Práctica 1

Introducción al programa Mathematica (I)

1.1. Introducción

Mathematica es un programa que realiza cálculos de manera tanto numérica como simbólica, asimismo, permite realizar gráficos e integrar los resultados en un mismo documento. Además, dispone de un lenguaje de programación propio que permite la construcción de programas que se ajustan a las necesidades del usuario.

Desde sus inicios, en 1988, su uso se ha ido extendiendo a los nuevos campos de la Ciencia y la Ingeniería. En la página oficial de Mathematica

`http://www.wolfram.com`

se puede encontrar mucha información y utilidades para este programa.

En la instalación de que se dispone en la Universidad, se puede cargar el programa abriendo las carpetas **upvnet**, **aplicaciones científicas** y **Mathematica**. A continuación seleccionamos la versión 5.0 y aparece una pantalla como la que se muestra en la figura 1.1:

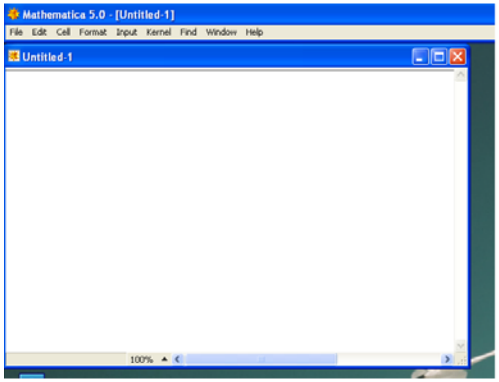


Figura 1.1.- Pantalla del Mathematica 5.0.

donde podemos observar que tenemos un Notebook (ó cuaderno de notas) en blanco y una barra de menús.

Los Notebooks se componen de celdas donde se agrupa la información y se diferencian fácilmente unas de las otras porque vienen delimitadas por corchetes en la parte derecha. Existen distintos tipo de celdas que responden a diferentes cometidos y se distinguen (unas de otras) por la forma de los corchetes que las abarca.

Por ejemplo, las delimitadas por corchetes con una doble rayita en la parte superior son de texto y las que tienen un pequeño triangulito son de entradas y resultados de cálculo. Por defecto, todas las celdas son de cálculo (Input) y si queremos cambiar el estilo, por ejemplo, que éstas sean de texto, debemos de seleccionar en la barra de herramientas: **Format-Style-Text** o bien presionar **Alt 7** . De igual forma se procede para otros estilos.

Ejemplo 1.1 *Para escribir en una celda de texto: "Esta es la primera práctica", abrimos el menú* **Format** *y seleccionamos el submenú* **style** *que nos permite decidir el estilo, en este caso, de texto. (También se puede acceder al submenú* **style** *colocando el cursor del ratón sobre el corchete que se encuentra a la derecha de la celda y hacemos click con el botón izquierdo. A continuación, presionando el botón derecho del ratón sale un menú que nos permite editar dicha celda.*

Para evaluar una operación se utilizan las teclas MAYÚSCULA (SHIFT,

$\Uparrow$) e INTRO (ENTER, $\longleftrightarrow$) simultáneamente, o bien teclear INTRO del teclado numérico.

Las evaluaciones las ejecuta el Kernel, o núcleo del programa. Éste se carga en la memoria del ordenador al realizar la primera operación, por lo que tardará unos segundos en aparecer el resultado.

Ejemplo 1.2 *Escribir y evaluar la operación 3+4 . Al realizar esta operación siguiendo los pasos anteriores se obtiene el resultado mostrado en la figura 1.2.*

Figura 1.2.- *Resultado de hacer 3+4.*

Fijaos que tanto las entradas (`In[]`) como las salidas o resultados (`Out[]`) se van numerando según el orden de ejecución. El Kernel ó núcleo guarda todas las asignaciones y evaluaciones hechas desde que se comenzó la sesión hasta que se sale de ésta.

Para añadir o crear una celda nueva dentro de un Notebook situamos el cursor del ratón al final del Notebook o bien entre dos celdas que ya existan, hasta que el cursor aparezca de forma horizontal. Pulsando con el botón izquierdo aparece una barra horizontal a lo largo de toda la ventana. Si lo que queremos es borrar una celda, una vez seleccionada se le da a la tecla **Supr**.

Además del manejo de estas celdas, es conveniente que realices un breve paseo por otros menús interesantes que se encuentran en la barra de herramientas, por ejemplo, **File-New**: Abre un archivo ó Notebook en blanco.

File-Open: Carga un Notebook concreto. **File-Print**: Imprime un Notebook.

Hay otros menús de gran utilidad que aparecen en la parte superior de la pantalla: **Edit**: Donde aparecen los comandos habituales que nos permiten acciones como Cortar (**Cut**), Copiar (**Copy**) ó Pegar (**Paste**). Aquí también podemos encontrar las 'Palettes', útiles para escribir expresiones matemáticas y realizar cálculos simbólicos. Es conveniente que abras estos 'Palettes' y observes las posibilidades que nos ofrecen para escribir y realizar operaciones matemáticas.

Format: permite cambiar el tamaño y estilo de texto. **Kernel**: Para interrumpir algún cálculo o anular alguna evaluación (por error o porque el programa tarde demasiado en dar la respuesta) se pueden usar las opciones Interrumpir (**Interrupt**) ó Abortar (**Abort**) de este menú, etc.

1.2. Sintaxis general

Para evitar posibles errores es conveniente escribir de forma adecuada las expresiones (respetando siempre el formato y la sintaxis de cada sentencia).

Las normas siguientes afectan a todas las órdenes y expresiones de Mathematica:

1. El programa distingue los caracteres escritos en mayúscula de los escritos en minúscula. Todas las funciones, y constantes que tiene implementadas el programa comienzan con la primera letra en mayúscula.

2. Es conveniente usar sólo letras minúsculas para definir nuestras propias variables y funciones (así se podrán diferenciar con facilidad de las anteriores).

3. Mathematica interpreta un espacio en blanco como la operación de multiplicar. Por esto, no se deben dejar espacios en blanco dentro del nombre de una función o variable.

4. Para el producto de dos variables se puede usar $*$ ó espacio en blanco. Sin embargo, si escribimos `xy` el Mathematica lo interpretará como el nombre de una nueva variable con dos letras. Existe una excepción a esta regla, cuando multiplicamos un número por una variable. Por ejemplo, $3{*}x$ se puede escribir como $3x$. Sin embargo, no obtendremos la misma interpretación al escribir $x3$. Por esta razón, no podremos dar un

nombre a una variable ó constante que empiece por un número, siempre debe de empezar con una letra, aunque después contenga números. Los nombres a, aa, a1, a11, etc son válidos para variables; no lo son 1a, 2a, 11a, 1a1, etc.

