de él. Puede servir como referencia de un cálculo previamente obtenido, dado que el programa almacena en el *Kernel* todas las evaluaciones y cálculos realizados durante una sesión de trabajo.

`1 + 4 (* suma de dos números *)`

5

- **Text.** Celdas cuyo contenido no es evaluable; permiten introducir texto sin necesidad de presentarlo como un comentario.
- **Otros tipos de celdas.** Puede darse a las celdas diferentes formatos. Esto permite organizar la presentación de la información contenida en un *notebook* y utilizar el programa como un editor de texto con la ventaja de poder intercalar celdas tipo *Input* que muestren cálculos. Las distintas opciones (*Title*, *Chapter*, *Subchapter*, *Section*, *Subsection*, ...) se encuentran en el menú *Formato*→*Estilo*. También, puede mostrarse de forma permanente en la barra de herramientas: *Ventana*→*Barra de herramientas*→*Formato*.

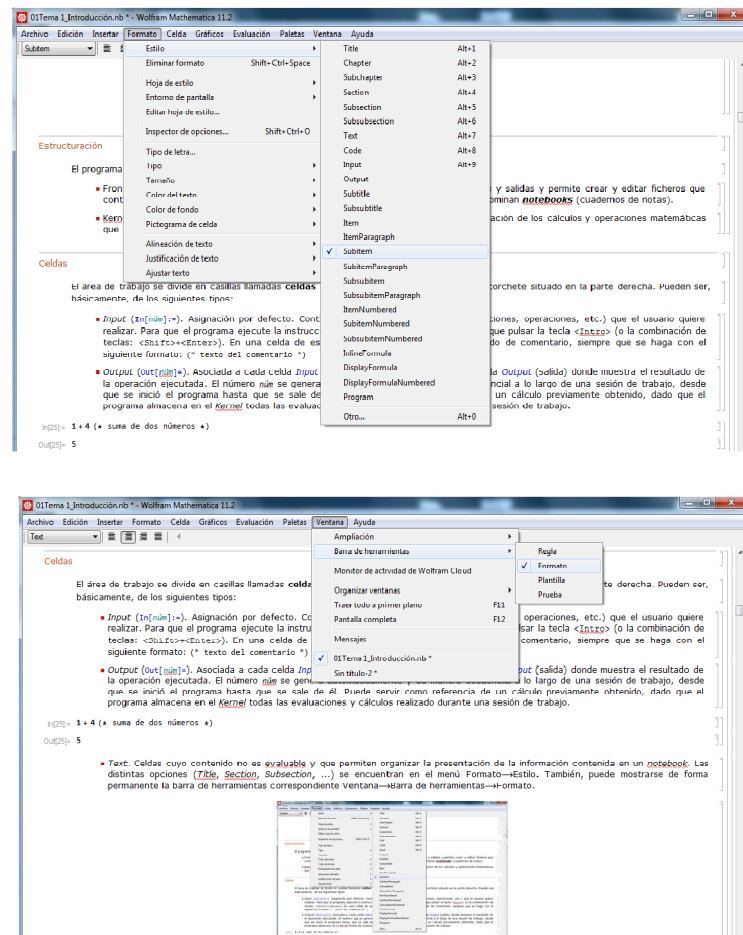


Figura 1. Cómo mostrar de manera permanente el formato de celdas

Sintaxis

Las diferentes funciones definidas en *Mathematica* tienen una sintaxis precisa. Puede consultarse en el menú *Ayuda*→*Documentación Wolfram*. Para que una función se ejecute de forma correcta en una celda de entrada debe estar escrita de manera que se ajuste estrictamente a las especificaciones de su sintaxis. En caso contrario, el programa devuelve un mensaje en el que se indica el error cometido.

Deben tenerse en cuenta las siguientes reglas básicas :

- **Mayúsculas/minúsculas.** *Mathematica* distingue unos caracteres de otros. Se recuerda que las funciones y palabras reservadas comienzan siempre por mayúscula.

$$\text{Simplify}\left[\frac{(2 + 2a + b + a^2)(a - 1)^2}{(a^3 - a^2 - a + 1)(b^2 + 3b + 2)}\right] \quad (* \text{ simplificación de una expresión simbólica } *)$$

$$\frac{1}{1 + b}$$

E = 1

Set: Symbol e is Protected.

