Introducción a *Mathematica*

Mathematica es un paquete de software matemático, de uso fácil y gran capacidad, que nos permitirá realizar las operaciones necesarias para resolver problemas que ya se abordaron en su momento en las diferentes asignaturas de matemáticas en cuatrimestres anteriores. En este primer apartado nos iniciaremos en el conocimiento básico de las funciones más importantes de Mathematica que nos serán imprescindibles para el posterior desarrollo de las prácticas de las diversas asignaturas. En cada práctica se hará un estudio de los comandos específicos que son necesarios para el desarrollo de las mismas

1.- Introducción

Cuando se comienza a utilizar el programa se activa el Front End con lo que se puede comenzar a introducir datos y expresiones que posteriormente serán evaluadas. En el momento en que se desea realizar la primera operación se activa el Kernel que es el módulo que realiza el cálculo. En el Kernel se encuentran introducidos los procedimientos de cálculo relacionados con una gran cantidad de operaciones (las más habituales). Sin embargo, dada la potencia de cálculo del paquete, operaciones y procedimientos más complejos se encuentran almacenados en diferentes Packages que deben ser activados antes de realizar los cálculos con ese tipo de sentencias.
Muchos de los comandos que ejecuta Mathematica han sido introducidos en ventanas gráficas llamadas Paletas que permiten una más rápida implementación de las sentencias y evitan memorizar parte de los comandos utilizados.
Todas estas características permiten un manejo sencillo y rápido de modo que, con una pequeña introducción a las funciones básicas, al modo de introducir datos y la utilización de la ayuda, se puede manejar con soltura en un corto espacio de tiempo.

2.- Introducción de datos y operaciones

Comenzar a trabajar con Mathematica es muy sencillo, basta con introducir la operación que se desea realizar y pulsar las teclas Shift + Intro. Así Mathematica realiza la operación. También ejecuta las operaciones pulsando el boton de Intro del teclado numérico.

```
2 * 3
```

```
6
```

De este modo las operaciones quedan numeradas en el orden en que se van realizando, con lo que se pueden ir utilizando los resultados previos con solo indicar en que momento se obtuvieron. El resultado de la última operación se puede recuperar utilizando el símbolo %, el penúltimo mediante %% y en general el resultado de la k-ésima operación con el símbolo %k.Por tanto, la siguiente operación hace referencia al resultado obtenido en la primera operación

```
(3 + 9) / %1
```

```
2
```

En *Mathematica* el producto de dos factores se puede representar tanto por el asterisco como por un espacio en blanco. Es importante recordar que cuando se opera con variables no es lo mismo escribir x y que xy ya que *Mathematica* en el primer caso entiende x*y mientras que en el segundo considera una nueva variable llamada (xy).
Las operaciones algebraicas comunes y el orden de evaluación de los diferentes factores sigue el mismo criterio que cualquier lenguaje de programación. En cualquier caso siempre se puede controlar la operación que se realiza colocando paréntesis.

3.- Precisión en el cálculo

Mathemática intenta siempre llegar al resultado más aproximado en cada operacion que realiza y siempre que puede llega al resultado exacto. De este modo cuando se realiza el cociente de dos números enteros matemática da como resultado dicho cociente puesto que si sacara un número decimal perdería precisión.

```
3 / 5
```

$$\frac{3}{5}$$

Lo mismo sucede cuando se realizan operaciones con variables. Siempre que no se pueda simplificar Mathematica presenta el valor que se ha introducido.

```
(x^3 + 5) / x^2
```

$$\frac{5 + x^3}{x^2}$$

En ocasiones puede interesar llegar a un resultado aproximado en vez de al resultado exacto, para hacerse una idea del orden de magnitud del resultado obtenido. Para ello se pueden aplicar tres métodos

a) Introducir alguno de los factores de la operación que se desea realizar de forma aproximada. Para ello basta con poner en forma decimal algún factor. Ej: 10 = 10.0 (con poner tan solo el punto decimal es suficiente)

```
3. / 5
```

```
0.6
```

b) Utilizar el comando N[ ]. Con este comando se puede obtener el resultado de forma aproximada e incluso indicar el número de decimales que se desea que aparezcan. Primero se introduce la operación a realizar y separado con una coma el número de dígitos significativos.

```
N[20 / 17, 5]
```

```
1.1765
```

c) Utilizar el comando N pero al final de la expresión que se desea evaluar de la siguiente manera expr //N (Por defecto *Mathematica* presenta 6 dígitos significativos).

```
20 / 17 // N
```

```
1.17647
```

4.- Comandos y variables predefinidas

Mathematica incorpora tanto sus propios comandos como algunas variables de uso muy frecuente de modo que la primera letra es siempre mayúscula. Para evitar confusiones o errores entre las variables definidas por el usuario y las ya definidas por el programa se recomienda definir variables propias que comiencen por letras minúsculas. Entre las variables definidas

en el programa se encuentran Pi, E, I.

```
N[Pi]
```

```
3.14159
```

```
N[E]
```

```
2.71828
```

```
√(-1)
```

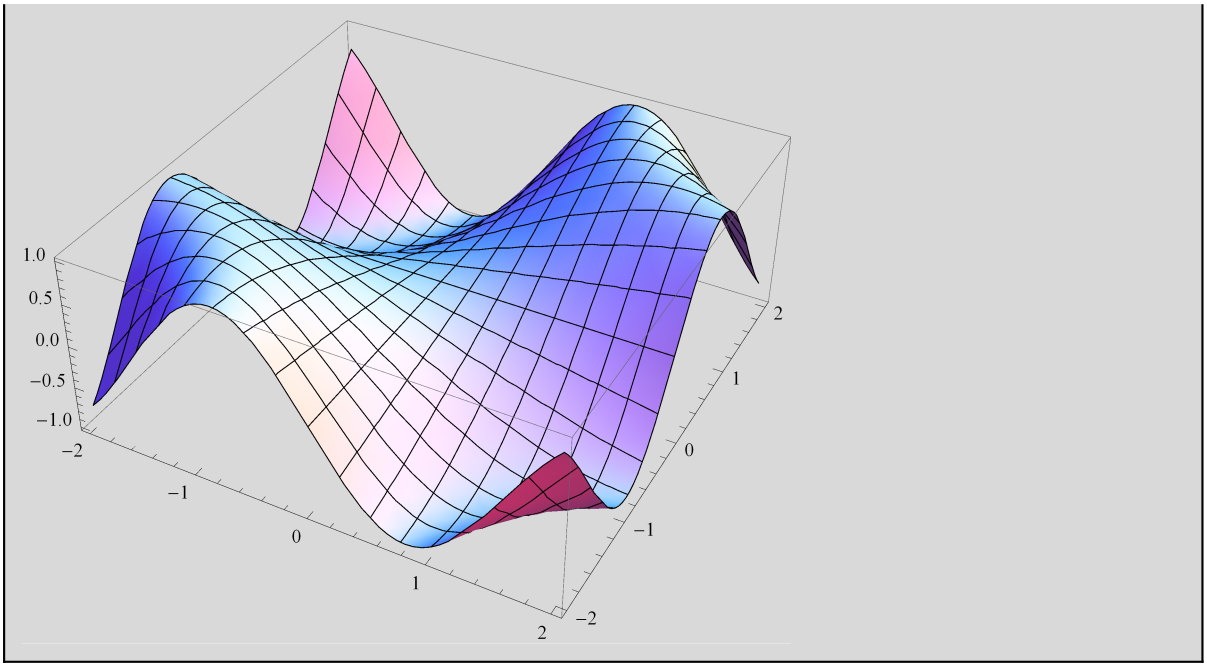
```
i
```

En cuanto a los comandos, todos ellos comienzan también con mayúscula y además el argumento que introducimos así como el resto de datos para el control de la operación van entre corchetes. Comando[arg , control_1, control_2, . . .]

```
Cos[Pi / 3]
```

$$\frac{1}{2}$$

```
Plot3D[Sin[x * y], {x, -2, 2}, { y, -2, 2}]
```



4.1.- Funciones Matemáticas Comunes

Las funciones matemáticas más comunmente utilizadas pueden introducirse directamente atendiendo al comando o también mediante las paletas que proporciona *Mathematica*.

Para uitlizar estas Paletas basta con situarse en la barra superior en el comando File y en Palettes seleccionar la paleta Basic Calculations. En esta paleta aparecen tanto funciones como comandos muy utilizados.
Por ejemplo, para indicar que se desea realizar una raiz cuadrada se puede utilizar el comando Sqrt[ ] o también se puede utilizar el icono correspondiente que aparece en la paleta $\sqrt{\square}$ en este caso basta con rellenar el recuadro con la expresión que se desea evaluar.
En esta paleta también aparecen funciones trigonometricas tanto directas como inversas Sin[ ] , ArcCos[ ] , y las correspondientes expresiones Hiperbólicas Cosh[ ], ArcTanh[ ].
Repasando esta paleta se pueden encontrar funciones como el factorial !, el entero mas cercano a x Round[x], etc.

