

# Cálculo Simbólico con MATLAB. Introducción a la Programación.

Bloque IV: Gráficos 2D. Gráficos 3D.

Cristina Solares

Universidad de Castilla-La Mancha

18 de septiembre de 2022

El comando `plot(x,y)` dibuja una **gráfica a partir de valores numéricos**  $(x,y)$ . Primero se definen los vectores numéricos  $x$  e  $y$  con el mismo número de elementos. A continuación ejecutamos el comando `plot(x,y)`.

**Ejemplo 1:** Dibujamos la función  $y = 2x$  en el intervalo  $[0, 10]$

```
>> x=0:0.1:10;  
>> y=2*x;  
>> plot(x,y)
```

La primera orden se puede sustituir por

```
>> linspace(0,10,101)
```

que calcula 101 valores entre 0 y 10. Si no ponemos el número de valores, calcula 100.

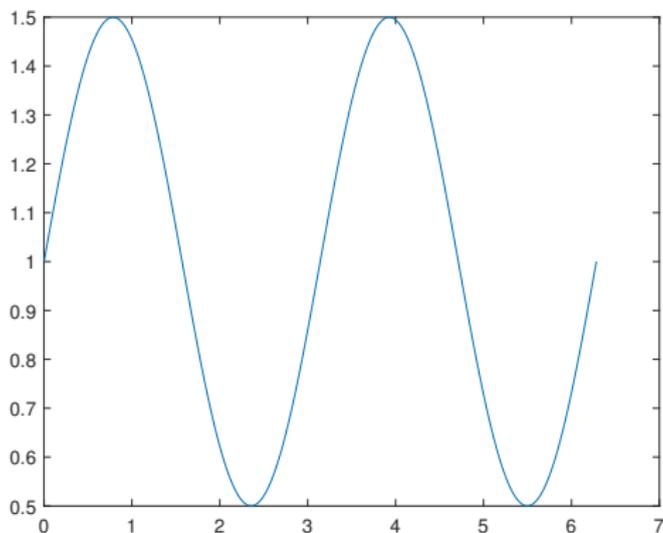
**Ejemplo 2:** Dibujamos las funciones  $y = x^2$  e  $y = x + 1$  en la misma figura

```
>> x=0:0.1:10;  
>> y=x.^2;  
>> plot(x,y)  
>> hold on  
>> y=x+1;  
>> plot(x,y)
```

Escribimos `hold on` para mostrar las dos gráficas en la misma figura.

**Ejemplo 3:** Dibujamos la función  $y = \sin(x) \cos(x) + 1$  en el intervalo  $[0, 2\pi]$

```
>> x=linspace(0,2*pi,200);  
>> y=sin(x).*cos(x)+1;  
>> plot(x,y)
```



Para dibujar una función  $f(x)$  con el comando `plot`, el usuario debe crear primero un vector de valores de  $x$  para el dominio en el que se va a dibujar la función. Luego, se crea un vector  $y$  con los valores correspondientes de  $f(x)$  utilizando cálculos elemento a elemento.

**Ejemplo 4:** Dibujar la función

$$f(x) = \begin{cases} -x - 2, & \text{if } x < -2 \\ 4 - x^2, & \text{if } -2 \leq x < 2 \\ x + 3, & \text{if } x \geq 2, \end{cases}$$

on the interval  $[-3, 3]$

```
>> x1=-3:0.1:-2;
>> y1=-x1-2;
>> x2=-2:0.1:2;
>> y2=4-x2.^2;
>> x3=2:0.1:3;
>> y3=x3+3;
>> x=[x1,x2,x3];
>> y=[y1,y2,y3];
>> plot(x,y)
```

**Ejemplo 5:** La temperatura máxima diaria en Nueva York durante el mes de Enero de 2001 viene dada por el siguiente vector:

$$\text{TNY} = [31, 26, 30, 33, 33, 39, 41, 41, 34, 33, 45, 42, 36, 39, 37, 45, 43, 36, 41, 37, 32, 32, 35, 42, 38, 33, 40, 37, 36, 51, 50].$$

Si queremos dibujar los datos anteriores hacemos:

```
>> x=1:31;
>> y=[ 31,26,30,33,33,39,41,41,34,33,45,42,36,39,37,45,43,36,...
41,37,32,32,35,42,38,33,40,37,36,51,50];
>> plot(x,y,'*')
```

