

Diplomado : Herramientas de Programación para Ciencias e Ingeniería

Módulo: MATLAB (Clase 5)

Docente: Juan Sebastián Salcedo Gallo

Universidad Nacional de Colombia Sede Manizales

Contenido

- Gráficos 2D
- Gráficos 3D

Gráficos en MATLAB

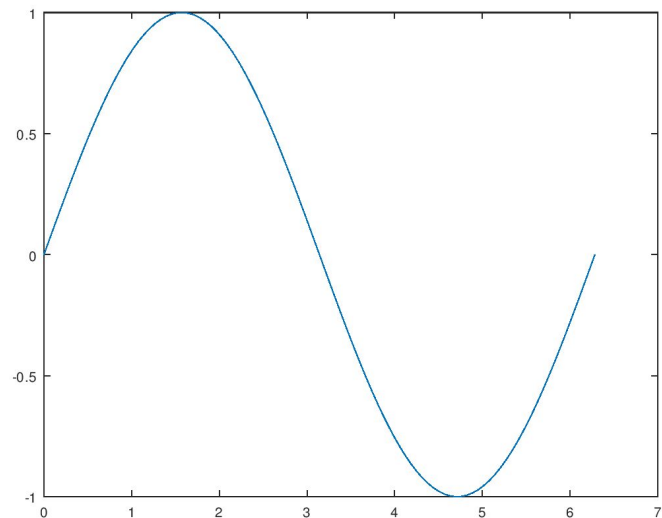
El sistema gráfico de MATLAB permite presentar gráficamente datos mediante comandos de alto nivel. Esto incluye la visualización en 2 y 3 dimensiones, el procesamiento digital de imágenes, elaboración de gráficos para presentaciones (histogramas, diagramas de barras, ...). Incluso el uso de efectos de animación, iluminación y movimientos.

Comando Plot de MATLAB

El comando ***plot*** sirve para crear gráficas de líneas bidimensionales.

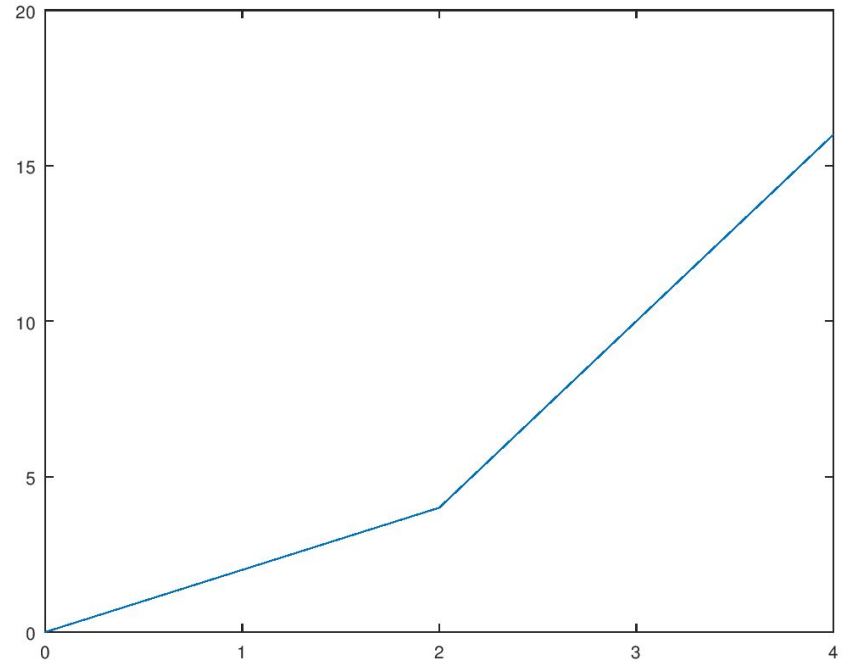
Ejemplo: grafique el valor de la función seno de 0 a 2π

```
>> x = linspace(0, 2*pi);  
>> y = sin(x);  
>> plot(x,y)  
>> |
```