Ejercicios de las secciónes 2,3,4

- Utilice **N** para calcular π con 50 decimales.
- Utilice N para ver a que entero se aproxima E^{π}. A continuación utilizar los comandos Floor[E^{π}] y Ceiling[E^{π}].
- Calcule dos números aleatorios con Random[ ] y a continuación multiplíquelos

5.- Listas

En muchas ocasiones es necesario utilizar datos que se encuentran relacionados entre si y que pueden ser todos ellos el argumento de una función. En esas ocasiones utilizamos listas para definir todos estos elementos.
Una lista contiene elementos que se encuentran separados por comas de la siguiente manera:

```
{1, 3, 6, 9, 12}
```

```
{1, 3, 6, 9, 12}
```

De este modo todos estos elementos se pueden utilizar como una única variable.

```
2^%
```

```
{2, 8, 64, 512, 4096}
```

En una lista se pueden introducir elementos que no sean del mismo tipo, como valores numéricos, variables, texto, etc. En estos casos deberemos referirnos a uno específicamente para poder realizar la operación correspondiente. Para ello se considera la posición que ocupa dicho elemento en la lista y se hace referencia a él de la siguiente manera:

```
a = {2.4, x, {2, 6}, Pi}
```

```
{2.4, x, {2, 6}, π}
```

Una vez definida la lista pasamos a operar con los elementos de la misma. La expresión a[[i]] hace referencia al elemento i-esimo de la lista.

```
3 * a[[1]]
```

```
7.2
```

```
a[[2]]^3
```

$$x^3$$

```
Cos[a[[4]]]
```

```
-1
```

```
2 * a[[3]]
```

```
{4, 12}
```

El tercer elemento de la lista es a su vez una lista por lo que se puede hacer referencia al elemento en si, a[[3]], o a cada uno de los elementos que lo componen a[[3,1]] y a[[3,2]] pudiendo operar con ellos por separado

```
a[[2]]^a[[3, 1]]
```

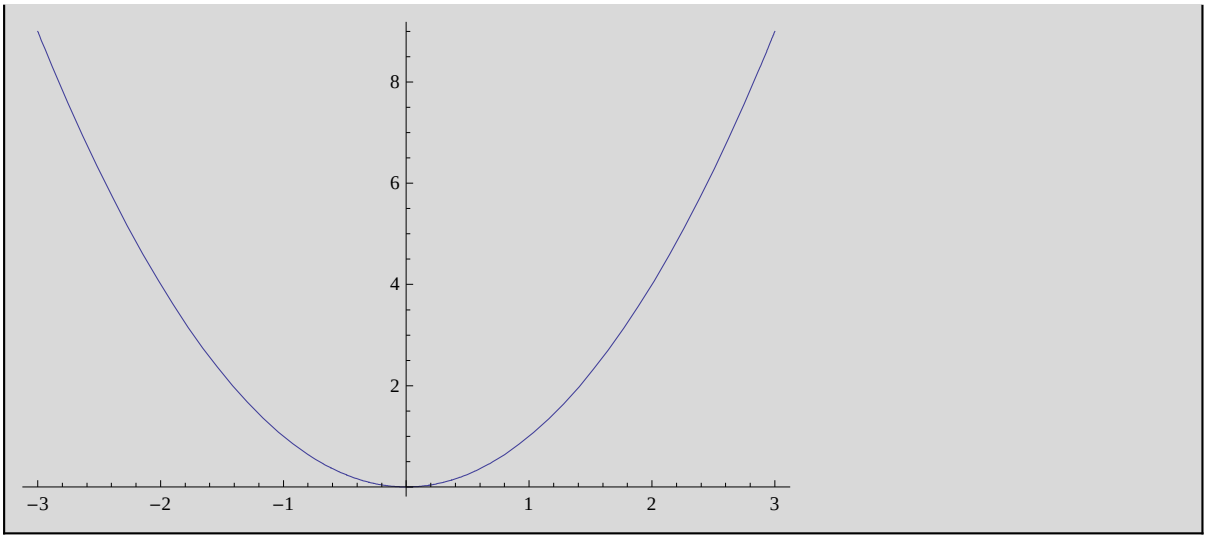
$$x^2$$

```
a[[2]]^a[[3, 2]]
```

$$x^6$$

Este tipo de listas tienen gran aplicación, para dar límites de integración, zonas donde se desea dibujar una función y fundamentalmente para el trabajo con matrices. Toda matriz se expresa como una lista, de modo que se puede definir un vector como una lista unidimensional {1,2,3}mientras que una matriz es una lista bidimensional {{1,3,5},{2,4,6}} (matriz 2x3). *Mathematica* cuando devuelve resultados también los devuelve en forma de lista, por ejemplo cuando proporciona las raices de un polinomio.
Ejemplos:

```
Plot[x^2, {x, -3, 3}]
```




```
Integrate[x^2, {x, 0, 2}]
```

```
8
3
```

```
a = {{1, 3, 5}, {2, 4, 6}}
```

```
{{1, 3, 5}, {2, 4, 6}}
```

```
MatrixForm[a]
```

$$\begin{pmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{pmatrix}$$

```
Chop[Solve [x^3 + 2 x^2 - 5 x - 4 == 0, x] // N]
```

```
{{x → 1.85577}, {x → -3.17741}, {x → -0.678363}}
```

Las operaciones 24 y 25 hacen referencia a operaciones con matrices. La 24 define una matriz de dos filas y tres columnas mientras que MatrixForm[ ] representa la Lista anterior en forma matricial. La última operación calcula las raices de la ecuación y devuelve una lista de tres listas cada una de ellas con un elemento (una solución de la ecuación).
Como veremos posteriormente, no será necesario memorizar todo este tipo de comandos puesto que una de las herramientas más útiles de *Mathematica* 3.0 (las paletas de comandos) nos permitirá introducir estas sentencias de otra forma

Utilizando el comandoTable[ ] se pueden generar listas sin dar explícitamente cada uno de los elementos que la componen. Basta con indicar la expresión a través de la que se genera y el número de elementos que la componen de la siguiente manera:
Table[expresión,{i, min,max,paso}] donde la expresión es una expresión en la que i variará entre los valores mínimo y máximo especificados y considerando el paso indicado en la lista.

```
Table[x^i, {i, 2, 9, 2}]
```

$$\{x^2, x^4, x^6, x^8\}$$

Se pueden generar de esta manera listas de varias dimensiones, introduciendo en vez de una única expresión una lista de expresiones

```
Table[{E^(j - 2), j * 2 / 3}, {j, 3, 5}]
```

$$\left\{\{e, 2\}, \left\{e^2, \frac{8}{3}\right\}, \left\{e^3, \frac{10}{3}\right\}\right\}$$

Si no se indica el paso se considera que vale uno. De igual manera, si no se indica el valor mínimo se comienza desde uno

```
Table[i, {i, 4}]
```

```
{1, 2, 3, 4}
```

Asi como conseguimos listas dimensiones introduciendo varias expresiones,también se pueden conseguir listas de este tipo introduciendo varios índices.

```
Table[i + j, {i, 1, 3}, {j, 2, 5}]
```

```
{{3, 4, 5, 6}, {4, 5, 6, 7}, {5, 6, 7, 8}}
```

```
MatrixForm[%]
```

$$\begin{pmatrix} 3 & 4 & 5 & 6 \\ 4 & 5 & 6 & 7 \\ 5 & 6 & 7 & 8 \end{pmatrix}$$

Se ha obtenido una matriz bidimensional en las columnas hacen referencia a todos distintos valores de j (2,3,4,5) mientras que las filas hacen referencia a los valores de i (1,2,3)

Ejercicios de la sección 5

- Utilice Table para hacer una lista de cinco nueves
- Utilice Table para hacer una lista conjunta de los cuadrados y los cubos de los numeros pares del uno al nueve
- Utilice Table para crear una lista con las potencias de x de 2 a 9 con paso 3
- Seleccione de la lista anterior x^5 y evalúelo para x = 2
- Crear con el comando Table una matriz 3 x 2

6.- Funciones definidas por el usuario

En general los ejercicios que realizaremos con *Mathematica* no serán únicamente operaciones aritméticas sino que se necesitará utilizar funciones definidas por el usuario para realizar el análisis de las mismas asi como para operar con elllas. Por ello se debe aprender a definir funciones de una o varias variables, así como evaluar las mismas para cualquier punto.

6.1.-Definición de funciones

Para definir funciones, igual que cualquier otra variable definida por el usuario, se recomienda utilizar palabras que empiecen por minúscula o letras minúsculas, para evitar que las variables definidas por el usuario se confundan con variables o comandos propios de *Mathematica*.
La definición de funciones es muy sencilla basta con indicar la expresión correspondiente de la siguiente manera:

```
f[x_] := x^3 + 2 x^2 + 3 x + 5
```

Esta sentencia no da ningún resultado puesto que tan solo se trata de una definición. Para ver que efectivamente la expresión que se ha introducido se corresponde con la deseada basta con preguntar cual es la función f que se ha definido, de la siguiente manera:

```
?f
```

```
Global`f
```

$f[x_] := x^3 + 2\,x^2 + 3\,x + 5$

Es fundamental definir de esta manera las funciones, colocando como variable independiente la variable x_, este símbolo permite considerar la variable x como una variable global que puede ser tanto un valor numérico como una expresión en la

que haya definidas otras variables

```
f[y - z]
```

$$5 + 3\,(y - z) + 2\,(y - z)^2 + (y - z)^3$$

```
f[3]
```

```
59
```

Del mismo modo se pueden asignar valores a las variables de las funciones a través de la expresión /. de la forma siguiente:

```
f[x] /. x -> 3
```

```
59
```

```
f[x] /. x -> u + v
```

$$5 + 3\,(u + v) + 2\,(u + v)^2 + (u + v)^3$$

El símbolo /. se puede considerar como "Tal que" de forma que la última operación se puede traducir como: calcular cuanto vale f[x] tal que x=a+b. Lo mismo que se ha realizado para una función de una variable se podría reañizar para funciones de varias variables.

```
m[x_, y_] := x^2 + 2 y x + 5 x + 2
```

```
m[x, y] /. {x -> 2, y -> 5}
```

```
36
```

```
m[x, y] /. {x -> r - s, y -> t}
```