Nótese que se puede modificar el tipo de línea: -, :, -; el tipo de punto: ., o,x,+,\*. También se puede cambiar el color: red (r), blue(b), yellow (y), green(g), black (b). Se pueden ver todas las opciones haciendo >> help plot.

```
>> x=1:31;
>> y=[ 31,26,30,33,33,39,41,41,34,33,45,42,36,39,37,45,43,36,...
41,37,32,32,35,42,38,33,40,37,36,51,50];
>> plot(x,y,'--r')
>> xlabel('x');
>> ylabel('y');
>> title('Datos de Nueva York');
```

Nótese que se puede colocar etiquetas en los ejes (xlabel e ylabel) y se puede poner un título a la gráfica con el comando title.

Se pueden ver todas las opciones con `>> help plot`.

Color	Marker	Line Style
y yellow	. point	— solid
m magenta	o circle	: dotted
c cyan	x <i>x-mark</i>	-. dashdot
r red	+ plus	-- dashed
g green	* star	
b blue	s square	
w white	d diamond	
k black	v triangle (down)	
	U triangle (up)	
	< triangle (left)	
	> triangle (right)	

Otras opciones:

- `text(x,y,'text')`: incluye "text" en el punto de coordenadas  $x$ ,  $y$ .
- `legend()`: define las etiquetas para las diferentes líneas utilizadas en la figura.
- `grid on/off`: activa/desactiva una rejilla en el gráfico.

El comando que utiliza Matlab para dibujar la **gráfica de una función o expresión simbólica (en una variable)** es `fplot`. La instrucción `fplot(f, [xmin, xmax])` dibuja la función  $f(x)$  en el intervalo  $[xmin, xmax]$ . Si no se indica el intervalo, Matlab dibuja la función en el intervalo  $[-5, 5]$ .

**Ejemplo 6:** Dibujamos la función  $f = \sin(x) \cos(x) + 1$  en el intervalo  $[0, 2\pi]$

```
>> syms x;  
>> f=sin(x)*cos(x)+1;  
>> fplot(f, [0,2*pi])
```

También se pueden definir la función anterior como una función anónima o como una función simbólica. Se pueden representar varias funciones en un mismo gráfico con el comando `hold on`.

**Ejemplo 7:** Dibujamos las funciones  $f = \sin(x)$  y  $g = \cos(x)$  en una misma figura

```
>> syms x g(x)  
>> f=@(x) sin(x);  
>> g(x)=cos(x);  
>> fplot(f(x), [0,2*pi])  
>> hold on  
>> fplot(g(x), [0,2*pi])
```

Para desactivar la opción anterior se escribe `hold off` o se crea una nueva figura con `figure`. Se pueden colocar etiquetas en los ejes (`xlabel` e `ylabel`) y poner un título (`title`) al gráfico de forma análoga a como se ha hecho con el comando `plot`. Se puede hacer una figura para cada gráfico haciendo `figure(1)`, `figure(2)`, ...

El comando que utiliza Matlab para dibujar la gráfica de una **función dada en implícitas** es `fimplicit`. La instrucción `fimplicit(f, [xmin,xmax,ymin,ymax])` dibuja la función  $f(x)=0$  en el intervalo  $[xmin,xmax]$  para la variable independiente e  $[ymin,ymax]$  para la variable dependiente.

**Ejemplo 8:** Dibujamos la función  $x^2 + (y^2/2) - 1 = 0$  en el intervalo  $[-2,2]$  para las variables  $x$  e  $y$ .

```
>> syms x y
>> f=x^2+(y^2/2)-1;
>> fimplicit(f,[-2,2,-2,2])
```

El comando que utiliza Matlab para dibujar la gráfica de una **función dada en paramétricas** es `fplot`. La instrucción `fplot(x,y,[tmin,tmax])` dibuja la función  $x=x(t)$ ,  $y=y(t)$  en el intervalo  $[tmin,tmax]$ .

**Ejemplo 9:** Dibujamos  $\{x = 2t \cos(t), y = 2t \sin(t)\}$  en el intervalo  $t \in [0, 4\pi]$ .

```
>> syms t
>> x=2*t*cos(t);
>> y=2*t*sin(t);
>> fplot(x,y,[0,4*pi])
```

El comando `mesh(X,Y,Z)` dibuja una **gráfica de malla a partir de valores numéricos**  $(X,Y,Z)$ . En primer lugar se generan los vectores  $x$  e  $y$  con las coordenadas en los correspondientes ejes. En segundo lugar se genera una retícula  $[X,Y]$  con el comando `meshgrid`. A continuación, se calcula el valor de  $Z$  en cada punto de la retícula. Por último se dibuja el gráfico de malla a partir de los vectores  $X,Y$  y  $Z$ .