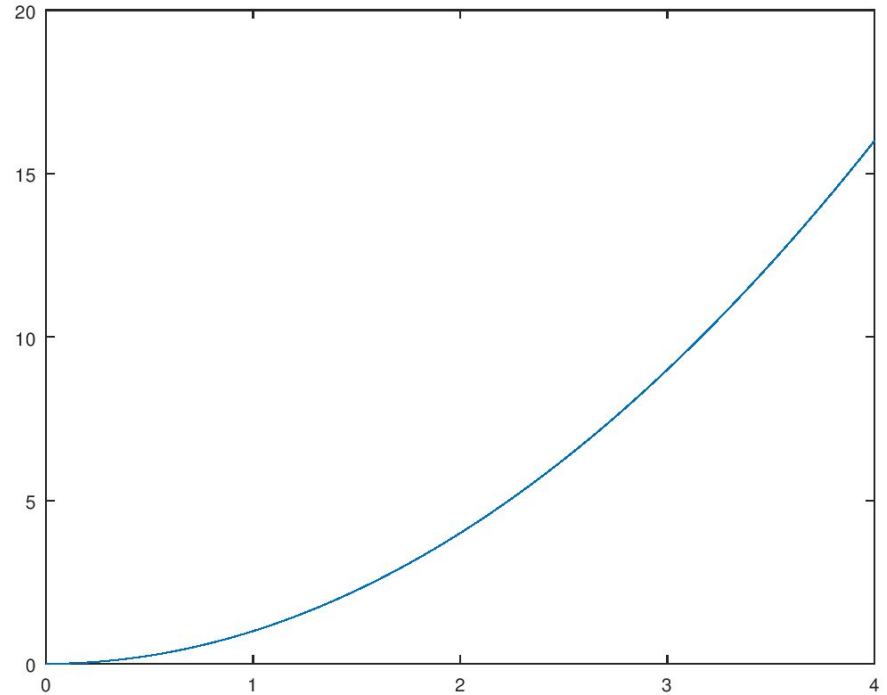
Ejemplo

```
>> x = [0.0, 2.0, 4.0];  
>> y = x.^2;  
>> plot(x, y)  
>> |
```



Ejemplo

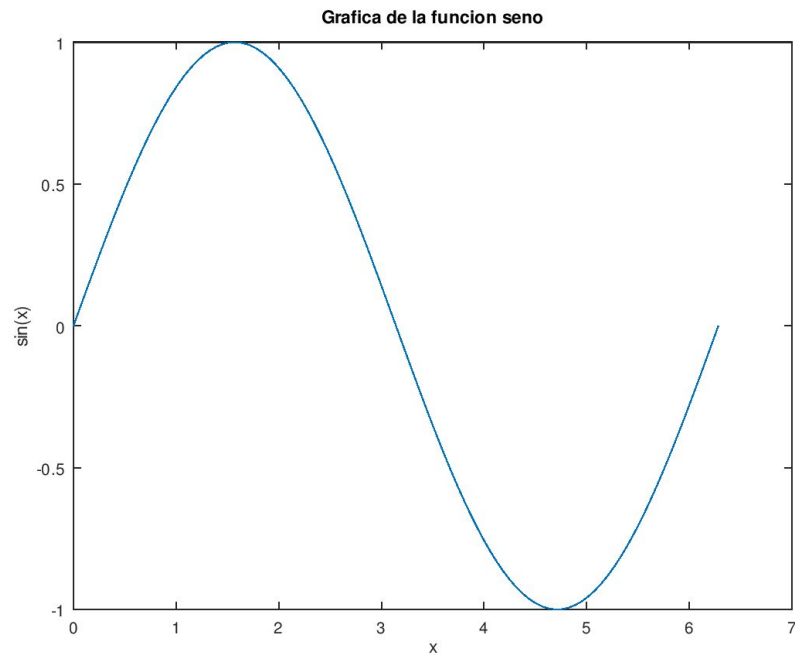
```
>> x = linspace(0,4);  
>> y = x.^2;  
>> plot(x,y)  
>> |
```



Ejes y Título

El programa le permite etiquetar los ejes y agregar un título.

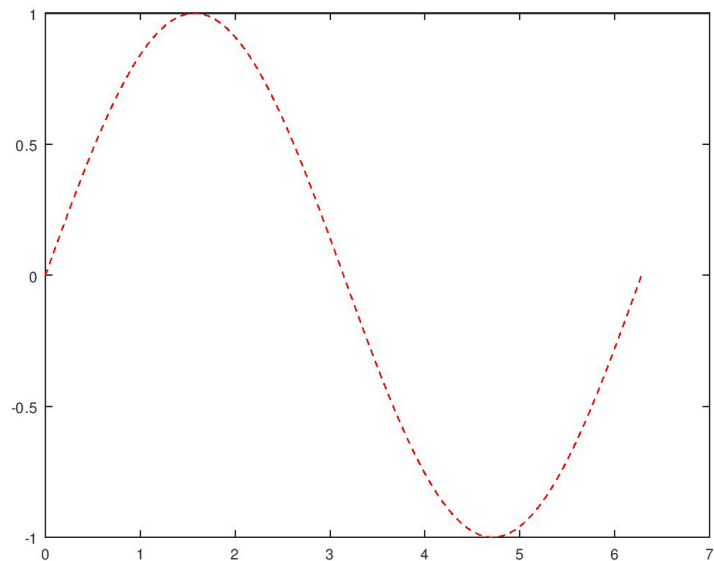
```
>> x = linspace(0, 2*pi);  
>> y = sin(x);  
>> plot(x,y);  
>> xlabel("x");  
>> ylabel("sin(x)");  
>> title("Grafica de la funcion seno");  
>> |
```