Los corchetes, [ ], delimitan los argumentos de todas las funciones y órdenes en Mathematica, y también aquellas que defina el usuario. Cuando hay dos o más argumentos se separan por comas.

Las llaves, { }, se usan para la definición de listas de elementos, que pueden ser de cualquier tipo. También se pueden interpretar como vectores o matrices (que serán listas de listas). Estas estructuras se verán con detalle más adelante. También se usan para incluir ciertas opciones dentro de algunas órdenes de Mathematica.

Los paréntesis, ( ), se utilizan para indicar la proridad de las operaciones de tipo aritmético. Se pueden usar en conjunción con asteriscos para incluir comentarios, que no se evalúan. Éstos pueden estar en una línea o a cualquier lado de un cálculo y permitirán dar información adicional en cualquier punto de un programa, como en el ejemplo siguiente[1]:

```
(* esto es un comentario *)
In[]:= 1+2*3
Out[]= 7
```

Si no se desea que aparezca el resultado de una orden se pone ; al final de la sentencia (la expresión se evalúa pero no se muestra el resultado en la pantalla).

1.3. Cálculos numéricos. Funciones y constantes básicas

Mathematica se puede utilizar como una calculadora, donde las operaciones básicas se introducen con los siguientes símbolos del teclado:

[1]En general en las explicaciones usaremos la notación `In[]:=` para indicar las intrucciones que debemos introducir, y `Out[]=` para indicar el resultado que nos devuelve el Mathematica. Así, en el ejemplo que sigue nosotros teclearemos `1 + 2*3` (SIN EL `In[]`) y el programa nos devolverá el resultado.

Suma	+
Diferencia	−
Producto	∗ o espacio
Cociente	/
Potencia	ˆ

En este programa las operaciones aritméticas se realizan siguiendo un determinado orden, por ello, si se quiere alterar el orden en el que se realizan las operaciones habrá que hacer uso de los paréntesis (NO CORCHETES) para que el programa realice justo la operación deseada.

Ejemplo 1.3 *Podemos comprobar que el resultado obtenido no es el mismo si se hace uso o no de los paréntesis.*

Por ejemplo, tecleamos las expresiones:

```
In[]:=2^4 + 3
In[]:=2^(4+3)
```

que corresponden a las siguientes operaciones: 2^4+3 *y* $2^{(4+3)}$*, respectivamente. Al pulsar Intro obtenemos los resultados:*

```
Out[]=19
Out[]=128
```

Ejemplo 1.4 *Calcular*

$$\frac{2^5}{15-\sqrt[3]{27}} - \frac{1}{7(2-3^5)} \ .$$

Para ello basta teclear:

```
In[]:=2^5/(15-27^(1/3))-1/(7*(2-3^5))
```

Al pulsar mayúsculas *y* Enter*, y obtendremos el resultado*

```
Out[]= 13499/5061
```

1.3.1. Utilización de resultados previos

Los resultados que vamos obteniendo se van numerando como `Out[1]` `Out[2]`, etc. Si queremos reutilizarlos en posteriores operaciones podemos hacer uso de lo siguiente:

- El símbolo %, que se utiliza para llamar al último resultado obtenido.
- El símbolo %% Llama al penúltimo resultado obtenido.
- El símbolo $\%n$ Llama al resultado obtenido en la salida `Out[n]`.

Por ejemplo, si queremos sumarle al primer resultado de la sesión (que valía 4, por ejemplo,) tres unidades, escribiremos

$$\%1 + 3$$

obteniendo como resultado, `Out[]=7`.

1.3.2. Resultados exactos y aproximados

A diferencia de otros programas, si se trabaja con números enteros Mathematica proporciona el resultado exacto independientemente del número de dígitos necesarios. Por ejemplo,

```
In[]:=7^12
Out[]= 13841287201
```

Si intervienen números racionales realiza los cálculos de forma exacta, esto es,

```
In[]:=  2/3+3/4
Out[]=  17/12
```

Si escribimos un número con punto decimal, Mathematica lo toma como un número aproximado y expresa el resultado con un número fijado de cifras decimales, por ejemplo,

```
In[]:=232./96
Out[]=2.41667
```


Podemos pedirle que nos dé una aproximación numérica con cierta precisión, poniendo al final del input //N o usando la función `N[]`.

Ejemplo 1.5 *Usar la función `N[]` para obtener una aproximación de* $\frac{17}{12}$.

```
In[]:= N[17/12]
Out[]=1.41667
```

Aún más, podemos obtener una aproximación decimal, indicando el número de dígitos significativos que deseamos conocer, para ello introducimos

```
In[]:= N[9864101/3628800,17]
Out[]=2.7182818011463845
```

1.3.3. Funciones y constantes básicas

Recuerda que todos los comandos, funciones, variables, opciones y constantes del Mathematica empiezan con letra mayúscula. Los argumentos de las funciones van siempre entre corchetes, como por ejemplo, `Integrate[ ]`, `Plot[ ]`. Ejemplos de algunos de los comandos o funciones propios del Mathematica son los siguientes:

`Sqrt[x]`	Raíz cuadrada de `x`
`Exp[x]`	Exponencial de `x`
`Sin[x]`	Seno de `x`
`Cos[x]`	Coseno de `x`
`Tan[x]`	Tangente de `x`
`Sinh[x]`	Seno hiperbólico de `x`
`Cosh[x]`	Coseno hiperbólico de `x`
`Tanh[x]`	Tangente hiperbólica de `x`
`Abs[x]`	Valor absoluto de `x`
`Log[x]`	Logaritmo neperiano de `x`
`Log[b,x]`	Logaritmo en base `b` de `x`
`N[expr]`	Valor aproximado de `expr`
`N[expr,n]`	Valor aproximado de `expr` con `n` dígitos significativos.
`Factorial[n]` o `n!`	Factorial de `n`
`Binomial[n,m]`	Número combinatorio $\binom{m}{n}$

Por otra parte, el Mathematica también tiene implantado el valor de las constantes matemáticas más frecuentes, como son:

`Pi`	Número π
`E`	Número e
`Infinity`	∞
`I`	La unidad imaginaria

1.4. Ayuda

Ante una duda, es conveniente consultar el manual de órdenes de Mathematica o algún libro con explicaciones y ejemplos.