**Ejemplo 10:** Dibujamos la función  $z = x + y$ .

```
>> x=-2:0.1:2;
>> y=-2:0.1:2;
>> [X,Y]=meshgrid(x,y);
>> Z=X+Y;
>> mesh(X,Y,Z)
```

**Ejemplo 11:** Dibujamos las funciones  $z = x^2 + y^2$  e  $z = x + y$  en la misma figura

```
>> x=-2:0.1:2;
>> y=-2:0.1:2;
>> [X,Y]=meshgrid(x,y);
>> Z=X+Y;
>> mesh(X,Y,Z)
>> hold on
>> Z=X.^2+Y.^2;
>> mesh(X,Y,Z)
```

Escribimos `hold on` para mostrar las dos gráficas en la misma figura.

**Ejemplo 12:** Dibujamos la función  $z = \sin(x) + \cos(y)$ .

```
>> x=0:pi/10:2*pi;
>> y=0:pi/10:2*pi;
>> [X,Y]=meshgrid(x,y);
>> Z=sin(X)+cos(Y);
>> mesh(X,Y,Z)
```

El comando `surf(X,Y,Z)` se utiliza de forma análoga al comando `mesh`, con la diferencia de que **genera una superficie tridimensional** en vez de una malla.

**Ejemplo 13:** Dibujamos la función  $z = 1 + \sin(x)$ .

```
>> x=0:pi/10:2*pi;
>> y=0:pi/10:2*pi;
>> [X,Y]=meshgrid(x,y);
>> Z=1+sin(X);
>> surf(X,Y,Z)
```

Para obtener las **curvas de nivel de una superficie** utilizamos el comando `contour(X,Y,Z)`. El comando `contour` se utiliza de forma análoga al comando `mesh`.

**Ejemplo 14:** Dibujamos las curvas de nivel de la función  $z = x^2 + y^2$ .

```
>> x=-2:0.1:2;
>> y=-2:0.1:2;
>> [X,Y]=meshgrid(x,y);
>> Z=X.^2+Y.^2;
>> contour(X,Y,Z)
```

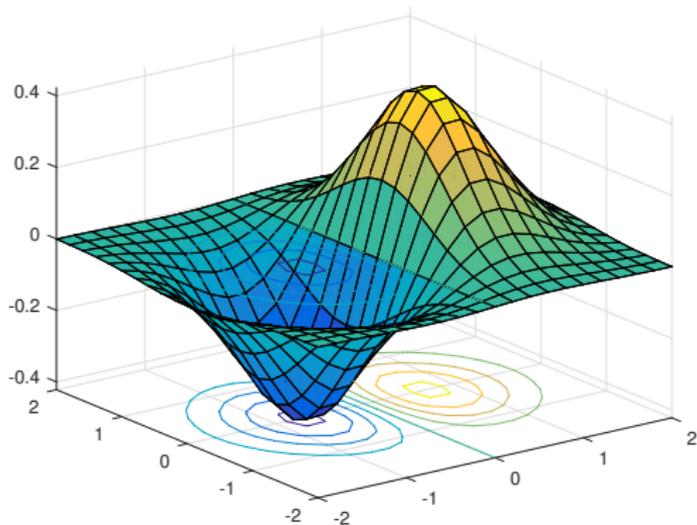
**Ejemplo 15:** Dibuja la función  $z = \sin(\sqrt{x^2 + y^2})/\sqrt{x^2 + y^2}$  con los comandos `mesh` y `surf`

```
>> u = -8:0.5:8; v = u;  
>> [U,V] = meshgrid(u,v);  
>> R = sqrt(U.^2+V.^2);  
>> W = sin(R)./R;  
>> mesh(U,V,W);pause(5);surf(U,V,W)
```

El comando `surf(X,Y,Z)` genera una **gráfica de superficie combinada con su gráfica de contorno**.