Estilo de línea

Si agrega un tercer argumento de entrada a la función plot, puede representar gráficamente las mismas variables mediante el uso de una línea discontinua roja.

```
>> plot(x,y, "r--");  
>> |
```



Marcadores


La cadena 'r--' es una especificación de línea. Cada especificación puede incluir caracteres para el marcador, el estilo y el color de la línea. Un marcador es un símbolo como +, o * que aparece en cada punto dibujado en la gráfica. Por ejemplo, 'g:*' especifica una línea verde punteada con marcadores *.

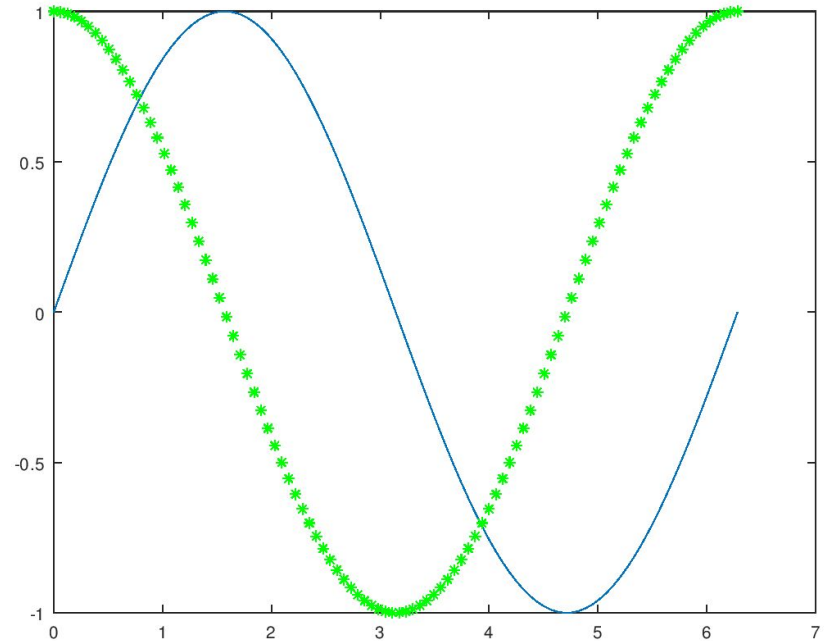
Observe que el nombre de los ejes y el título desaparecieron. MATLAB crea un lienzo nuevo, cada que se ejecuta el comando plot. A no ser que esté antepuesto por el comando **>> hold on;**

Marcadores

Valor	Descripción
'o'	Círculo
'+'	Signo más
'*'	Asterisco
'.'	Punto
'x'	Cruz
'square' o 's'	Cuadrado
'diamond' o 'd'	Rombo
'^'	Triángulo hacia arriba
'v'	Triángulo hacia abajo
'>'	Triángulo hacia la derecha
'<'	Triángulo hacia la izquierda
'pentagram' o 'p'	Estrella de cinco puntas (pentagrama)
'hexagram' o 'h'	Estrella de seis puntas (hexagrama)
'none'	Sin marcadores