La versión que usamos de Mathematica, dispone de una ayuda en la que puedes consultar todo lo referente al programa Mathematica. Para acceder a esta ayuda seleccionaremos con el ratón el menú **Help** y al abrise éste se buscará la opción **Find In Help**. En este meú se buscará la opción **Browser**, entonces se abre una ventana de ayuda, donde puedes localizar, entre otras cosas:

Built-in Functions : permite localizar y encontrar ejemplos de cualquier función predefinida en Mathematica.

Add-ons : proporciona un listado de los distintos paquetes de Mathematica.

Mathematica Book : que contiene el libro: Stephen Wolfram. *The Mathematica Book, Wolfram Media*, Cambridge University Press.

Para obtener ayuda sobre las distintas funciones implementadas en el programa, también podemos ayudarnos de los operadores ? y ??. Así, si introducimos,

`?Nombre`, obtendremos la información sobre la estructura que tiene el comando `Nombre`.

`??Nombre`, obtendremos la información completa sobre este comando.

`?Nomb*`, obtendremos un listado de todas las funciones que empiezan por `Nomb`.

Ejemplo 1.6 *Averiguar cómo se utiliza el comando* `Sum[]` *y utilizarlo para calcular*

$$1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \cdots + \frac{1}{10!}.$$

Buscando en la ayuda que aparece en pantalla, o bien introduciendo,


```
In[]:=?Sum
```

nos aparece en pantalla un mensaje similar al siguiente:

> $\texttt{Sum}[f, \{i, imax\}]$ *evaluates the sum of the expressions f as evaluated for each i from 1 to imax.* $\texttt{Sum}[f, \{i, imin, imax\}]$ *stars with* $i = imin$. $\texttt{Sum}[f,\ \{i, imin, imax, di\}]$ *uses steps di.* $\texttt{Sum}[f,\ \{i, imin, imax\}, \{j, jmin, jmax\}, \ldots]$ *evaluates a sum over multiple indices.*

Para realizar la suma indicada anteriormente, introducimos

```
In[]:=Sum[1/Factorial[i-1],{i,1,11}]
```

y obtenemos el resultado

$$\texttt{Out}[] = \frac{9864101}{3628800}$$

1.5. Operaciones con números complejos

El programa Mathematica permite realizar operaciones con números complejos. Basta tener en cuenta que la *unidad imaginaria*, i se escribe en mayúsculas: `I`.

Ejemplo 1.7 *Para expresar en forma binómica el número complejo*

$$2 - i - \frac{2i}{1+4i},$$

basta introducir

```
In[]:=(2-I)-(2 I)/(1+4 I)
```

obteniendo al pulsar Intro

$$\texttt{Out}[] = \frac{26}{17} - \frac{19}{17}\,I$$

El programa tiene implementadas funciones que realizan operaciones básicas con los números complejos:

`Re[ ]`	Calcula la parte real de un número complejo
`Im[ ]`	Calcula la parte imaginaria de un número complejo
`Abs[ ]`	Calcula el módulo de un número complejo
`Arg[ ]`	Calcula el argumento de un número complejo comprendido en el intervalo $[-\pi, \pi]$
`Conjugate[ ]`	Calcula el número complejo conjugado de uno dado

Ejemplo 1.8 *Calcular el argumento y módulo de* $z = \dfrac{i}{-2-2i}$.

Directamente, introducimos:

```
In[]:= Arg[I/(-2-2I)]
In[]:= Abs[I/(-2-2I)]
```

Obteniendo:

```
Out[]= -3Pi/4
Out[]=1/ 2 Sqrt[2]
```

Además, si deseamos, en este caso, comprobar las relaciones

$$Arg\left[\frac{z_1}{z_2}\right] = Arg[z_1] - Arg[z_2]$$

$$\left|\frac{z_1}{z_2}\right| = \frac{|z_1|}{|z_2|}$$

introducimos

```
In[]:= Arg[I]-Arg[-2-2 I]
In[]:= Abs[I]/Abs[-2-2 I]
```

y obtenemos los mismos resultados que antes.

Para calcular el complejo conjugado del $2 + 3i$ tenemos

```
In[]:=Conjugate[2+3I]
Out[]=2-3i
```


1.6. Matrices: Estructura de lista

En el programa Mathematica, para trabajar y manipular una colección de objetos se utiliza una estructura llamada “lista”. En una lista se presentan los elementos separados por comas y entre llaves, de la siguiente manera

```
{elemento1,elemento2,...,elementon}
```

pudiendo ser, a su vez, cada “elemento” otra lista. Así, el programa Mathematica considera a los vectores y matrices como listas.

Si, por ejemplo, tenemos una matriz, que llamaremos `matrizA` (se le puede llamar con cualquier otro nombre, `a`, `A`, etc.),

$$\texttt{matrizA} = \begin{bmatrix} 1 & 0 & 2 & 3 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

en Mathematica, la escribiremos como una lista donde cada elemento es una de las filas de la matriz que, a su vez, será otra lista. De este modo, se introduce

```
In[]:=matrizA={{1,0,2,3},{1,1,1,1},{0,1,0,1}}
```

Si queremos ver la matriz en su forma habitual, bastará con utilizar el comando `MatrixForm[ ]`. Escribiendo

```
In[]:=MatrixForm[matrizA]
```

se obtiene

$$\begin{matrix} 1 & 0 & 2 & 3 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{matrix}$$

Si lo que queremos es escribir el vector

$$V = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix},$$

escribiremos

```
In[]:=V= {{1},{2},{3}}
```

que corresponde a un vector columna que visualizaremos escribiendo `MatrixForm[V]`, obteniendo:

$$\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$$

Existen funciones como `Table[]`, que nos permiten construir matrices si los elementos de una matriz están dados por una expresión. Por ejemplo, si consideramos la matriz

$$\mathrm{B} = \begin{bmatrix} 2 & 3 & 4 \\ 3 & 4 & 5 \\ 4 & 5 & 6 \\ 5 & 6 & 7 \\ 6 & 7 & 8 \end{bmatrix},$$

que se obtiene al evaluar $i+j$, variando $i = 1, 2, 3, 4, 5$ y $j = 1, 2, 3$. Para generar esta matriz mediante el Mathematica utilizaremos el comando `Table[ ]`, que tiene la estructura

```
Table[expr.,{i,min,max,paso},{j,min,max,paso}].
```