**Ejemplo 16:** Dibujamos la función  $z = xe^{-x^2-y^2}$ .

```
>> x=-2:0.2:2;  
>> y=-2:0.2:2;  
>> [X,Y]=meshgrid(x,y);  
>> Z=X.*exp(-X.^2-Y.^2);  
>> surf(X,Y,Z)
```



El comando que utiliza Matlab para dibujar la gráfica de una función de dos variables es `fsurf`. La instrucción `fsurf(f, [xmin,xmax,ymin,ymax])` dibuja la función  $f(x,y)$  en el intervalo  $[xmin,xmax]$  para  $x$ ,  $[ymin,ymax]$  para  $y$ . Si el intervalo es el mismo para las variables  $x$  e  $y$ , se puede escribir `fsurf(f, [min,max])`. Si no se indica el intervalo, Matlab considera el  $[-5, 5]$  para ambas variables.

**Ejemplo 17:** Dibujamos la función  $z = \sin(x) + \cos(y)$ .

```
>> syms x y
>> f=sin(x)+cos(y);
>> fsurf(f, [0,2*pi,0,2*pi])
```

**Ejemplo 18:** Dibujamos la función  $z = x e^{-x^2-y^2}$ .

```
>> syms x y
>> f=x*exp(-x^2-y^2);
>> fsurf(f, [-2,2])
```

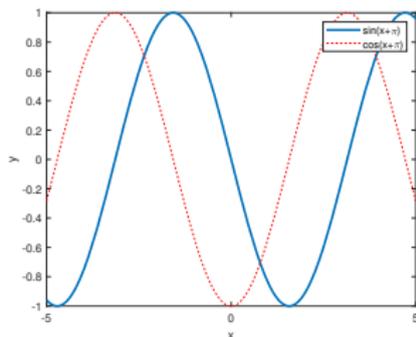
Para dibujar las curvas de nivel de una función  $f(x,y)$  se puede utilizar el comando `fcontour(f)`.

```
>> syms x y
>> fcontour(sin(x)+cos(y))
```

A continuación se muestran las gráficas de dos funciones de una variable, se especifican diferentes estilos y se muestra una leyenda.

**Ejemplo 19:** Dibujamos las funciones  $f(x) = \sin(x + \pi)$  y  $g(x) = \cos(x + \pi)$

```
>> fplot(@(x) sin(x+pi),'Linewidth',2); % grosor
>> hold on
>> fplot(@(x) cos(x+pi),'--r');% tipo y color
>> xlabel('x')
>> ylabel('y')
>> legend
```



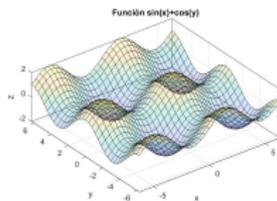
Si queremos colocar el origen de coordenadas en el (0,0) escribimos el siguiente código

```
>> ax = gca;
>> ax.XAxisLocation = 'origin';
>> ax.YAxisLocation = 'origin';
```

A continuación se muestra la gráfica de una función de dos variables, se especifican nombres de los ejes y título, incluye caja, misma escala en los ejes y transparencia.

**Ejemplo 20:** Dibujamos la función  $f(x, y) = \sin(x) + \cos(y)$

```
>> fsurf(@(x,y) sin(x)+cos(y), [-2*pi,2*pi,-2*pi,2*pi])
>> xlabel('x'); ylabel('y'); zlabel('z')
>> box on %caja
>> axis equal %escala
>> alpha(0.2)%transparencia
>> title('Funcion sin(x)+cos(y)')
```



Si queremos cambiar las marcas de los ejes

```
>> ax = gca;
>> ax.XTick = -2*pi:pi:2*pi;
>> ax.XTickLabel = {'-2\pi', '-\pi', '0', '\pi', '2\pi'};
>> ax.YTick = -2*pi:pi:2*pi;
>> ax.YTickLabel = {'-2\pi', '-\pi', '0', '\pi', '2\pi'};
```

El comando `plot3(X,Y,Z)` **dibuja puntos con coordenadas en el espacio 3-D**. Se deben especificar las coordenadas X, Y, y Z como vectores de la misma longitud.

**Ejemplo 21:**

```
>> A = [0 1 1 0 0 0 1 0 1 1 0 0 1 1 1 1 0 0;  
        0 0 1 1 0 0 0 0 0 1 1 0 0 0 1 1 1 1;  
        0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 1 1 0];  
>> plot3(A(1,:),A(2,:),A(3,:))
```