Ejemplo

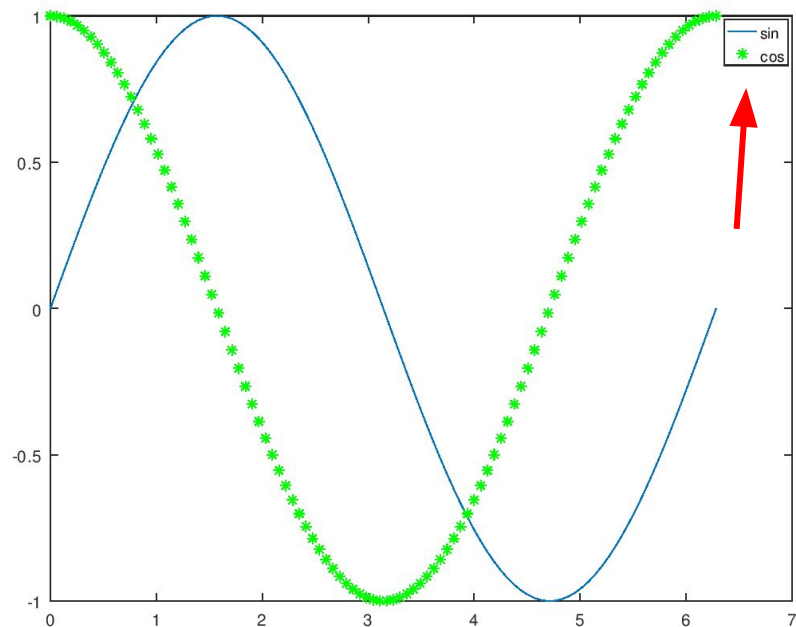
```
>> x = linspace(0, 2*pi);  
>> y = sin(x);  
>> z = cos(x);  
>> plot(x, y);  
>> hold on;   
>> plot(x, z, "g*");  
>> |
```



Etiquetas

Para añadir leyendas, use el comando `>> legend("lo que representa la gráfica 1", "lo que representa la gráfica 2")`

```
>> x = linspace(0, 2*pi);  
>> y = sin(x);  
>> z = cos(x);  
>> plot(x, y);  
>> hold on;  
>> plot(x, z, "g*");  
>> legend("sin", "cos");
```



Ejercicio

- Dibuje un círculo usando el comando **plot**.

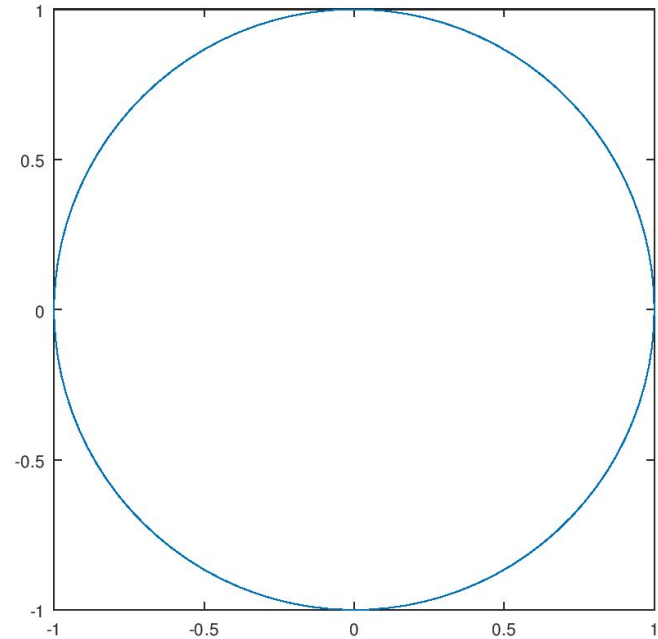
Herramienta útil: Para cuadrar el aspecto de los ejes, use el comando `>>daspect([1 1])`. Configura el aspecto de los ejes a igual proporción. Si por ejemplo quisiera que el eje “y” tuviese el doble del aspecto del eje x. Debería escribir `>>daspect([1 2])`.

Pista: Convierte las coordenadas cartesianas en coordenadas polares.

Solución

Command Window

```
>> t = linspace(0, 2*pi);  
>> x = cos(t);  
>> y = sin(t);  
>> plot(x, y);  
>> daspect([1 1]);  
>> |
```