1

E // N (* salida en formato numérico con seis cifras significativas *)

2.71828

- Espacios en blanco. Un espacio colocado entre dos variables o números en una celda *Input* se interpreta como el signo de multiplicar.

E E

e^2

N[E E] (* salida en formato numérico con seis cifras significativas *)

7.38906

- Paréntesis, corchetes y llaves. Los paréntesis, (), se utilizan para agrupar diferentes términos e indicar prioridad. Los corchetes, [], son de uso exclusivo de las funciones para delimitar sus argumentos. Las llaves, { }, se utilizan para definir listas de elementos: vectores, matrices, etc.

2 * (E + 3)

2 (3 + e)

2 * E + 3

3 + 2 e

{2 * (E + 3), 2 * E + 3}

{2 (3 + e), 3 + 2 e }

Cos [Pi]

-1

Una de las facilidades suministradas en las últimas versiones del programa consiste en la aparición de un menú desplegable que muestra las distintas *Funciones* del programa que comienzan con las letras que se van tecleando:

- permite seleccionar la opción deseada sin necesidad de terminar de escribir; a continuación, se muestran los desplegables que aparecen sucesivamente a medida que se teclea **Cont**:

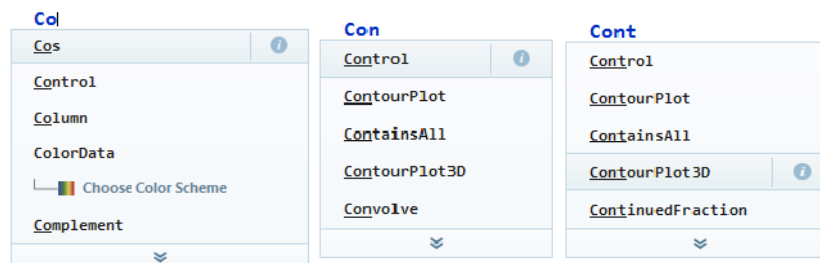


Figura 2. Menús desplegables de ayuda para la introducción de funciones

- una vez escogida la *Función* deseada, el programa permite la opción de desplegar otro menú donde se especifica su sintaxis; por ejemplo, si se elige **ContourPlot3D**:

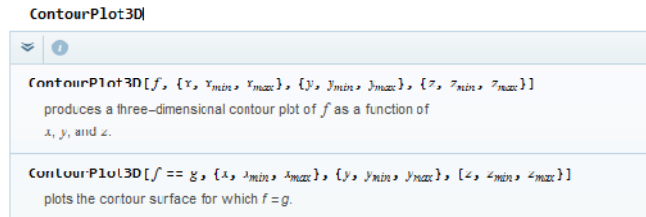


Figura 3. Sintaxis de la función seleccionada

- tras pinchar en una de las opciones (puede que una *Función* presente diferentes sintaxis según lo que se pretenda realizar) sólo queda introducir los datos de nuestro caso particular:

`ContourPlot3D[f, {x, xmin, xmax}, {y, ymin, ymax}, {z, zmin, zmax}]`

- pinchando sobre el símbolo  se abre la ayuda de la función correspondiente

Paletas

El programa *Mathematica* incorpora diferentes paletas en el menú desplegable *Paletas*. Junto con los diferentes formatos de las celdas puede realizarse una completa edición de textos técnicos.