En este ejemplo, escribiremos

```
In[]:=B=Table[i+j,{i,1,5,1},{j,1,3,1}]
```

o bien

```
In[]:=B=Table[i+j,{i,1,5},{j,1,3}]
```

donde se ha prescindido del paso tanto para la `i` como para la `j`, en este caso Mathematica supone que es 1. Se obtendrá como salida

```
Out[]={{2,3,4},{3,4,5},{4,5,6},{5,6,7},{6,7,8}}
```

1.6.1. Extracción de elementos o bloques de una matriz

Una vez construida una matriz, por ejemplo, la matriz inicial `matrizA`, podemos extraer el elemento de la posición (i, j) haciendo

```
matrizA[[i,j]]
```


Si lo que nos interesa es una fila, una columna o un bloque de una matriz, podremos extraer esa parte de la matriz usando el comando `Table[ ]` de la siguiente forma. Por ejemplo, dada la matriz `matrizA`, para extraer la segunda fila, la primera columna y el bloque correspondiente a las dos últimas filas y las dos últimas columnas, respectivamente, escribimos

```
In[]:=f2=Table[matrizA[[i,j]],{i,2,2},{j,1,4}]
In[]:=c1=Table[matrizA[[i,j]],{i,1,3},{j,1}]
In[]:=b=Table[matrizA[[i,j]],{i,2,3},{j,3,4}]
```

y obtenemos

```
Out[]={{1,1,1,1}}
Out[]={{1},{1},{0}}
Out[]={{1,1},{0,1}}
```

Hay que hacer notar que si se suprime alguno de los extremos de un índice, significa que el índice empieza por 1 y termina en el valor que permanece.

Se pueden añadir elementos a una lista con las funciones `Append[]` y `Prepend[]`. Si tenemos la matriz

```
aa={1,2,3}
```

tenemos que si escribimos

```
In[]:=Append[aa,-1]
In[]:=Prepend[aa,-1]
```

obtenemos

```
Out[]={1,2,3,-1}
Out[]={-1,1,2,3}
```

1.6.2. Dimensión

Las matrices `matrizA` y `B` que se han introducido anteriormente, son matrices de dimensiones 3×4 y 5×3, respectivamente, lo cual podemos saber utilizando el comando `Dimensions[ ]`. Escribiendo

```
In[]:=Dimensions[matrizA]
In[]:=Dimensions[B]
```

y se obtiene

```
Out[]={3,4}
Out[]={5,3}
```

Para eliminar la estructura de listas dentro de una lista dada se utiliza la instrucción `Flatten[ ]`. Así, dada la lista

```
CC={{a,b},{c}}
```

tenemos

```
In[]:=Flatten[CC]
Out[]={a,b,c}
```

1.7. Cálculo con matrices

Las operaciones básicas de suma y producto de matrices se realizan con los símbolos "+" y ".", respectívamente. A la hora de realizar un producto hay que tener en cuenta que las matrices sean compatibles. Por ejemplo, si usamos las matrices anteriores, se puede realizar el producto `B.matrizA` y no `matrizA.B`.

En esta sección vamos a introducir algunos de los comandos básicos relacionados con la teoría de matrices. Así, por ejemplo, tendremos:

`Transpose[a]` Da la matriz transpuesta de la matriz `a`.

`Det[a]` Da el determinante de la matriz `a`, siendo `a` una matriz cuadrada.

`Inverse[a]` Da la matriz inversa de la matriz `a`.

`IdentityMatrix[n]` Da la matriz identidad de tamaño $n \times n$.

`Join[a,b]` Da la matriz resultado de unir en columna las matrices `a` y `b`.

`DiagonalMatrix[lista]` Da la matriz diagonal que tiene como elementos de la diagonal los elementos de la lista.

Ejemplo 1.9 *Introducir las matrices*

$$a = \begin{bmatrix} 4 & 6 & -1 \\ 3 & 0 & 2 \\ 1 & -2 & 5 \end{bmatrix},\ b = \begin{bmatrix} 2 & 4 \\ 0 & 1 \\ -1 & 2 \end{bmatrix},\ c = \begin{bmatrix} 3 \\ 1 \\ 2 \end{bmatrix},$$

y calcular aquellas expresiones que estén bien definidas:

a) ab, ba, bc, $c^T b$, $b(ac)^T$ *y* $(ab)^T c$.

b) $\det(a)$, $\det(b)$, a^{-1}.

c) Utilizar las funciones `Transpose[]` *y* `Join[]` *para construir la matriz* $[a|I_3]$.

Introduciremos las matrices de la siguiente forma:

```
In[]:=a={{4,6,-1},{3,0,2},{1,-2,5}}
In[]:=b={{2,4},{0,1},{-1,2}}
In[]:=c={{3},{1},{2}}
```

Si calculamos las dimensiones de estas matrices,

```
In[]:=Dimensions[a]
In[]:=Dimensions[b]
In[]:=Dimensions[c]
```

obtenemos,

```
Out[]={3,3}
Out[]={3,2}
Out[]={3,1}
```

a) Podremos hacer los productos ab, $c^T b$ *y* $(ab)^T c$. *Para ello, introducimos*

```
In[]:=a.b
In[]:=Transpose[c].b
In[]:=Transpose[a.b].c
```

obteniendo

```
Out[]={{9,20},{4,16},{-3,12}}
Out[]={{4,17}}
Out[]={{25},{100}}
```

b) El determinante y la matriz inversa de b no existen porque no es una matriz cuadrada. Si introducimos

```
In[]:=Det[a]
In[]:=Inverse[a]
```

obtenemos

$\texttt{Out}[\,] = -56$
$\texttt{Out}[\,] = \left\{\left\{-\frac{1}{14}, \frac{1}{2}, -\frac{3}{14}\right\}, \left\{\frac{13}{56}, -\frac{3}{8}, \frac{11}{56}\right\}, \left\{\frac{3}{28}, -\frac{1}{4}, \frac{9}{28}\right\}\right\}$

Observad que cuando introduzcamos en el Mathematica alguna de las expresiones anteriores que no se puedan realizar, nos dará un mensaje de error, por ejemplo, si introducimos:

```
In[]:=Det[b]
```

obtenemos

```
Argument {{2,2},{0,1},{-1,2}}
at position 1 is not a square matrix
```

c) Si introducimos

```
In[]:=MatrixForm[Join[a,IdentityMatrix[3]]]
```

obtenemos la matriz

$$\begin{bmatrix} a \\ I_3 \end{bmatrix}$$

esto es,

$$\begin{bmatrix} 4 & 6 & -1 \\ 3 & 0 & 2 \\ 1 & -2 & 5 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Como

$$[\ a \mid I_3] = \begin{bmatrix} a^T \\ I_3 \end{bmatrix}^T,$$

tendremos que introducir


```
MatrixForm[Transpose[Join[Transpose[a],IdentityMatrix[3]]]]
```

y obtendremos

$$\begin{bmatrix} 4 & 6 & -1 & 1 & 0 & 0 \\ 3 & 0 & 2 & 0 & 1 & 0 \\ 1 & -2 & 5 & 0 & 0 & 1 \end{bmatrix}$$

1.8. Resolución de ecuaciones

El programa Mathematica cuenta con funciones que nos permiten resolver ecuaciones y sistemas de ecuaciones. Por ejemplo:

`Solve[ecuaciones,variables]` Resuelve formalmente ecuaciones y sistemas de ecuaciones poliómicas.