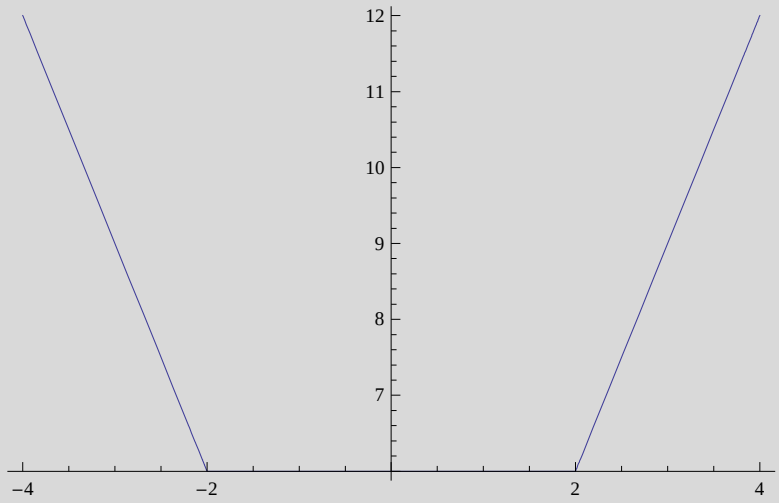
$$2 + 5\,(r - s) + (r - s)^2 + 2\,(r - s)\,t$$

Las funciones que se han definido hasta el momento tienen un dominio que para las funciones polinómicas, por ejemplo, es todo el conjunto de los números reales. Sin embargo se pueden definir también funciones a trozos, definiendo en cada caso en que dominio esta definida la función.

Realmente la fórma mas sencilla de definir una función a trozos es utilizando el comando Piecewise[{{val_1,$cond_1$},{val_2,$cond_2$},...,}]

```
h[x_] := Piecewise[{{-3 x, x < -2}, {6, -2 <= x <= 2}, {3 x, x > 2}}]
```

```
Plot[h[x], {x, -4, 4}]
```



El comando Piecewise se puede generar también a traves de la paleta mediante el gráfico $\begin{cases} \square & \square \\ \square & \square \end{cases}$

Para incorporar mas de dos condiciones basta con pulsar la teclas control+enter , lo que genera una condición mas.

La función h[x], podría haberse escrito por tanto como

```
h[x_] = Piecewise[{{-3 x, x < -2}, {6, -2 ≤ x ≤ 2}, {3 x, x > 2}}]
```

$$\begin{cases} -3x & x < -2 \\ 6 & -2 \le x \le 2 \\ 3x & x > 2 \\ 0 & \text{True} \end{cases}$$

6.2.- Operando con expresiones algebraicas

Cuando se opera con expresiones algebraicas *Mathematica* proporciona el resultado exacto (si puede conseguirlo). Sin embargo las expresiones que se obtienen pueden parecer muy complejas puesto que en ese proceso el paquete no se preocupa de reducir al máximo la expresión. Por ello siempre que las expresiones resulten complejas es muy ùtil considerar el comando Simplify[ ] que simplifica al máximo la expresión con que se trabaja. Por ejemplo:

```
x^3 + 3 x^2 + 3 x + 1
```

$$1 + 3x + 3x^2 + x^3$$

```
Simplify[%]
```

$$(1 + x)^3$$

```
∫ 1/(x^4 - 1) dx
```

```
-ArcTan[x]/2 + 1/4 Log[1 - x] - 1/4 Log[1 + x]
```

```
D[%, x]
```

```
-1/(4 (1 - x)) - 1/(4 (1 + x)) - 1/(2 (1 + x^2))
```

```
Simplify[%]
```

```
1/(-1 + x^4)
```

En las tres ùltimas operaciones se ha realizado una integral indefinida, a continuación se ha realizado la derivada del resultado y por ùltimo se ha simplificado la expresión
En este último ejemplo vemos como obtenemos el resultado inicial al derivar el resultado obtenido de la integral anterior, aunque si no hubieramos utilizado el comando Simplify[ ] resultaría difícil darse cuenta de ello.

Otra operación que se puede realizar es la de obtener los factores de una expresión algebraica, muy ùtil para obtener las raices de un polinomio, por ejemplo. Para ello se utiliza el comando Factor[ ]

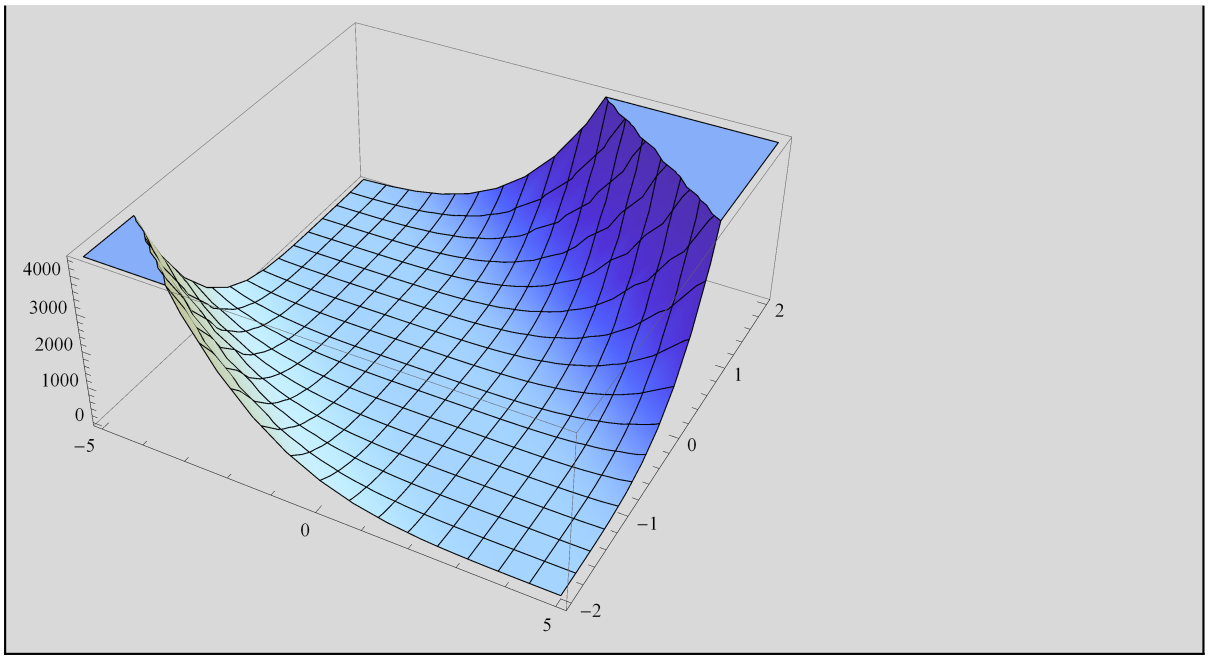
```
1 + 4 x + 6 x^2 + 4 x^3 + x^4 + 12 y + 36 x y + 36 x^2 y +
  12 x^3 y + 54 y^2 + 108 x y^2 + 54 x^2 y^2 + 108 y^3 + 108 x y^3 + 81 y^4
```

```
1 + 4 x + 6 x^2 + 4 x^3 + x^4 + 12 y + 36 x y + 36 x^2 y +
  12 x^3 y + 54 y^2 + 108 x y^2 + 54 x^2 y^2 + 108 y^3 + 108 x y^3 + 81 y^4
```

```
Factor[%]
```

```
(1 + x + 3 y)^4
```

```
Plot3D[%%, {x, -5, 5}, {y, -2, 2}]
```



Del mismo modo puede interesar en ocasiones obtener el desarrollo completo de una expresión algebraica que está simplificada. Para ello se utiliza le comando Expand[ ] como veremos en el siguiente ejemplo

```
(x^2 + 1)^2 * (x + 2)^3
```

$$(2 + x)^3 \left(1 + x^2\right)^2$$

```
Expand[%]
```

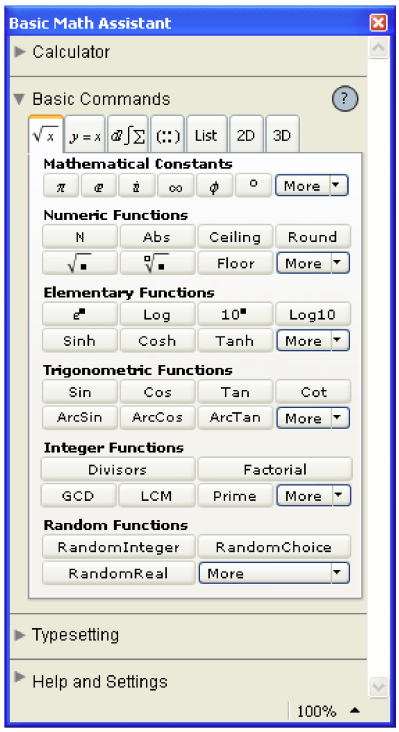
$$8 + 12\,x + 22\,x^2 + 25\,x^3 + 20\,x^4 + 14\,x^5 + 6\,x^6 + x^7$$

Ejercicios de la sección 6

- Representar como f(x) la función (x+2)$(x-1)^2\,(x-2)^3$
- Dibujar dicha función con el comando Plot entre x = -3 , x = 3
- Expandir la expresión de f(x) hasta obtener un polinomio de grado 6 en x
- Aplicar la expresión Factor [ ] para llegar a la expresión de partida.