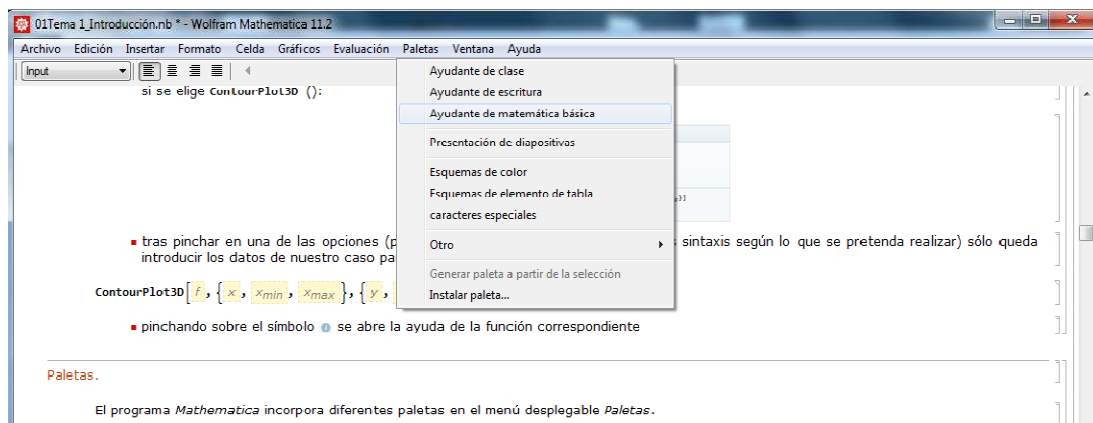


Figura 4. Forma de mostrar el Ayudante de matemática básica

La paleta *Ayudante de matemática básica* permite introducir notación simbólica estándar. En este curso resulta de gran utilidad ya que, por una parte, facilita la edición de informes de tipo técnico y, por otra, permite plantear operaciones sin utilizar la sintaxis original de forma que resultan más comprensibles, incluso para quien no domine el programa.

Ejemplo. Cálculo de la primera derivada de la función $f(x) = \frac{2 - \sqrt[4]{x^3}}{1 - x^2}$

- Sintaxis del programa

`D[(2 - Power[x^3, 1/4]) / (1 - x^2), x] // Simplify`

$$\frac{x \left(-3x - 5x^3 + 16(x^3)^{3/4} \right)}{4(x^3)^{3/4}(-1 + x^2)^2}$$

- Notación matemática estándar

$$\partial_x \left(\frac{2 - \sqrt[4]{x^3}}{1 - x^2} \right) // \text{Simplify}$$

$$\frac{x \left(-3x - 5x^3 + 16(x^3)^{3/4} \right)}{4(x^3)^{3/4}(-1 + x^2)^2}$$

Operadores aritméticos

Operadores

Mathematica puede utilizarse como una calculadora convencional aunque con una importante diferencia: la precisión en el cálculo. Las operaciones se realizan siempre en forma exacta o bien con el grado de precisión que se indique.

Se reconocen los operadores aritméticos habituales: suma (+), diferencia (-), producto (*), cociente (/) y potencia (^).

Jerarquía

En ciertas expresiones que involucren varios de los operadores pueden aparecer problemas de interpretación.

En estos casos, el programa evalúa la expresión de **izquierda a derecha** y aplica los criterios de **prioridad** habituales entre los operadores: la mayor corresponde a la potencia, el cociente y el producto se encuentran en el siguiente nivel de prioridad y, por último, la suma y la resta.

Los paréntesis se usan para priorizar operaciones con menor jerarquía.

Producto

El operador * puede sustituirse por un espacio en blanco. Cuando se utiliza entre números el programa lo sustituye de forma automática por el símbolo $\times$.

3×9

27

Debe prestarse especial cuidado al utilizar el espacio en blanco en cálculos simbólicos. No se recomienda porque es fuente de errores. Es preferible adquirir el hábito de teclear el asterisco para realizar productos.

`x = 2; (* asignación del valor 2 a una variable x *)`

El producto de las variable `x` e `y` puede plantearse de cualquiera de las dos formas siguientes:

`{x*y, xy}`

`{2 y, 2 y}`

Si se olvida el espacio blanco, el programa interpreta la secuencia de texto `xy` como una variable con ese nombre que no tiene asignado ningún valor.

`xy`

`xy`

Funciones del programa

Sintaxis

Mathematica incorpora una serie de funciones predefinidas. En este curso se utilizarán únicamente aquellas que permitan realizar las operaciones matemáticas necesarias para la resolución de los ejercicios planteados.

Una función se ejecuta correctamente en una celda de entrada siempre que no se hayan cometido errores sintácticos al teclear; en caso contrario, en la salida se devuelve el correspondiente mensaje de error.

Además de la forma indicada anteriormente, la sintaxis de cada función puede consultarse en el menú de *Ayuda* → *Documentación Wolfram* o seleccionando el nombre de la función y pulsando la tecla <F1> .