`NSolve[ecuaciones,variables]` Resuelve numéricamente ecuaciones y sistemas de ecuaciones polinómicas.

1.8.1. Igualdad en Mathematica

Hay que hacer notar que en el programa Mathematica para introducir una igualdad entre dos expresiones para construir una ecuación, se tiene que escribir un doble igual ==. Por otra parte, una sola vez el símbolo = representa una asignación o identificación.

Veamos cómo se hace uso de esto para la resolución de ecuaciones. Si queremos obtener las raíces de la ecuación polinómica

$$x^2 + 2x + 1 = 0$$

escribimos

```
In[]:=Solve[x^2+2*x+1==0,x]
```

obteniendo

```
Out[]={{x->-1},{x->-1}}
```

El resultado lo da como una lista de reglas de sustitución, donde, por ejemplo `x->-1` indica que la solución se obtiene sustituyendo `x` por -1.

Ejemplo 1.10 *Dado el sistema*

$$\begin{aligned} x+y-z &= 2 \\ 3x+3y+z &= 2 \\ x+z &= 0, \end{aligned}$$

para resolverlo introducimos las tres ecuaciones de la siguiente forma

```
In[]:=Solve[{x+y-z==2,3x+3y+z==2,x+z==0},{x,y,z}]
```

el resultado nos lo da como sigue

```
Out[]={{x->1,y->0,z->-1}}
```

En el caso de sistemas con algún grado de libertad, si los tratamos de resolver con la función `Solve[ ]`, el programa nos lo comunica y nos da la solución de las primeras incógnitas en función de la o las restantes.

Ejemplo 1.11 *Resolver el sistema*

$$\begin{aligned} 2x-y+z &= 0 \\ x+y &= 2 \end{aligned}$$

Al escribir

```
In[]:=Solve[{2x-y+z==0,x+y==2},{x,y,z}]
```

obtenemos el resultado:

```
Solve:: suars: Equations may not give solutions
for all ''solve'' variables.
```

$\{\{x \to \frac{2}{3} - \frac{z}{3}, y \to \frac{4}{3} + \frac{z}{3}\}\}$

Por otra parte, si escribimos

```
In[]:=Solve[{2x-y+z==0,x+y== 2},{y,z}]
```

nos da las soluciones de y y z en función de x, sin ninguna observación.

$$\{\{y \to 2-x, z \to 2-3x\}\}$$

Es conocido que, en general, las raíces de los polinomios de grado mayor o igual que 5 no tienen una expresión en términos de fórmulas elementales. No obstante, se pueden obtener aproximciones numéricas de las raíces con la función `NSolve[]`. Así, si escribimos

```
In[]:= NSolve[x^6 + 2x^2 + x - 1 == 0, x]
```

obtenemos

```
{{x -> -0.855444}, {x -> -0.709118 - 0.985817 I},
{x -> -0.709118 + 0.985817 I}, {x -> 0.495076},
{x -> 0.889302 - 0.900174 I}, {x -> 0.889302 + 0.900174 I}}
```

1.9. Ejercicios

1. Calcular 20! y aproximaciones numéricas para $\sqrt[5]{8}$, $(\frac{1}{5})^{99}$ y $e^7 + \ln(3)$.
2. Obtener una aproximación con 20 dígitos del número
$$\frac{(\sqrt[3]{5} - \sqrt{7} + 13)^7}{4568}.$$
3. Expresar en forma binómica los números complejos
$$\text{a)}\quad \frac{5}{(1-i)(2-i)(3-i)}\ ,\quad \text{b)}\quad (\sqrt{3}-i)^6\ ,\quad \text{c)}\quad \frac{3i^{30}-i^{19}}{2i-1}\ .$$
4. Obtener la parte real, la imaginaria y el argumento del número complejo
$$\frac{(3+4i)(2+i)}{(1+2i)(3-4i)}\quad .$$
5. *a)* Utilizando la función `Table[]`, formar la matriz, a, 3×3 cuyo término general es 2^{i+j}, donde $i = 1, 2, 3$ es el contador de las filas y $j = 1, 2, 3$ el contador de las columnas.

   b) Utilizando la función `Table[]`, construir una matriz, $f1$, que sea la primera fila de a y otra, $f2$, que sea la segunda fila de a. ¿ Son proporcionales?. ¿ Cuánto valdrá el deteminante de a?.

6. Calcular la matriz inversa, si existe, de las matrices

$$(a)\ d = \begin{bmatrix} 1 & -3 & 4 \\ 2 & -5 & 7 \\ 0 & -1 & 1 \end{bmatrix}, \quad (b)\ e = \begin{bmatrix} 2 & 4 & 6 \\ 4 & 5 & 6 \\ 3 & 1 & -2 \end{bmatrix}.$$

7. Construye una matriz `A` que sea una matriz de ceros 50×50, y una matriz

$$B = \left[\ I_{50} \,\middle|\, \mathtt{A} \,\middle|\, I_{50}\ \right]$$

donde I_{50} es la matriz identidad 50×50.

8. Resolver el sistema

$$\begin{aligned} x + 2y - z &= 0 \\ 3x - 3y + 2z &= 0 \\ -x - 11y + 6z &= 0. \end{aligned}$$

Clasifícalo y razona la respuesta.

Práctica 2

Introducción al programa Mathematica (II)

2.1. Introducción

El objetivo de esta práctica es continuar introduciendo algunas de las caracteríticas básicas del porgrama Mathematica. De este modo, aprenderemos algunas instrucciones para manipular expresiones algebraicas, a construir funciones de una o varias variables, obtener el valor de la función cuando se le da valores a la variable o variables independientes y representar gráficamente, tanto curvas como superficies.