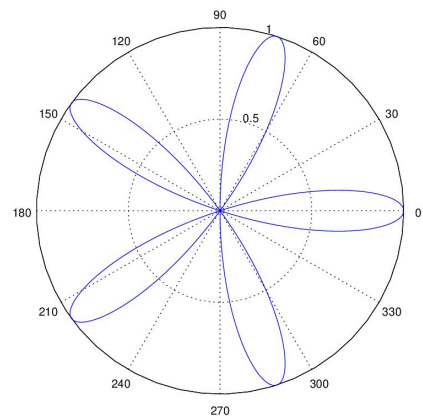
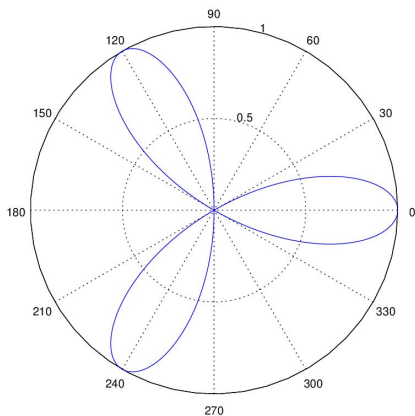
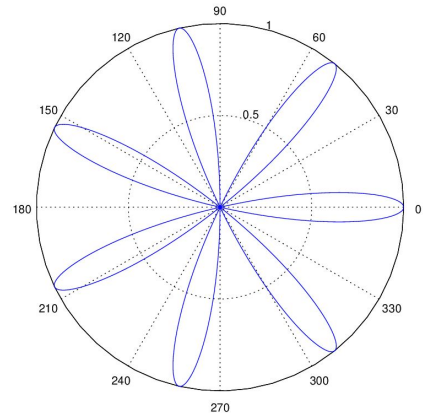
El comando polar

El comando `>> polar(theta, rho)` crea una gráfica 2-D en coordenadas polares, y toma como argumentos las variables `theta` y `rho(theta)`.

Ejemplo

Dibujar una Rosa Polar con 3, 5 y 7 pétalos.

Ver Página de Wikipedia.

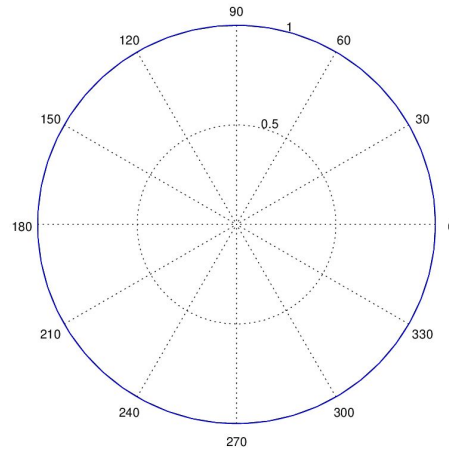


Ejercicio

Grafique un círculo usando la función `polar(theta, rho)`.

Solución

```
1 theta = linspace (0,2*pi,1000);  
2 rho = sqrt(sin(theta) .^ 2 + cos(theta) .^ 2);  
3 polar(theta, rho);
```



Comandos de interés relacionados con plot

Comando	¿Qué hace?
xlabel("texto")	etiqueta sobre el eje X de la gráfica actual:
ylabel("texto")	etiqueta sobre el eje Y de la gráfica actual
title("texto")	título en la cabecera de la gráfica actual
text(x, y, "texto")	texto en el lugar especificado por las coordenadas
gtext("texto")	texto, el lugar lo indicamos después con el ratón
grid	dibujar una rejilla

Comandos de interés relacionados con plot

Comando	¿Qué hace?
axis([xmin xmax ymin ymax])	fija valores máximo y mínimo de los ejes
axis equal	fija que la escala en los ejes sea igual:
axis square	fija que la gráfica sea un cuadrado
axis normal	desactiva axis equal y axis square
hold on	abre una ventana de gráfico
hold off	borra lo que hay en la ventana de gráfico
scatter(x,y)	Grafica los puntos, sin unirlos entre sí

Algunos comandos de presentación de datos

Comando	¿Qué hace?
area(x,y)	Colorea el área bajo la gráfica (integral)
bar	Diagrama de barras (verticales)
barh	Diagrama de barras (horizontales)
hist(x)	histograma
pie(x)	sectores
rose(x)	Histograma polar
stairs(x,y)	Gráfico de escalera
stem(x,y)	Secuencia de datos discretos

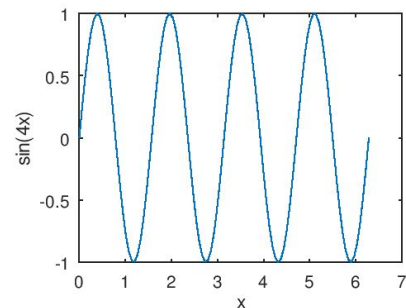
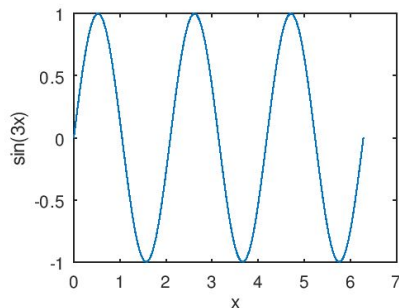
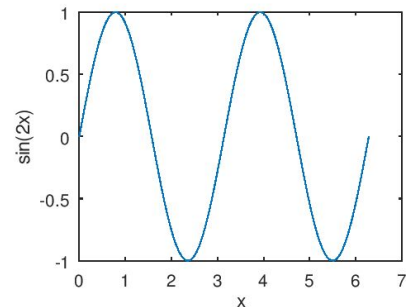
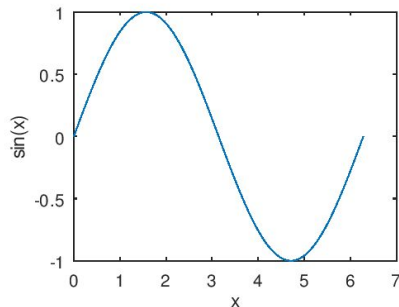
El comando subplot(m,n)