Deben seguirse las siguientes reglas al teclear las funciones:

- el nombre se escribe en minúsculas salvo la **primera letra** que siempre es **mayúscula**
- contienen uno o varios **argumentos** encerrados entre corchetes y separados por comas
- sintaxis: **Función**[*arg 1*, *arg 2*, ..., *arg n*]

Ejemplo. Algunos ejemplos de funciones matemáticas (pueden consultarse en el menú de *Ayuda*):

`Sqrt[4*(12+2)]` (* raíz cuadrada *)

$2\sqrt{14}$

`Log[E]` (* logaritmo natural *)

1

`Log[10, 1000]` (* logaritmo decimal *)

3

`Power[a, 3]` (* potencia cúbica *)

a^3

`Exp[2]` (* función exponencial *)

e^2

Función de una función

Pueden anidarse funciones y operaciones con éstas teniendo en cuenta que todas ellas deben tener sus argumentos entre corchetes.

Ejemplo. Evaluar la expresión: $\left| \frac{\sin(0.5) + \ln(e^2)}{\cos(1.2) - \sqrt{23}} \right|$

`Abs[(Sin[0.5] + Log[E^2]) / (Cos[1.2] - Sqrt[23])]`

0.559251

Funciones especiales

Salida previa: %

El símbolo % se utiliza para hacer referencia al resultado de la última celda ejecutada.

Ejemplo. Acceso a resultados de celdas Output.

`N[9*Sqrt[3/(Log[2]+E)]]`

8.43985

`{Cos[%], Cos[8.43985]}`

{-0.552927, -0.552923}

Ejecución simultánea: //

El símbolo // permite la ejecución simultánea en la misma celda de dos funciones de forma que la situada más a la izquierda es el argumento de la que está a su derecha.

Ejemplo. Ejecución simultánea de la operación planteada en el ejercicio anterior.

```
9 * Sqrt[3 / (Log[2] + E)] // N
```

```
8.43985
```

Se procede de igual manera con un mayor número de funciones.

Factorial: !

El símbolo ! está habilitado para calcular el factorial de un número $n \in \mathbb{N} \cup \{0\}$.

Ejemplo. Cálculo de factoriales.

```
{0!, 4!, 7!}
```

```
{1, 24, 5040}
```

Inhibición de la salida: ;

El símbolo ; escrito al final de la línea de comando inhibe la salida. Es decir, la celda Input se ejecuta pero no se muestra el resultado en la correspondiente celda Output. Puede resultar útil para reducir la longitud de un *notebook*.

Precisión numérica

Introducción

Mathematica en todas las operaciones realizadas con números enteros o racionales devuelve el valor exacto.

```
2^512
```

```
13 407 807 929 942 597 099 574 024 998 205 846 127 479 365 820 592 393 377 723 561 443 721 764 030 073 546 976 801 874 298 166 903 427 \
690 031 858 186 486 050 853 753 882 811 946 569 946 433 649 006 084 096
```

Si en una operación interviene un número decimal *Mathematica* devuelve siempre un valor aproximado con una precisión de 6 dígitos significativos (aunque internamente trabaja con una precisión mayor).

```
2.0^512
```

```
1.34078 × 10154
```

Cociente y funciones trascendentes

En el caso de los cocientes, para evitar la falta de precisión, *Mathematica* devuelve una fracción simplificada.

```
51 / 34
```

```
3
—
2
```

El programa trabaja igual con las funciones trascendentes cuyos resultados no pueden expresarse con un número finito de decimales.

`{Cos[3], Sqrt[2] / 3}`

$\left\{\cos[3], \frac{\sqrt{2}}{3}\right\}$

Si en las operaciones aparecen números decimales, como ya se ha indicado, se opera normalmente.

`51.0 / 34`

`1.5`

`{Cos[3.0], Sqrt[2] / 3.0}`

`{-0.989992, 0.471405}`

La función N[]

Función habilitada para la resolución numérica de los diferentes cálculos que se planteen en el curso.

Sintaxis. Pueden introducirse uno o dos argumentos dependiendo de la precisión requerida. Por defecto, presenta una precisión de 6 cifras significativas aunque, internamente, trabaje con mayor precisión. El número deseado de cifras significativas puede indicarse con un segundo argumento, n .

Ejemplo. Evaluar con 5, 6 y 15 cifras significativas la expresión: $\cos(3 \text{ rad})$

`{N[Cos[3], 5], N[Cos[3], 6], N[Cos[3], 15]}` (* ejecución de tres operaciones en la misma línea de comando *)

`{-0.98999, -0.989992, -0.989992496600445}`

Ejemplo. Evaluar con 6 y 4 cifras significativas la expresión: $\sqrt{2}$

`{N[Sqrt[2], 6], N[Sqrt[2], 4]}`

`{1.41421, 1.414}`

Constantes matemáticas

El programa *Mathematica* tiene predefinidas una serie de constantes habituales en Matemáticas, como son:

- el número π : **Pi**
- número e : **E**
- la unidad imaginaria i : **I**
- factor de conversión de radianes a grados: **Degree**

Para hacer referencia al infinito se usa **Infinity**.