2.2. Expresiones algebraicas

El Mathematica permite hacer operaciones con expresiones de forma simbólica. Por ejemplo, El comando `Simplify[]` trata de reducir una expresión algebraica a su forma más sencilla. Así, podemos hacer

```
In[]:= a=x + y + 1
In[]:= b=2*x +z +2
In[]:= Simplify[a+b]
Out[]=3+3x+y+z
```

Si un polinomio tiene raíces enteras y/o fracionarias, mediante la función `Factor[ ]` podemos descomponerlo en el producto de los binomios correspondientes a estas raíces. Por ejemplo,

```
In[]:=Factor[2*x^4-13*x^3+16*x^2+19*x-12]
```

nos da como resultado

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

De aquí se deduce que este polinomio tiene tres raíces enteras, $x=4$, $x=3$, $x=-1$, y una raíz racional $x=\frac{1}{2}$. Ahora bien, si un polinomio tiene raíces irracionales y/o complejas el comando `Factor[ ]` no nos proporciona la factorización del polinomio respecto a este tipo de raíces. Es decir, si queremos calcular las raíces del polinomio

$$P(x)=x^3+x^2-2x-2$$

al aplicar el comando `Factor[]`,

```
In[]:=Factor[x^3+x^2-2x-2]
```

obtenemos

```
Out[]=(1+x)(-2+x^2)
```

De aquí se deduce que este polinomio tiene una raíz entera $x=-1$ y las otras dos raíces serán irracionales o complejas ya que el comando `Factor[ ]` no ha factorizado el polinomio $-2+x^2$. En este caso, se trata de dos raíces irracionales.

También nos podemos plantear el problema inverso, es decir, obtener el desarrollo de un producto de polinomios. Por ejemplo, si escribimos

```
In[]:=Expand[(-4 + x)*(-3 + x)*(1 + x)*(-1 + 2x)]
```

obtenemos como resultado

$$-12+19x+16x^2-13x^3+2x^4$$

Los comandos `Expand[expr,Trig ->True]` o `TrigExpand[expr]` y `Factor[expr,Trig ->True]` o `TrigFactor[expr]` desarrollan o factorizan, respectivamente, expresiones trigonométricas. Por ejemplo, al introducir:

```
In[]:=Expand[Cos[a+b], Trig -> True]
```


o

```
In[]:=TrigExpand[Cos[a+b]]
```

obtenemos

```
Out[]=Cos[a] Cos[b]-Sin[a] Sin[b]
```

Al introducir

```
In[]:=Factor[Cos[a]Cos[b]-Sin[a]Sin[b], Trig -> True]
```

o bien

```
In[]:=TrigFactor[Cos[a] Cos[b]-Sin[a] Sin[b]]
```

obtenemos el resultado

```
Out[]=Cos[a+b]
```

2.3. Construcción de funciones

Nota.- Como ya hemos comentado, las funciones ya definidas en el programa Mathematica tienen el nombre empezando por mayúscula, es conveniente, con el fin de evitar duplicaciones, nombrar con minúsculas las funciones que construye el usuario.

Si deseamos construir una función $f(x)$ que le asigne a x una determinada expresión, haremos lo siguiente:

$$f[t\_] = \text{ expresión}$$

o

$$f[x\_] := \text{ expresión}$$

Por ejemplo, si queremos definir la función $y=x^2$ escribiremos:

```
In[]:=f[x_]:=x^2
```

Si queremos calcularla para un cierto valor, escribiremos

```
In[]:=f[2]
```

obteniendo

```
Out[]=  4
```

o bien, para un parámetro `t`, podemos escribir

```
In[]:=f[t]
```

obteniendo

```
Out[]=  t^2
```

La construcción de funciones de varias variables se hace de forma análoga a como se ha explicado anteriormente, basta separar las variables independientes con comas.

Ejemplo 2.1 *La velocidad de caída de un paracaidista con condiciones iniciales nulas* ($v_0 = t_0 = 0$)*, viene dada por la expresión*

$$v = \frac{mg}{k} - \frac{mg}{k}e^{-\frac{k}{m}t}.$$

Construir una función cuyas variables sean m, k, t *y con ella obtener la velocidad de caída de un paracaidista de* $70kg$ *al cabo de 10 segundos, si la constante* k *para su paracaídas es* $k = 0{,}5kg/s$*. Escribiremos la funcion* v*:*

```
In[]:=v[m_,k_,t_]:=m*9.8/k-m*9.8/k*Exp[-k*t/m]
```

y calcularemos el valor de `v` *para m=70, k=0.5 y t=10*

```
In[]:=v[70,0.5,10]
```

obteniendo

```
Out[]=94.5819
```


2.3.1. Funciones definidas a trozos

Para construir funciones definidas a trozos se emplea la notación

expresión /; condición

que indica que la función toma la *expresión* indicada "si" x satisface la *condición* establecida.

Por ejemplo, para definir la función

$$g(x) = \begin{cases} 1 & \text{si } -\frac{1}{2} \leq x \leq \frac{1}{2} \\ 0 & \text{si } |x| > \frac{1}{2} \end{cases}$$

escribiremos las instrucciones

```
In[]:=g[t_]:=1/;-1/2<=t<=1/2 ;
In[]:=g[t_]:=0/; Abs[t]>1/2 ;
```

Podemos comprobar que la función está bien definida tecleando `?g`.

Nota.- Observar que en el caso de una función definida a trozos necesitamos escribir el símbolo " := " en vez de " = ".

Cuando asignamos un valor a una variable o bien definimos una función, el Mathematica guarda el valor o la definición en la memoria durante toda la sesión. En ocasiones es interesante borrar la signación o la definición de la función de la memoria, para ello, se utiliza la función `Clear[ ]`. Por ejemplo, si introducimos

```
In[]:= a=5
In[]:= ?a
```

al ejecutar estas intrucciones obtenemos un mensaje indicándonos que `a` es una variable local y vale 5. Si hacemos,

```
In[]:= Clear[a]
In[]:= ?a
```

observamos que la variable `a` ya no toma el valor 5. De forma similar se procede para las funciones.

2.3.2. Asignación de valores

El comando `/.` nos permite sustituir en una expresion el valor de una variable por un valor usando una regla o conjunto de reglas de sustitución. Así tenemos

```
In[]:= x^2 + a x + 1 /. x->1
Out[]= 2 + a
```

y

```
In[]:= x^2 + a x + 1 /. {x->1, a->2}
Out[]= 4
```

Otros ejemplos son los siguientes. Al escribir:

```
In[]:=y[t_] = t^2 /. t -> x^3
```

sustituimos en t^2 la t por x^3, obteniendo la función $y(x) = x^6$.