7.- Paletas

Una de las mejoras sustanciales de Mathematica 3.0 en cuanto a la comunicación con el usuario es la presentación de paletas que permiten introducir sentencias y realizar operaciones de forma más sencilla. Para algunas sentencias de *Mathematica* como Sqrt[ ] (Raiz Cuadrada), Sum[ ] (Sumatorio), Product[ ] (Producto),etc. resulta complicado escribir en el notebook cual es la operación que se desea realizar, mientras que con las paletas estas operaciones vienen representadas por sus símbolos matemáticos por lo que basta con introducir los datos de la operación para que ésta se pueda llevar a cabo. Los símbolos que utilizan estas paletas para realizar las operaciones anteriores son: $\sqrt{\square}$,$\sum_{\square=\square}^{\square}\square$, $\prod_{\square=\square}^{\square}\square$. Introduciendo en los recuadros los valores deseados se completa la sentencia con lo que se puede realizar la operación. Basta con pulsar el tabulador para pasar de un recuadro a otro e ir llenando la sentencia completa.



En este sentido *Mathematica* introduce ya como predefinidas una serie de paletas en las que se representan tanto comandos, como operaciones, caracteres, matrices,etc.
Una de las aplicaciones mas comodas de las paletas es la definición de matrices puesto que partiendo de matrices 2 x 2 que aparecen en la paleta se puede llegar a conseguir matrices de cualquier dimensión. Se aumenta el número de filas con el comando " Control + Intro" mientras que el número de columnas aumenta al pulsar "Control + ,"

8.- Gráficos con *Mathematica*

Una herramienta de apoyo muy importante a la hora de resolver una gran cantidad de problemas matemáticos (sobre todo de una y dos variables) es la representación de funciones. De esta manera se pueden ver cuales son los extremos de una función, sus asíntotas, etc.
Mathematica dispone de una gran cantidad de comandos que permiten representar curvas y superficies. En este apartado se hará una exposición general de los comandos que se utilizarán posteriormente en las prácticas de las diferentes asignaturas. Debido a que la nomenclatura es exactamente igual se introducirán de forma conjunta las expresiones de dibujo en dos y tres variables.

8.1.- Plot, Plot3D

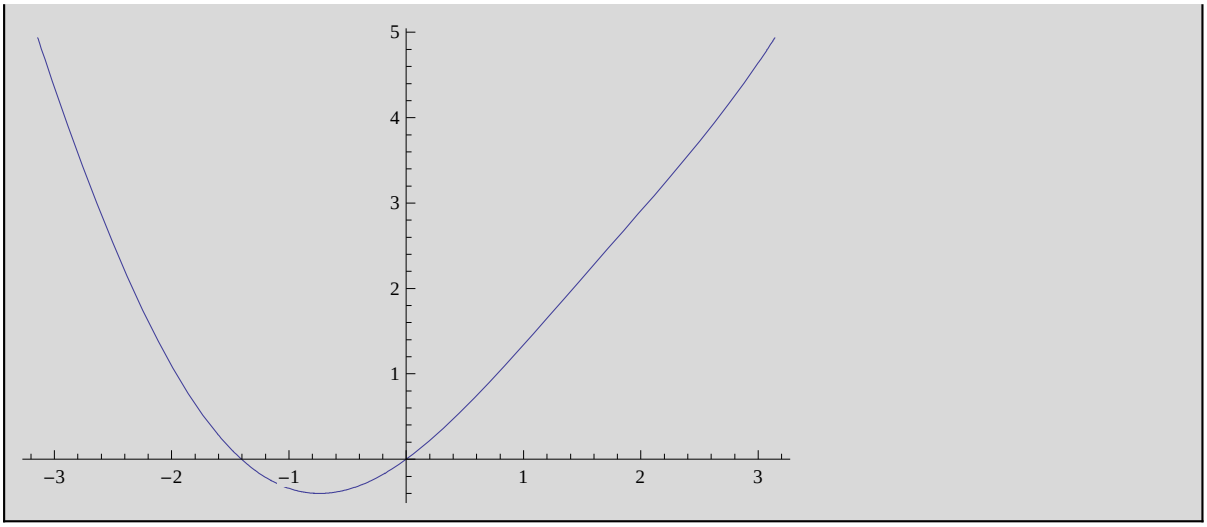
Estos comandos generan la función f en un recinto definido por el usuario. La forma de expresarlo es la siguiente:
Plot[f , {x,x_min,x_max}]
Plot3D[f , {x,x_min,x_max},{y,y_min,y_max}]
Ejemplos:
Definimos previamente dos funciones, una de una variable y otra de dos variables y posteriormente las representamos.

```
f[x_] := Sin[x] + x^2 / 2
```

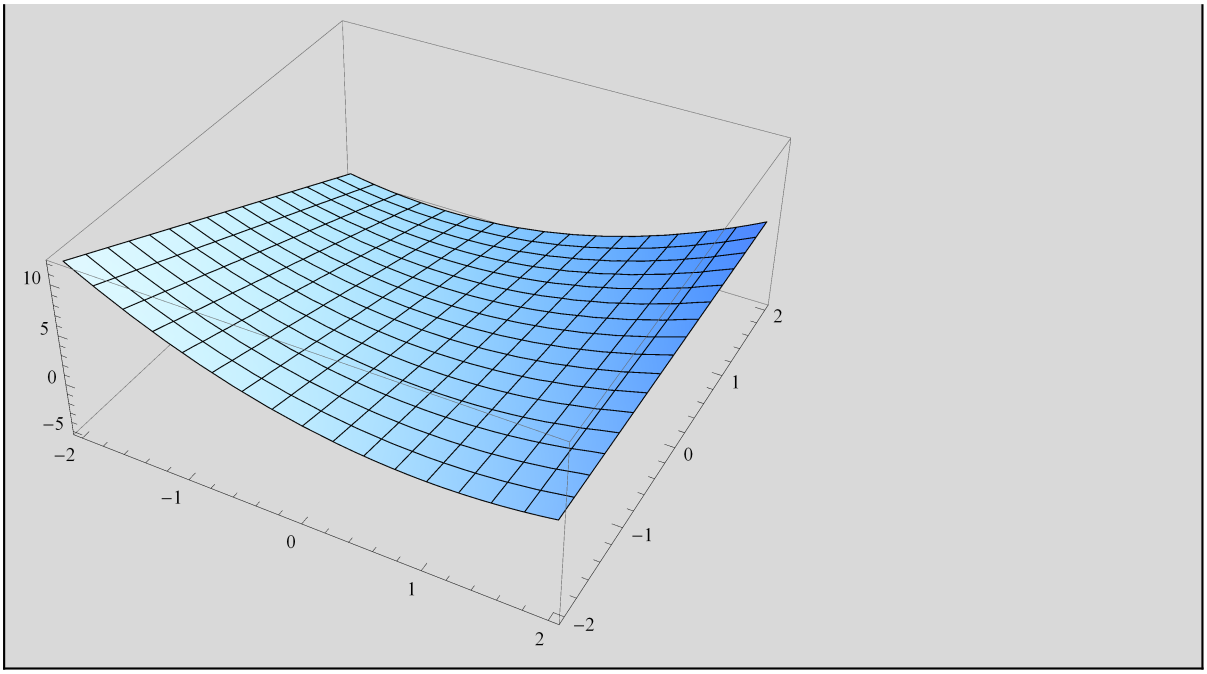
```
g[x_, y_] := x^2 + y * x - 2 y
```

A continuación pasamos a representar ambas funciones

```
Plot[f[x], {x, -Pi, Pi}]
```




```
Plot3D[g[x, y], {x, -2, 2}, {y, -2, 2}]
```



Estos dos comandos permiten dibujar funciones de la forma y = f(x) o z = g(x,y). Por otro lado una grán cantidad de funciones no vienen representadas de forma explícita e incluso es imposible despejar una de las variables en función del resto. Por ello se necesita otra serie de comandos que nos permitan representar esas funciones. Las otras dos formas en que mas habitualmente se presentan las funciones son la forma paramétrica y la forma implícita. A continuación veremos como se representan las funciones cuando viene definidas de esta manera.

8.2.- ParametricPlot, ParametricPlot3D

Estos comandos permiten representar funciones que vienen definidas a través de parámetros. Una función de dos variables se representaría de la siguiente manera:
x = x(t)