Ejemplo. Constantes matemáticas.

`{I, E, Pi, Cos[Pi / 4]}`

$\left\{i, e, \pi, \frac{1}{\sqrt{2}}\right\}$

`{I, E, Pi, Cos[45 Degree]} // N`

`{0. + 1. i, 2.71828, 3.14159, 0.707107}`

Variables

Definición

Aunque ya se ha definido con anterioridad una variable y se le ha asignado un valor ($x=2$), básicamente, sólo se han utilizado valores numéricos. El uso de variables permite almacenar información de forma temporal y recuperar esos datos para cálculos posteriores; por lo tanto, son de gran utilidad en problemas y operaciones de cierta complejidad o que puedan descomponerse en diferentes pasos. Al ser usadas por primera vez el programa las identifica y almacena por lo que no es necesario declararlas previamente.

Para introducir una expresión simbólica y asignarle posteriormente un valor basta con elegir un nombre con el cual representarla siempre teniendo en cuenta las siguientes consideraciones y recomendaciones:

- No pueden utilizarse como nombres de variable los reservados para designar a las instrucciones, funciones y constantes de *Mathematica*. Se recomienda no usar mayúsculas, al menos en la primera letra, para diferenciarlas de las funciones y comandos reservados por el programa.
- El nombre de la variable tiene que empezar con una letra (carácter no numérico). Es válido cualquier nombre siempre que no contenga caracteres especiales. No hay restricción para la longitud del nombre y debe tenerse en cuenta que se distingue entre mayúsculas y minúsculas.

Asignación de valores

Una variable representada por un nombre tiene un valor simbólico hasta que no se le asigne una expresión o un valor determinado. El nombre de una variable aparece en color azul hasta que se realiza la asignación, en este momento pasa a figurar en color negro.

La asignación puede realizarse de dos formas:

- Inmediata. Sintaxis: **variable=valor(o expresión)**.

Ejemplo. Asignación de valor a una variable, x , y utilización posterior para asignar el valor de otra variable, y .

$x = 2$

2

$y = 2x + 1;$

{y, x = 3, y}

{5, 3, 5}

- Diferida. Sintaxis: **variable:=valor(o expresión)**. No produce salida. La asignación no es inmediata sino que se produce cuando se utiliza la variable de forma que cambios posteriores a este tipo de asignación sí afectan al valor de la variable.

Ejemplo. Asignación de valor a una variable, x , y utilización posterior para asignar de forma diferida el valor de otra variable, y .

$x = 2$

2

$y := 2x + 1$

{y, x = 3, y}

{5, 3, 7}

Borrado, la función `Clear[]`

Esta función permite borrar los valores asignados de las variables especificadas como argumentos. No devuelve salida.

Sintaxis. Pueden introducirse uno o varios argumentos que son los nombres de las variables cuyos valores quieren borrarse.

Una variable mantiene el valor que se le ha asignado mientras no se realice una nueva asignación o se borre su contenido (en este caso, aparece de nuevo en color azul). El borrado de contenido de las variables es aconsejable para evitar posibles conflictos y resultados inesperados en operaciones posteriores.

Puede borrarse el contenido de todas las variables definidas hasta el momento en una sesión de trabajo con el argumento:

`Clear["Global`*"]`

Otra posibilidad para realizar este borrado global consiste en desplegar el menú *Evaluación* → *Salir del kernel* → *Local*.