También se puede usar para obtener el resultado de una función cuando a x se le asigna un valor. Por ejemplo, al escribir

```
In[]:=f[x]/. x->2
```

obtenemos el valor de $f(2)$,

```
Out[]= 4
```

donde $f(x)$ es la función $f(x) = x^2$ que habíamos definido anteriormente.

2.4. Representación gráfica de curvas y superficies

2.4.1. Curvas planas en coordenadas cartesianas

Para representar funciones de una variable se dispone de la función `Plot[]`. Su estructura es la siguiente,

$$\texttt{Plot}[\mathit{expr},\ \{x,\ \mathit{xmin},\ \mathit{xmax}\},\ \mathit{opciones}].$$

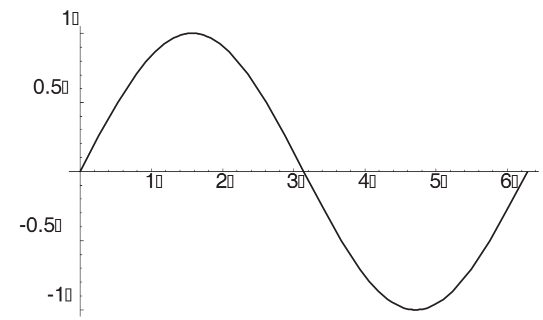
Como *expr* introducimos la función de x que deseamos dibujar; en $\{x,\ xmin,\ xmax\}$ introducimos el intervalo en el que varía x; y si consultamos la ayuda veremos que existen diferentes instrucciones que podemos añadir como opciones relativas, por ejemplo, la elección de escala, centro del sistema de referencia, etc...

Ejemplo 2.2 *Dibujar $y = \mathrm{sen}(x)$, con $x \in [0, 2\pi]$.*

Para ello, basta escribir

```
In[]:=Plot[Sin[x],{x,0,2*Pi}]
```

obteniendo



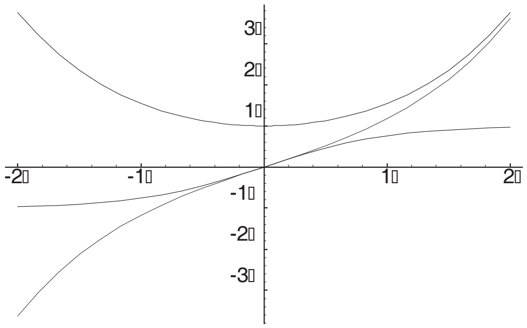
En el plano también es posible solapar la gráfica de varias curvas. Para poder distinguirlas podemos utilizar la opción:

```
PlotStyle->{{AbsoluteThickness[1]},AbsoluteThickness[1.5]}}
```

que nos dibujará cada curva de un grosor distinto. Por ejemplo, si queremos dibujar las funciones hiperbólicas, introduciremos

```
Plot[{Sinh[x],Cosh[x],Tanh[x]},{x,-2,2},
     PlotStyle->{{AbsoluteThickness[1]},
    {AbsoluteThickness[1.5]},{AbsoluteThickness[2]}}]
```

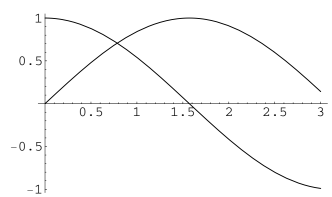
obteniendo



Una aplicación de la función `Plot[]` es el cálculo aproximado de las raíces reales de una función dada. Así, supongamos que se quiere saber dónde se cortan las funciones $f_1(x) = \text{sen}(x)$ y $f_2(x) = cos(x)$ en el intervalo $[0, 3]$. Para ello, basta dibujar las funciones y ver dónde se cortan. Se introduce la instrucción

```
Plot[{Sin[x], Cos[x],{x,0,3}]
```

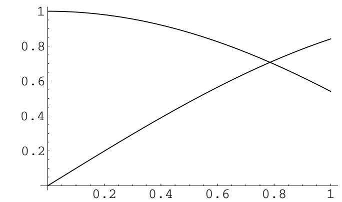
y se obtiene



Para mejorar la aproximación, podemos introducir

```
Plot[{Sin[x], Cos[x],{x,0,1}]
```

y se obtiene



con lo que obtenemos que el punto de corte está cercano al $x = 0{,}8$. Para mejorar la estimación, se puede utilizar la función `FindRoot[]`, utilizando el 0.8 como estimación inicial

```
FindRoot[Sin[x]-Cos[x]==0,{x,0.8}]
```

obteniendo como resultado

```
Out[]={x -> 0.785398}
```

2.4.2. Curvas en coordenadas paramétricas

Si la curva viene dada en coordenadas paramétricas:

$$x = x(t)$$
$$y = y(t),\ t \in\ [t_0,\ t_1],$$

se utiliza la función `ParametricPlot[]`, cuya estructura es la siguiente:

`ParametricPlot`$[\{x(t),\ y(t)\},\ \{t, tmin, tmax\}, opciones]$

Ejemplo 2.3 *Dibujar la curva dada por*

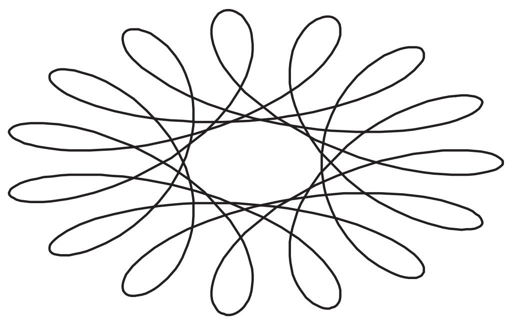
$$x = \quad 4\ \cos(-11\frac{t}{4}) + 7\cos(t),$$

$$y = \quad 4\ \text{sen}(-11\frac{t}{4}) + 7\text{sen}(t), \quad t \in [0, 8\pi].$$

Para ello, escribimos

```
ParametricPlot[{4 Cos[-11 t/4]+7 Cos[t],
4 Sin[-11 t/4]+7 Sin[t]}, {t,0,8 Pi},Axes->None]
```

(la opción `Axes->None`*, nos permite realizar la gráfica sin dibujar los ejes). De este modo, se obtiene la siguiente gráfica*



Con este comando también podemos dibujar varias curvas a la vez.