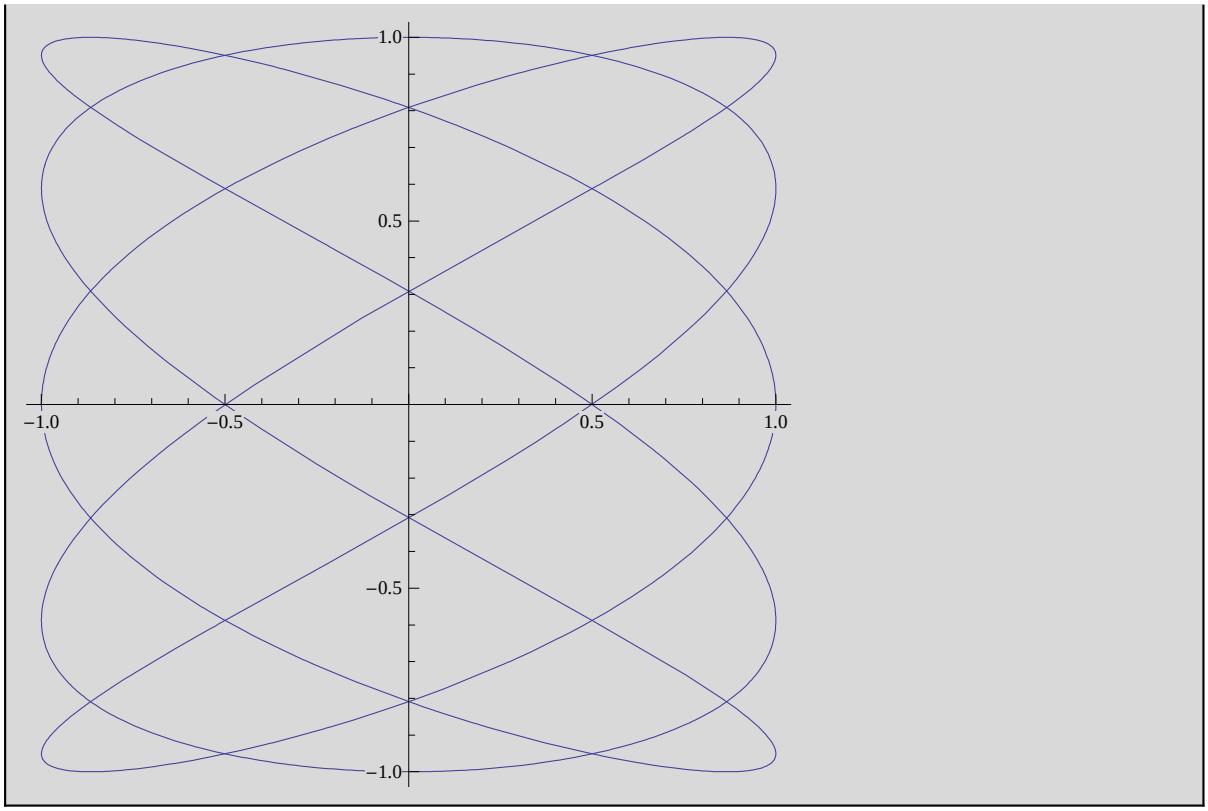
y = y(t)A medida que varía el parametro se obtienen puntos de la curva.

Para el caso de tres dimensiones a través de expresiones paramétricas se pueden representar tanto curvas como superficies. Una curva se representa de la siguiente manera:
x = x(t)
y = y(t)Variando t se obtiene una familia simplemente infinita de puntos (una curva)
z = z(t)

Para representar una superficie se necesitan dos parámetros, de forma que se obtiene una familia doblemente infinita de puntos que generan la superficie. La superficie se reptresenta de la siguiente manera:
x=x(u,v)
y=y(u,v)
z=z(u,v)

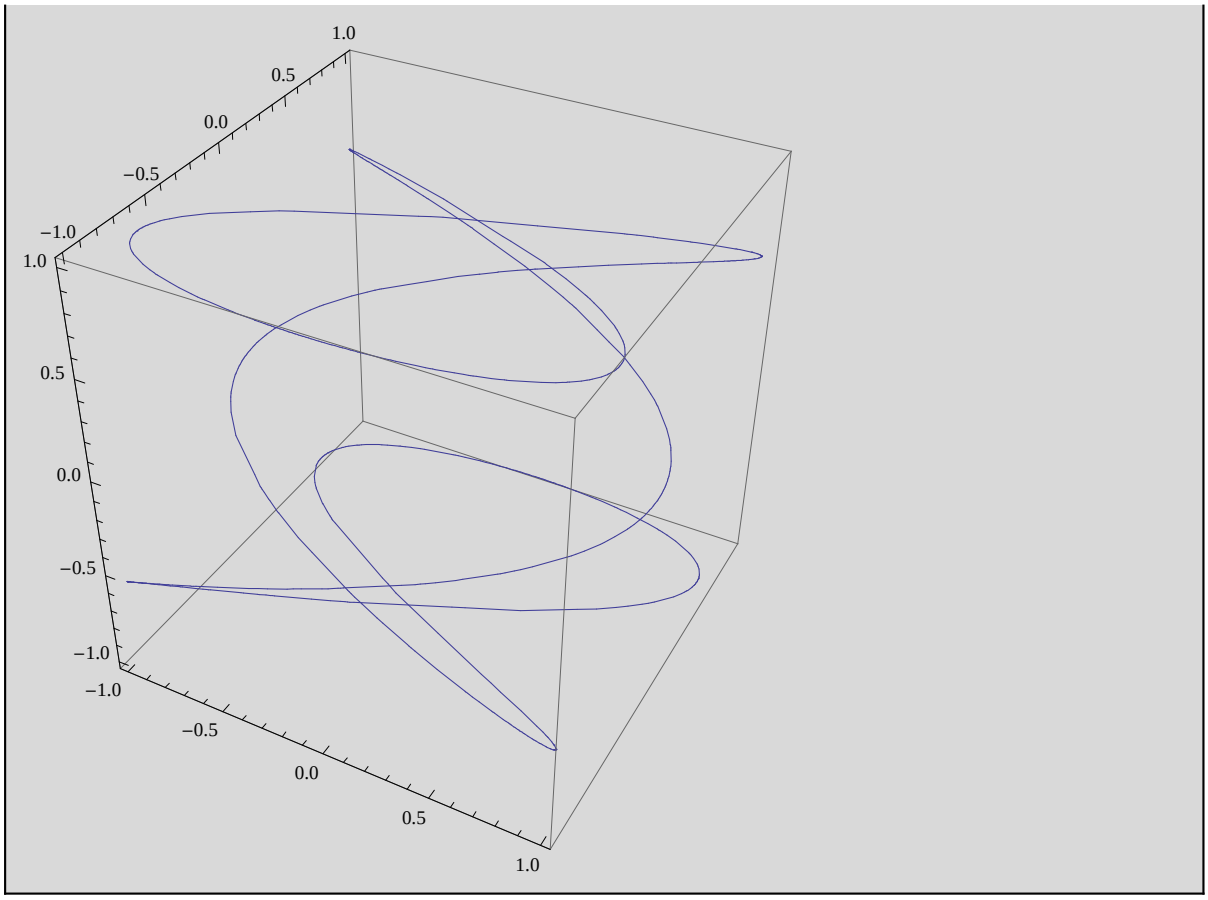
Vista esta introducción podemos presentar la forma de utilizar los comandos ParametricPlot y ParametricPlot3D.
ParametricPlot [{x,y},{t,t_min,t_max}]
ParametricPlot3D[{x,y,z},{u,u_min,u_max},{v,v_min,v_max}]

```
ParametricPlot[{Cos[5 t], Sin[3 t]}, {t, 0, 2 π}]
```



La función representada es x=Cos[5*t],y=Sin[3*t].
Para el caso de ParametricPlot3D veremos un ejemplo de una curva y otro de una superficie.

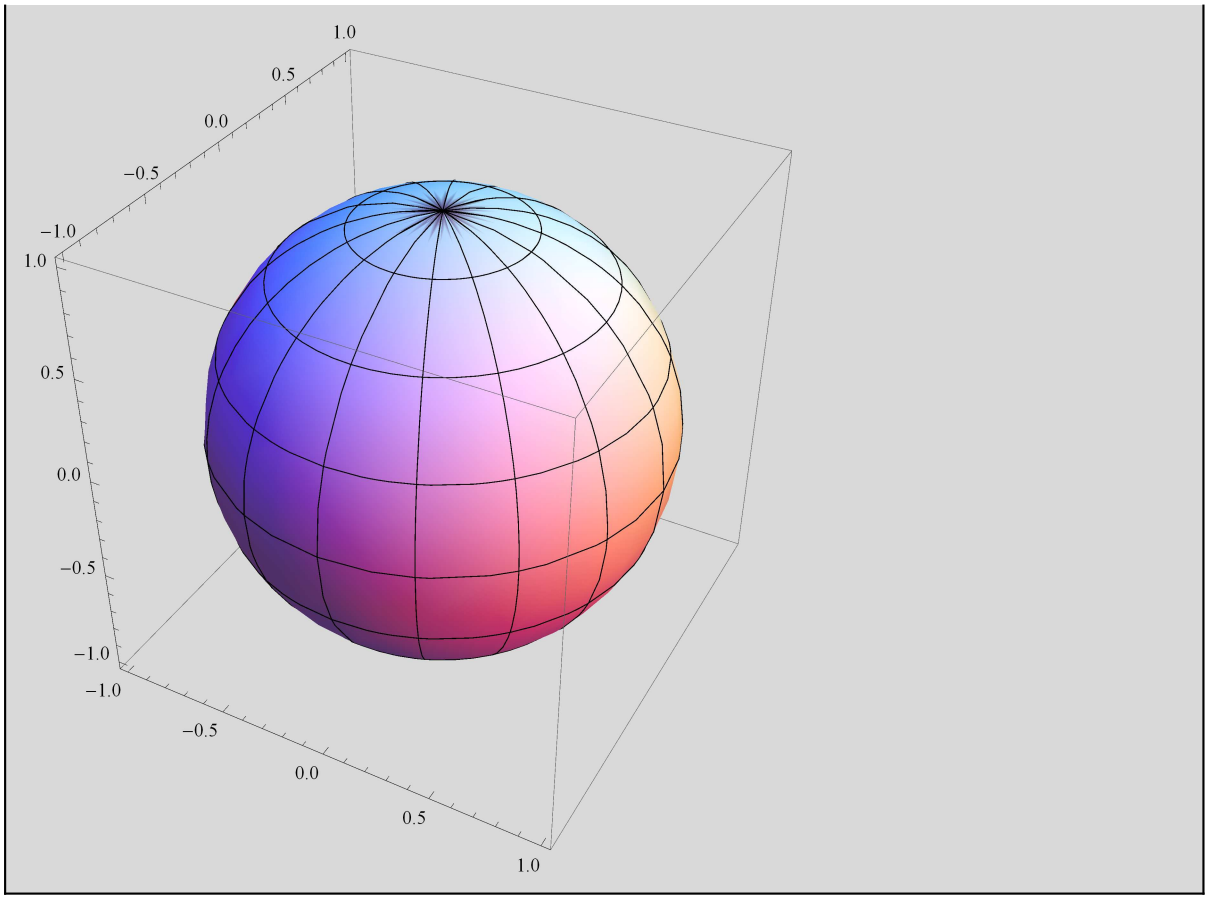
```
ParametricPlot3D[{Cos[5 t], Sin[3 t], Sin[t]}, {t, 0, 2 π}]
```