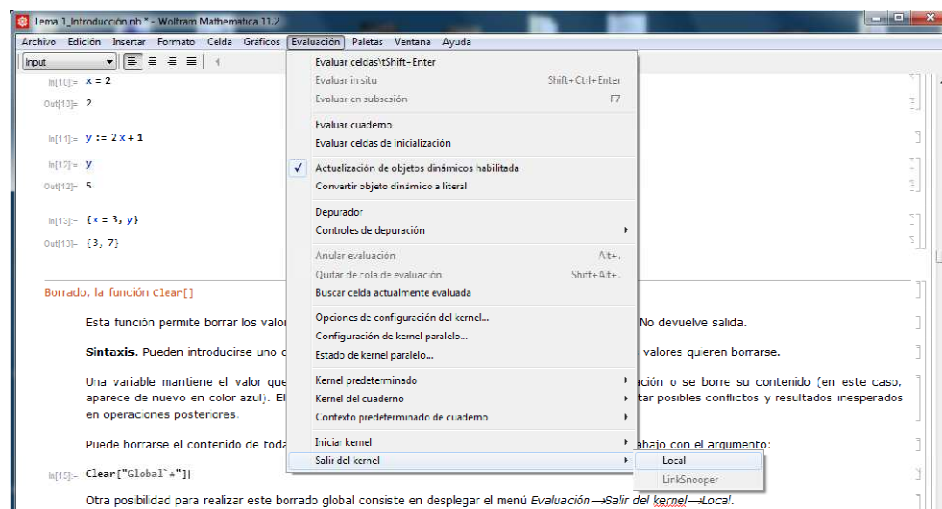


Figura 5. Borrado global de variables y funciones definidas por el usuario

Cálculo simbólico

Introducción

Mathematica permite realizar cálculos simbólicos, es decir, aquellos que involucran variables genéricas que no tienen asignado un valor numérico determinado.

Ejemplo. Asignación de valor a una variable, **f**, de la expresión simbólica: $b(z - 1)^2$.

`f = b * (z - 1) ^ 2`

`b (-1 + z) ^ 2`

Debe tenerse especial cuidado con la jerarquía de los operadores aritméticos, la sintaxis de las funciones matemáticas y el uso correcto de corchetes y paréntesis.

Operadores de reemplazamiento: sustitución (`/.`) y regla (`->`)

Estos operadores permiten realizar sustituciones en las expresiones indicadas. No se trata de una asignación de valores y, por tanto, el reemplazamiento no modifica el valor de ninguna variable.

Se hace notar que, siempre, una asignación de valor a una variable prevalece sobre el operador regla.

El operador $\rightarrow$ ($\Rightarrow$) reemplaza la expresión situada a su izquierda por la que figura a su derecha.

El operador $/.$ aplica una regla o conjunto de reglas situadas a su derecha de forma que se realizan una serie de sustituciones en la expresión que le precede.

Ejemplo. Sustituciones en la variable, f , del ejemplo anterior.

f

$$b (-1 + z)^2$$

$$f /. b \rightarrow 2$$

$$2 (-1 + z)^2$$

$$f /. \{b \rightarrow 2, z \rightarrow -3\}$$

$$32$$

f

$$b (-1 + z)^2$$

Simplificación, la función `Simplify[]`

Esta función realiza una serie de transformaciones algebraicas en una expresión con el fin de obtener la forma más sencilla posible.

Sintaxis. El argumento de la función es la expresión que quiere simplificarse. Pueden introducirse más argumentos como condiciones que deben asumirse durante la simplificación.

Ejemplo. Simplificación de una expresión simbólica.

$$\text{Simplify}\left[\frac{(2 + 2 a + b + a b) (a - 1)^2}{(a^3 - a^2 - a + 1) (b^2 + 3 b + 2)}\right]$$

$$\frac{1}{1 + b}$$

Expansión, la función `Expand[]`

Si es posible, transforma una expresión en otra polinomial en la que sólo aparecen sumas, diferencias, productos, cocientes y potencias positivas.

Sintaxis. El argumento es la expresión que quiere desarrollarse. Como segundo argumento puede indicarse un patrón que hace que no se desarrollen las partes de la expresión en las que no aparece.

Ejemplo. Desarrollo de una expresión simbólica.

$$\text{Expand}[(a + 1)^2 (x - 1) * (1 - x^2)]$$

$$-1 - 2 a - a^2 + x + 2 a x + a^2 x + x^2 + 2 a x^2 + a^2 x^2 - x^3 - 2 a x^3 - a^2 x^3$$

$$\text{Expand}[(a + 1)^2 (x - 1) * (1 - x^2), x]$$

$$-(1 + a)^2 + (1 + a)^2 x + (1 + a)^2 x^2 - (1 + a)^2 x^3$$

Factorización

Si es posible, factoriza la expresión polinomial dada.

Sintaxis. La más básica incluye un solo argumento que es la expresión que quiere factorizarse.