Ejemplo 2.4 *Representar gráficamente las curvas*

$$\left\{\begin{array}{l} x = 4\cos(t) \\ y = 4\mathrm{sen}(t) \end{array}\right. , \quad \left\{\begin{array}{l} x = 4\mathrm{ch}(t) \\ y = 4\mathrm{sh}(t) \end{array}\right.$$

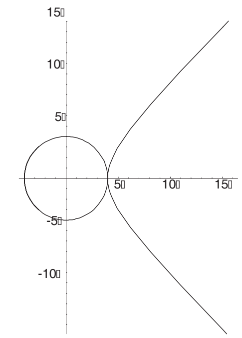
cuando $t \in [-4, 4]$.

Observar que se trata de las ecuaciones paramétricas de una circunferencia y de una hipérbola.

Introduciremos:

```
ParametricPlot[{{4 Cos[t],4 Sin[t]},
{4 Cosh[t],4 Sinh[t]}},{t,-4,4},AspectRatio->Automatic]
```

obteniendo



La opción `AspectRatio->Automatic` *nos ha servido para modificar la razón entre las escalas de los ejes. Se puede comprobar que si no se añade esta opción, por defecto, los dibujos aparecen deformados.*

2.4.3. Dibujo de curvas a partir de datos

En numerosas ocasiones es interesante saber la forma que presenta una función que viene definida a partir de una serie de datos. Para esto se utiliza la función `ListPlot[]`.

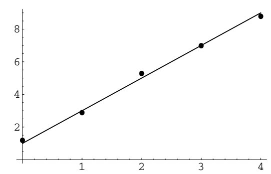
Ejemplo 2.5 *Se pretende comparar la función dada por los datos de la tabla*

x	*0.0*	*1.0*	*2.0*	*3.0*	*4.0*
y	*1.2*	*2.9*	*5.3*	*7.0*	*8.8*

con la función $y = 2x + 1$. *Para ello, hacemos*

```
In[]:= dibu1=Plot[2*x+1,{x,0,4}];
In[]:= lista={{0.0,1.2},{1.0,2.9},{2.0,5.3},
       {3.0,7.0},{4.0,8.8}};
In[]:=dibu2=ListPlot[lista];
In[]:= Show[dibu1,dibu2]
```

obtenemos la gráfica



2.4.4. Superficies

Para representar gráficamente una función de dos variables usaremos el comando `Plot3D`.

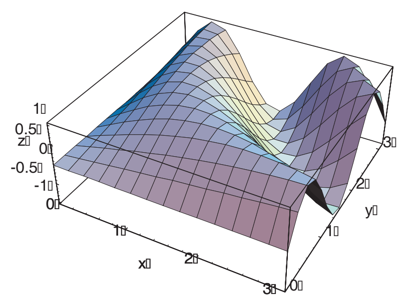
`Plot3D`[*expr*, $\{x, xmin, xmax\}$, $\{y, ymin, ymax\}$, *opciones*]

Las posibles opciones que se pueden incluir están explicadas en la ayuda, y nos pueden servir para dibujar o no los ejes, cambiar la escala, cambiar el punto de vista, etc...

Ejemplo 2.6 *Representar gráficamente la superficie* $z = \operatorname{sen}(xy)$, $x \in [0, \pi]$, $y \in [0, \pi]$. *Para ello, se escribe*

```
Plot3D[Sin[x*y],{x,0,Pi},{y,0,Pi},AxesLabel->{"x","y","z"}]
```

(la opción ***AxesLabel->***{ *"x ", ‘‘ y ","z "}* *, etiqueta los ejes coordenados). De este obtenemos la gráfica*



Con la opción


```
ViewPoint->{x0,y0,z0}
```

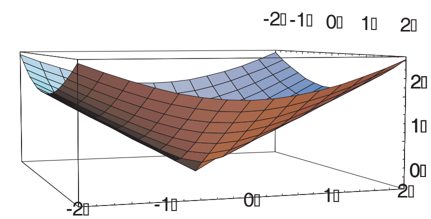
se puede cambiar el punto de vista del dibujo. Para la elección del nuevo punto de vista es de gran utilidad utilizar la ventana que se despliega al seleccionar **Input**, **3D View Point Selector** de la barra de herramientas. En esta ventana se elige la posición de los ejes y con `Paste`, pegamos la posición elegida dentro del comando `Plot3D`.

Ejemplo 2.7 *Representar gráficamente el cono* $z = \sqrt{x^2 + y^2}$, *con* $(x, y) \in D$, *siendo* $D = \{-2, 2\} \times \{-2, 2\}$.

Introducimos:

```
Plot3D[Sqrt[x^2+y^2],{x,-2,2},{y,-2,2},
        ViewPoint->{3.199,-1.067,0.279}]
```

obteniendo



2.5. Ejercicios

1. Calculad la expresión del binomio de Newton para

$$(a + b)^5 .$$

2. Hallar, usando el comando `Solve[]`, la intersección de las curvas $f_1(x) = -x^3$ y $f_2(x) = 2x - 12$. Comprobar este punto de corte dibujándolas simultáneamente, y sustituyendo el valor obtenido en las funciones, que previamente se habrán definido.

3. Comprobar gráficamente que la función $y(x) = 5 + e^{-2x}\cos(x)$ tiene una asíntota horizontal. ¿ Qué recta es?

4. Dadas las funciones $g(t) = t^3$ y $t(x) = \sqrt{x-5}$. Construir las funciones $h(x) = g(t(x))$ y $f(x) = g(x)t(x)$.

5. Representar la gráfica de las siguientes curvas o superficies:

   a)
$$f(x) = \begin{cases} x^3 & x \leq 0 \\ 1 & 0 < x < 2 \\ x^2 & x \geq 2 \end{cases}$$

   b) La elipse de semiejes $a = 2$ y $b = 5$.
$$\begin{aligned} &x = 2\cos(t) \\ &y = 5\,\mathrm{sen}(t), \quad t \in [0, 2\pi] \end{aligned}$$

   c) La esfera de centro $(0, 0, 0)$ y radio 5. (Utilizad una parametrización de la superficie esférica y la función `ParametricPlot3D[]`).

6. Obtén las raíces de la ecuación
$$x = x^2\,\mathrm{sen}(x) \ ,$$
comprendidas en el intervalo $[0, 10]$.

7. Encontrar la solución positiva más pequeña de la ecuación
$$\cos(3x) = \cos(7x) \ ,$$
en el intervalo $[0{,}5, 1{,}5]$ con una precisión de 6 cifras.