Se trata de una curva puesto que tan solo se tiene un parámetro. La función es de la forma:
x=Cos[5t]
y=Sin[3t]
z=Sin[t]

Para el caso de una superficie:

```
ParametricPlot3D[{Cos[u] * Cos[v], Sin[u] * Cos[v], Sin[v]},
 {u, 0, 2 * Pi}, {v, -Pi, Pi}]
```

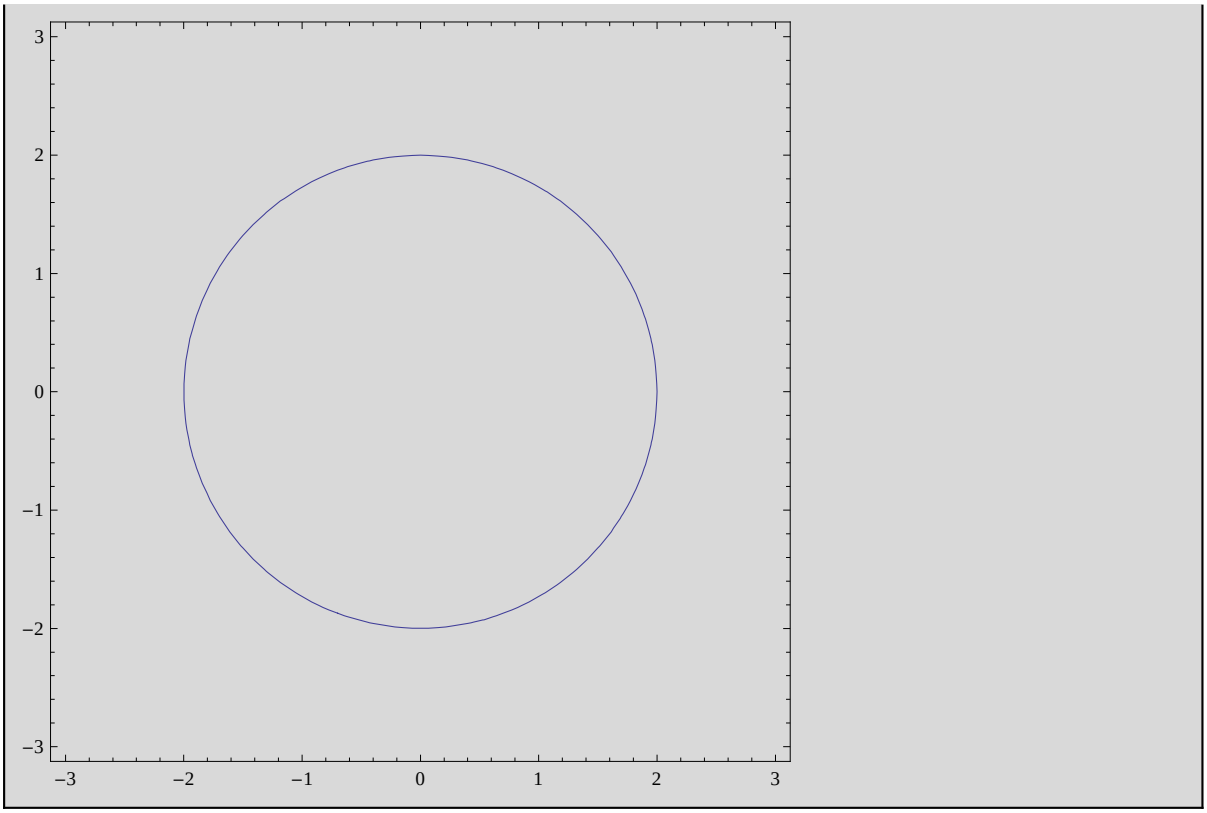


La superficie representada es :
x=Cos[u]*Cos[v]
y=Sin[u]*Cos[v]
z=Sin[v]

8.3.- ContourPlot

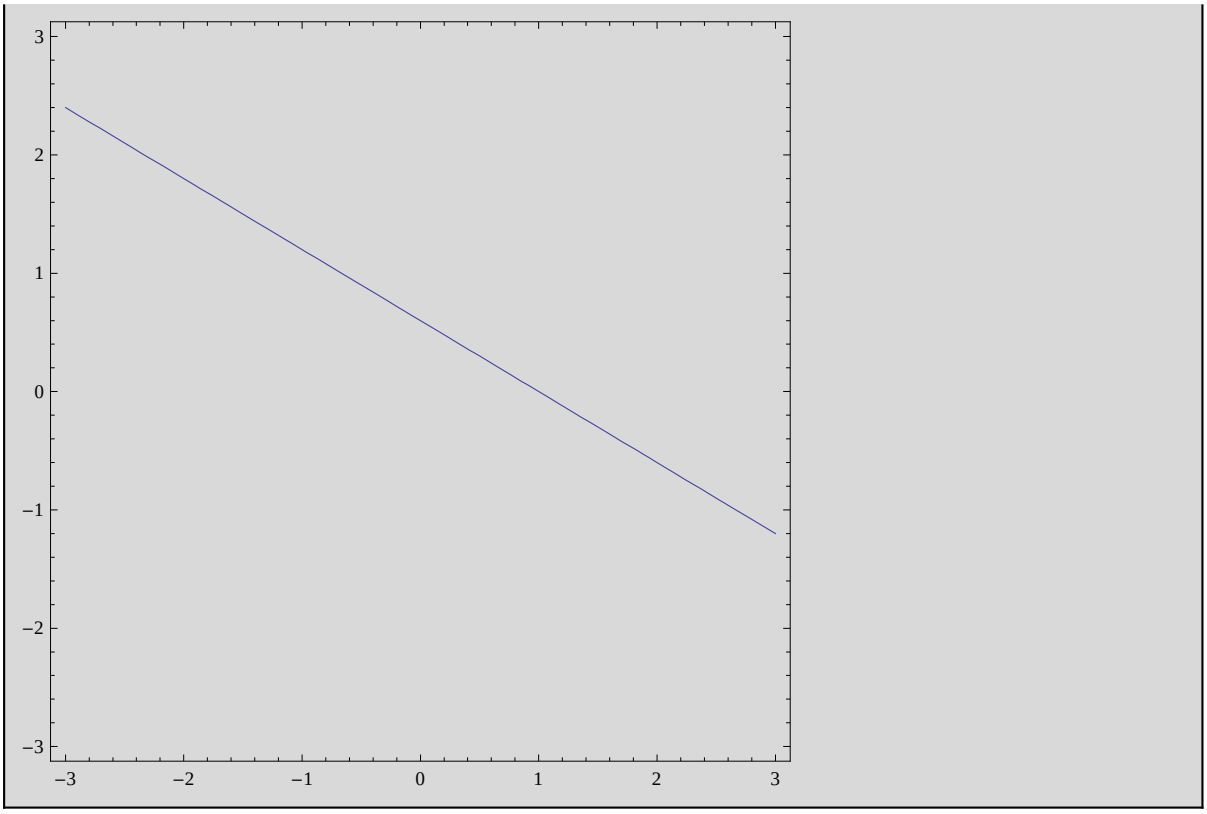
Este comando se emplea para representar funciones definidas de forma implicita de dos variables. P

```
ContourPlot[x^2 + y^2 - 4 == 0, {x, -3, 3}, {y, -3, 3}]
```