Ejemplo. Factorización de una expresión simbólica.

Factor $[64 - 80x - 20x^2 + 25x^3 + 10x^4 + x^5]$

$(-1 + x)^2 (4 + x)^3$

Comparación

Mathematica dispone de símbolos predefinidos que permiten realizar comparaciones entre las expresiones que se indican. La celda de salida contiene False o True en función del resultado de la comparación.

- Igualdad. **Unequal** (**!=** , **≠**) y **Equal** (**==**).
- Desigualdad. **LessEqual** (**<=**), **Less** (**<**), **GreaterEqual** (**>=**), **Greater** (**>**).

Ejemplo. Comparación de expresiones.

$\{x = 1, y = 0, z = -1\};$

$\{(x - y)^2 == z^2, x != y\}$

$\{True, True\}$

$\{(x - y)^2 <= z^2, (x - y)^2 > z^2\}$

$\{True, False\}$

Funciones definidas por el usuario

Introducción

Un usuario puede definir sus propias funciones matemáticas de una o varias variables de forma que la expresión analítica correspondiente se asigna a un nombre de función. Es habitual la asignación diferida que tiene efecto cuando se hace referencia a la función.

Sintaxis

La sintaxis correcta es:

nombrefuncion $[variable\ 1_, variable\ 2_, \dots, variable\ n_] := expresión$

El símbolo **_** (*underscore*) a continuación de cada variable en la parte izquierda de la asignación es muy importante, indica al programa las variables independientes de la función. Ese símbolo no se pone en la parte derecha de la asignación.

Para denotar una función (**nombrefuncion**) se recomienda evitar nombres que comiencen con letra mayúscula para prevenir confusiones con las funciones predefinidas del programa. Además, hay que asegurarse que el nombre no haya sido usado anteriormente en la sesión de trabajo o borrar su contenido preventivamente.

Ejemplo. Definición de tres funciones polinómicas de una variable:

$$p(x) = x^2 + 2x - 1, q(x) = -x^2, s(x) = p(x) + q(x)$$

Clear $[x, p, q, s]$

$\{p[x_] := x^2 + 2x - 1, q[x_] := -x^2, s[x_] := p[x] + q[x]\};$

Referencias posteriores

Una vez definida la función ya se puede utilizar para cualquier cálculo sin más que hacer referencia a ella con el nombre asignado.

Ejemplo. Referencias a funciones definidas previamente (se usan las del ejemplo anterior)

- evaluación para valores determinados de las variables

```
{p[0], q[0], s[0]}
```

```
{-1, 0, -1}
```

- obtención de listas de valores (la función **Table** se verá más adelante en este curso)

```
Table[s[i], {i, 10}]
```

```
{1, 3, 5, 7, 9, 11, 13, 15, 17, 19}
```

- cálculos numéricos

```
Sqrt[p[2] - q[0]]
```

$$\sqrt{7}$$

- derivación (**D** deriva la función planteada como primer argumento respecto a la variable indicada como segundo argumento)

```
D[p[x], x]
```

```
2 + 2 x
```

```
q'[0]
```

```
0
```

Ejemplo. Definición de una función de dos variables: $f(x, y) = x^2 + y^2 - 1$

```
Clear[f, x, y]
```

```
f[x_, y_] := x^2 + y^2 - 1
```

```
f[0, 0]
```

```
-1
```

```
{D[f[x, y], x], D[f[x, y], y]}
```

```
{2 x, 2 y}
```

Representación gráfica

Introducción

Se presentan una serie de funciones del programa *Mathematica* para la representación gráfica de funciones de varias variables.

No se considera ámbito de este curso la profundización en este área; por ello, el estudio de las diferentes opciones de dibujo que ofrecen se deja en manos del alumno en el menú de *Ayuda* (*Ayuda* → *Documentación Wolfram*).

Representación gráfica en 2D

Representación gráfica de funciones de una variable en el plano.

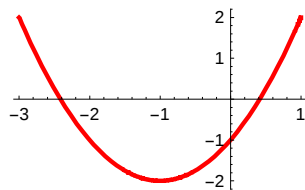
- **Plot**. Traza la gráfica de las funciones indicadas como argumento en el intervalo señalado.
- **ContourPlot**. Representación gráfica de una función en forma implícita o de los puntos que verifican una ecuación.
- **ParametricPlot**. Representación gráfica en forma paramétrica.
- **Show**. Combina en una sola gráfica diferentes representaciones que se han asignado previamente a variables.

Ejemplo. Representaciones gráficas en el plano.

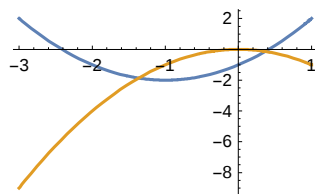
`Clear[x]`

`{p[x_] := x^2 + 2 x - 1, q[x_] := -x^2};`

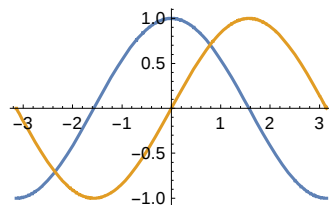
`g1 = Plot[p[x], {x, -3, 1}, PlotStyle -> {Red, Thickness[0.015]}]`



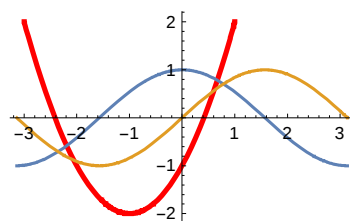
`g2 = Plot[{p[x], q[x]}, {x, -3, 1}]`



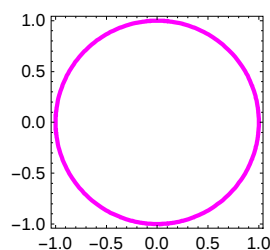
`g3 = Plot[{Cos[x], Sin[x]}, {x, -Pi, Pi}]`



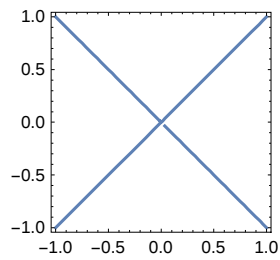
`Show[g1, g3, PlotRange -> All]`



`ContourPlot[x^2 + y^2 == 1, {x, -1, 1}, {y, -1, 1}, ContourStyle -> {Magenta, Thickness[0.02]}]`



`ContourPlot[x^2 == y^2, {x, -1, 1}, {y, -1, 1}]`



Representación gráfica en 3D

Representación gráfica de funciones de dos variables en el espacio.

- **Plot3D**. Traza la gráfica de las funciones indicadas como argumento en el intervalo señalado.
- **ContourPlot3D**. Análogo al explicado para funciones de una variable.
- **ParametricPlot3D**. Representación gráfica en forma paramétrica.

Ejemplo. Intersección de superficies junto con la proyección de la línea de intersección.

`Clear[x, y]`

`esf2 = ContourPlot3D[x^2 + y^2 + z^2 == 4, {x, -2, 2}, {y, -2, 2}, {z, 0, 2}, ContourStyle -> {Green, Opacity[0.5]},
Mesh -> False, AxesLabel -> {X, Y, Z}, Ticks -> {{-2, 2}, {-2, 2}, {-2, 0, 2}}, BoxRatios -> Automatic];`

`plan3 = ContourPlot3D[3 z + sqrt(3) y == 2 sqrt(3), {x, -2, 2}, {y, -2, 2}, {z, 0, 2},
ContourStyle -> {Yellow, Opacity[0.5]}, Mesh -> False, AxesLabel -> {X, Y, Z}, Ticks -> {{-1, 0, 1}, {-1, 1}, {0, 1}}];`

`(* proyección de la línea de intersección sobre z=0 *)`

`Eliminate[{x^2 + y^2 + z^2 == 4, 3 z + sqrt(3) y == 2 sqrt(3)}, z] (* elimina la variable z *)`

`8 + 4 y - 4 y^2 == 3 x^2`

`(* parametrización de la línea de intersección *)`

`{x[t_] = sqrt(3) Cos[t], y[t_] = 1/2 + 3/2 Sin[t], z[t_] = (2 sqrt(3) - sqrt(3) y[t])/3};`

`lin3 = ParametricPlot3D[{x[t], y[t], z[t]}, {t, 0, 2 pi}, PlotStyle -> {Magenta, Thickness[0.02]}];`

`lin3p = ParametricPlot3D[{x[t], y[t], 0}, {t, 0, 2 pi}, PlotStyle -> {Red, Thickness[0.02]}];`

`Show[esf2, plan3, lin3, lin3p, PlotRange -> All]`