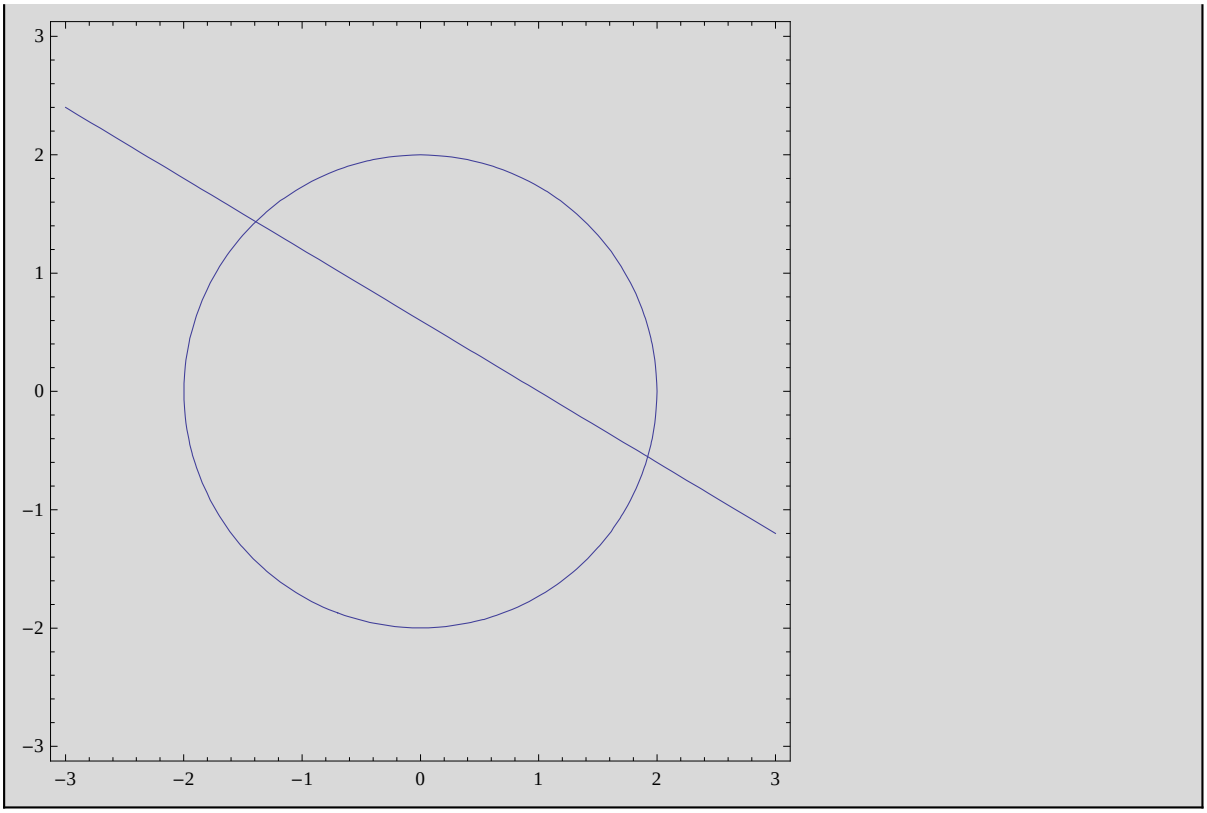



```
h[x_, y_] := 3 * x + 5 * y
```

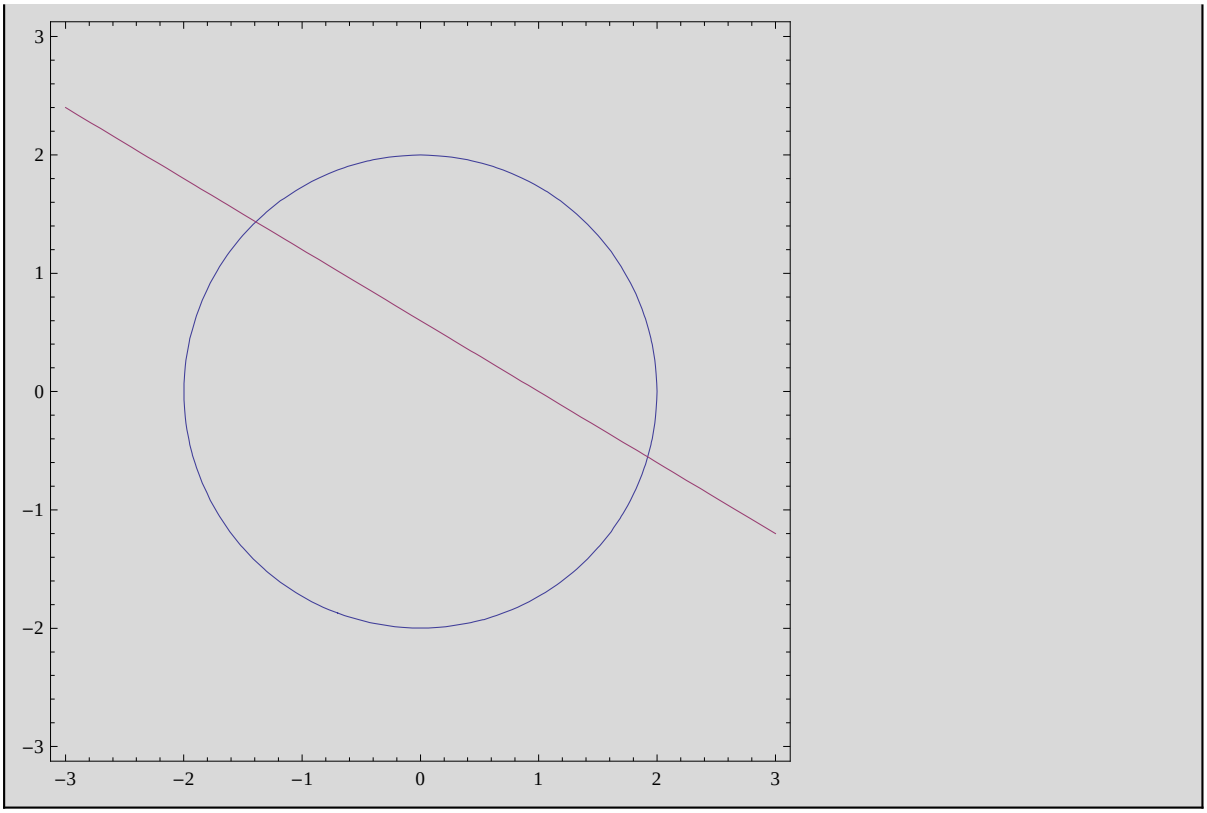
```
ContourPlot[h[x, y] == 3, {x, -3, 3}, {y, -3, 3}]
```



```
Show[%, %%%]
```

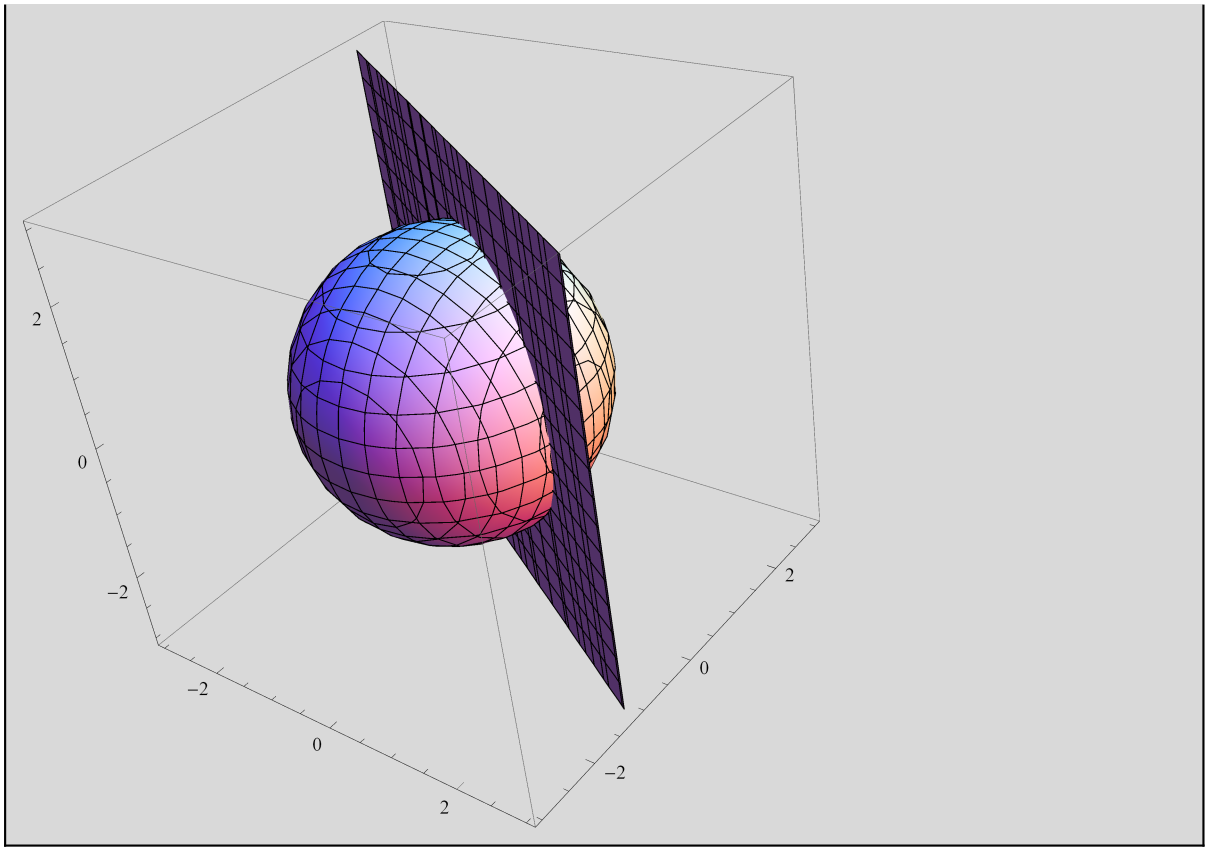



```
ContourPlot[{x^2 + y^2 - 4 == 0, h[x, y] == 3}, {x, -3, 3}, {y, -3, 3}]
```



En los dos últimos ejemplos se han presentado dos formas distintas de representar varias finciones en un mismo gráfico. El comando Show[ ] permite representar diferentes gráficos combinados.
Mientras que el comando ContourtPlot permite representar varios gráficos a la vez agrupados entre llaves

```
ContourPlot3D[{x^2 + y^2 + z^2 - 4 == 0, h[x, y] == 3},
 {x, -3, 3}, {y, -3, 3}, {z, -3, 3}]
```



Ejercicios de la sección 8

- Dibujar la función sen(x)/x así como la función sen(x) en el intervalo (-10,10)
- Dibujar la función x*y
- Dibujar la función paramétrica x = 4 Cos(-11t / 4)+7 Cos(t)
  y = 4 Sin(-11t / 4)+7 Sin(t) Desde 0 a 8Pi
- Dibujar la funcion x= Cos(u)Sin(v)
  y= Cos(u)Cos(v)
  z=v
- Dibujar la función : 4x^2+y^2=1

9.- Resolución de Ecuaciones

Mathematica dispone de un amplio grupo de sentencias que permiten resolver ecuaciones. Dependiendo del tipo de ecuación que se desee resolver y las variables y parámetros que tenga será conveniente utilizar uno u otro de los comandos que a continuación se exponen.

Roots[ecuacion,variable] permite obtener las raices de una ecuación polinómica en la variable que se indica en la expresión. Este comando funcionará correctamente siempre que se pueda obtener de forma exacta la raiz del polinomio.

```
Roots[(x + 2)^3 * (x - 3) == 0, x]
```

```
x == 3 || x == -2 || x == -2 || x == -2
```

```
Roots[(x + y - 1) * (x - 2) == 0, x]
```

```
x == 2 || x == 1 - y
```

```
Roots[(x + y - 1) * (x - 2) == 0, y]
```

```
y == 1 - x
```

La expresión Roots[ ] devuelve todas las raices de la ecuación.En caso de haber mas de una solución relaciona todas ellas mediante operadores lógicos (|| significa OR , && significa AND).
En caso que no se pueda obtener el valor exacto de la raiz se utilizarà el comando NRoots[ ] que proporciona una solución aproximada.

```
Roots[x^7 + x + 4 == 0, x]
```

```
x == Root[4 + #1 + #1^7 &, 1] || x == Root[4 + #1 + #1^7 &, 2] ||
  x == Root[4 + #1 + #1^7 &, 3] || x == Root[4 + #1 + #1^7 &, 4] ||
  x == Root[4 + #1 + #1^7 &, 5] || x == Root[4 + #1 + #1^7 &, 6] || x == Root[4 + #1 + #1^7 &, 7]
```

```
NRoots[x^7 + x + 4 == 0, x]
```

```
x == -1.16076 || x == -0.776478 - 0.89959 i || x == -0.776478 + 0.89959 i ||
  x == 0.226587 - 1.21468 i || x == 0.226587 + 1.21468 i ||
  x == 1.13027 - 0.566349 i || x == 1.13027 + 0.566349 i
```

En este último ejemplo se puede ver como intentando obtener las raices con el comando Roots no se llega a una solución por lo que hay que buscar la solución aproximada.
Para un caso mas general en que se desea resolver una ecuación cualquiera se utiliza el comando **Solve[ ]** que funciona de la misma manera que el comando Roots[ ]

```
Solve[(x + 2)^3 * (x - 3) == 0, x]
```

```
{{x → -2}, {x → -2}, {x → -2}, {x → 3}}
```

Los resultados se presentan en una lista en que se asignan a x un valor con el signo x->.
Cuando no se pueden obtener los valores exactos la solución queda en función del comando Root y se puede obtener la solución aproximada aplicando el comando N[ ]

```
Solve[x^7 + x + 4 == 0, x]
```

```
{{x → Root[4 + #1 + #1^7 &, 1]},
 {x → Root[4 + #1 + #1^7 &, 2]}, {x → Root[4 + #1 + #1^7 &, 3]},
 {x → Root[4 + #1 + #1^7 &, 4]}, {x → Root[4 + #1 + #1^7 &, 5]},
 {x → Root[4 + #1 + #1^7 &, 6]}, {x → Root[4 + #1 + #1^7 &, 7]}}
```

```
N[%]
```

```
{{x → -1.16076}, {x → -0.776478 - 0.89959 i}, {x → -0.776478 + 0.89959 i},
 {x → 0.226587 - 1.21468 i}, {x → 0.226587 + 1.21468 i},
 {x → 1.13027 - 0.566349 i}, {x → 1.13027 + 0.566349 i}}
```

Este Comando permite resolver ecuaciones que no sean polinómicas como por ejemplo:

```
Solve[Sin[x] Cos[x] == 0, x]
```

Solve::ifun: Inverse functions are being used by Solve, so some
solutions may not be found; use Reduce for complete solution information. >>

$$\left\{\{x \to 0\}, \left\{x \to -\frac{\pi}{2}\right\}, \left\{x \to \frac{\pi}{2}\right\}\right\}$$

En este caso se obtienen algunas soluciones triviales y devuelve un mensaje que indica la posible existencia de mas soluciones

Este comando Solve permite también resolver sistemas de ecuaciones que se representan en una lista o bien unidas a través del símbolo && (And). Ejemplos de este tipo se verán en las prácticas correspondientes al Algebra.
En muchas ocasiones se emplean parámetros que pueden ir variando según se desee y que mejoran o empeoran el comportamiento de un sistema. Por ello es muy interesante, poder resolver ecuaciones en las que se encuentren parámetros. Para ello el comando mas indicado en el paquete *Mathematica* es **Reduce[ ]**. La forma de presentarlo es idéntica a los anteriores. Veamos algún ejemplo:

```
Clear[a, x]
```

```
Reduce[3 a x^3 - 2 x + 3 == 0, x]
```

$$\left(a == 0 \&\& x == \frac{3}{2}\right) ||$$
$$\left(a \neq 0 \&\& \left(x == \text{Root}\left[3 - 2\,\#1 + 3\,a\,\#1^3 \&, 1\right] || x == \text{Root}\left[3 - 2\,\#1 + 3\,a\,\#1^3 \&, 2\right] ||\right.\right.$$
$$\left.\left. x == \text{Root}\left[3 - 2\,\#1 + 3\,a\,\#1^3 \&, 3\right]\right)\right)$$

En este ejemplo se ve claramente el funcionamiento del comando. Si a vale cero logicamente se trata de una ecuación en que solo hay una raiz. Si a es distinto de cero se obtienen tres raices distintas dado que estamos estudiando un polinomio de grado tres.
Este comando también se puede emplear cuando se trabaja con un sistema de ecuaciones como en el caso anterior, definiendo todos ellos entre llaves o con el símbolo &&. Ej:

```
Clear[a, r, s]
```

```
{{2, a}, {3, 1}}.{r, s} == {0, 1}
```

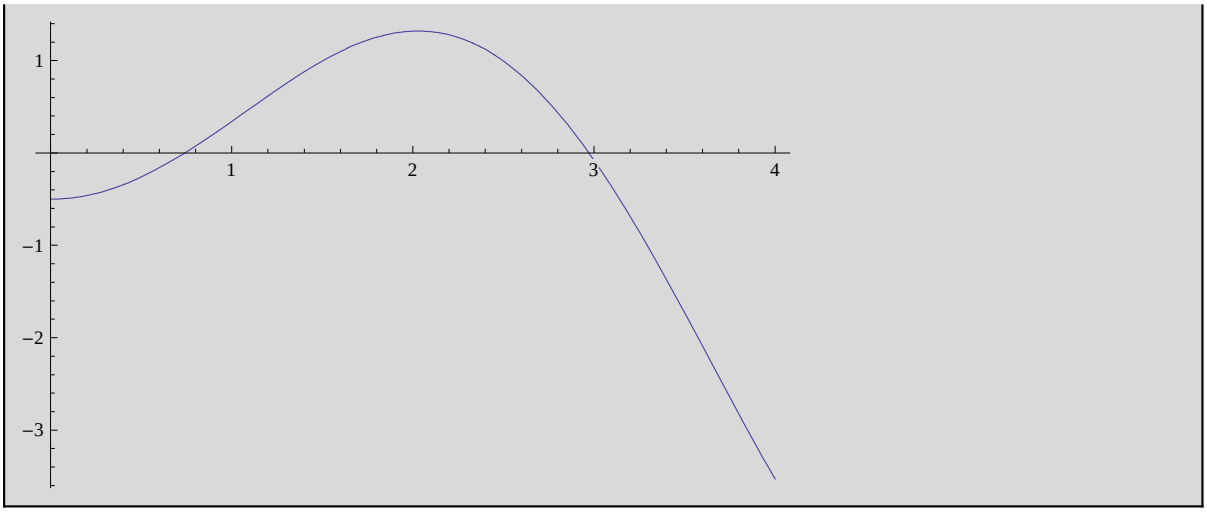
$$\{2\,r + a\,s,\ 3\,r + s\} == \{0,\ 1\}$$

```
Reduce[%, {r, s}]
```

$$-2 + 3\,a \neq 0 \&\& r == \frac{a}{-2 + 3\,a} \&\& s == 1 - 3\,r$$

Resuelve el sistema matricial considerando el valor del parámetro **a**. Lógicamente el valor del determinante de la matriz tiene que ser no nulo ya que el rango debe ser dos para que exista solución.
Cuando no se pueden obtener soluciones mediante los comandos que se han explicado se debe pasar a los métodos iterativos para la obtención de soluciones de la ecuación.
Para ello se utiliza el comando **FindRoot[ecuacion,{x,sol_aprox}]** en el que se indica la ecuación que se desea resolver y un valor cercano a la solución de la ecuación. Debido a esto se recomienda anteriormente dibujar la función para tener una idea aproximada de la posición de la solución. Ejemplo: Calcular las soluciones de la ecuación x Sin[x]-1/2= 0

```
Plot[x Sin[x] - 1 / 2, {x, 0, 4}]
```




```
FindRoot[x Sin[x] - 1 / 2 == 0, {x, 1}]
```

$$\{x \to 0.740841\}$$

```
FindRoot[x Sin[x] - 1 / 2 == 0, {x, 3}]
```

$$\{x \to 2.97259\}$$

Ejercicios de la sección 9

- Resolver x^2+2x+2=0
- Resolver x^2+a x+2=0
- Calcular raices de la ecuación Tan[x] == Cos[x] en el intervalo [0 , Pi]