Una ventana gráfica se puede dividir en m particiones horizontales y en n particiones verticales, de modo que cada subventana tiene sus propios ejes. Para ello vamos a usar el comando subplot(m,n,p), donde p indica la ventana activa.

Ejemplo

Command Window

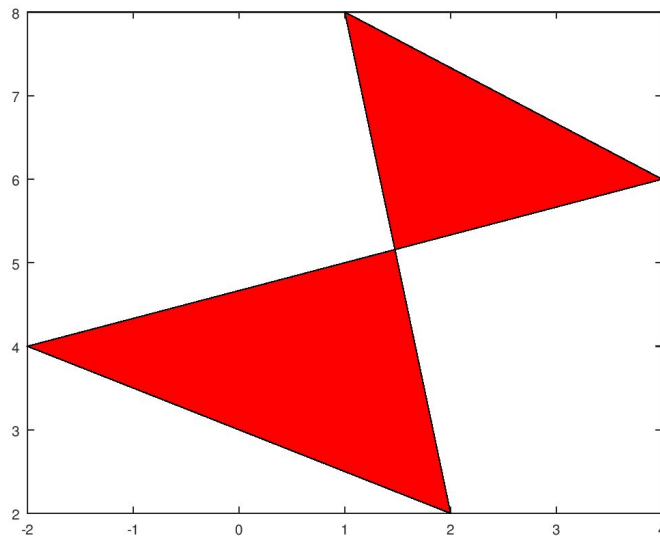
```
>> x = linspace(0.0, 2*pi);  
>> y1 = sin(x);  
>> y2 = sin(2*x);  
>> y3 = sin(3*x);  
>> subplot(2,2,1), plot(x,y1), xlabel("x"), ylabel("sin(x)");  
>> subplot(2,2,2), plot(x,y2), xlabel("x"), ylabel("sin(2x)");  
>> subplot(2,2,3), plot(x,y3), xlabel("x"), ylabel("sin(3x)");  
>> subplot(2,2,4), plot(x,y4), xlabel("x"), ylabel("sin(4x)");  
>> |
```



Gráficas 2D

Pueden generarse también polígonos, siempre y cuando sus puntos inicial y final coincidan, con el fin de crear una figura cerrada. En este caso podrá usarse el comando `>> fill(x, y, "color")`, con un tercer argumento que corresponde al color deseado.

```
>> x = [-2 4 1 2 -2];  
>> y = [4 6 8 2 4];  
>> fill(x,y,"r");  
>> |
```



Ejercicio

1. Resolver el sistema de ecuaciones lineales (hallar w, x, y, z):

$$\begin{cases} 3x - 10y + z = 1 \\ 4w + x - 2z = 3 \\ 2w - x + \frac{1}{3}y = 14 \\ 5w + 3z = 0 \end{cases} \quad (1)$$

Solución

```
>> A = [3 -10 1 0; 1 0 -2 4; -1 1/3 0 2; 0 0 3 5];  
>> B = [1; 3; 14; 0];  
>> inv_A = inv(A);  
>> X = inv_A * B  
X =  
  
-11.3615  
-3.8349  
-3.2640  
1.9584
```

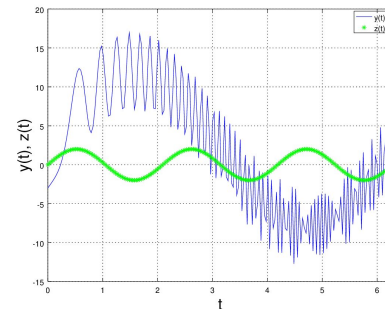
Ejercicios Gráficas 2D

2. Grafique las funciones:

$$y(t) = 10 \sin(t) - 5 \cos(10x^2) + 2 \quad (2)$$

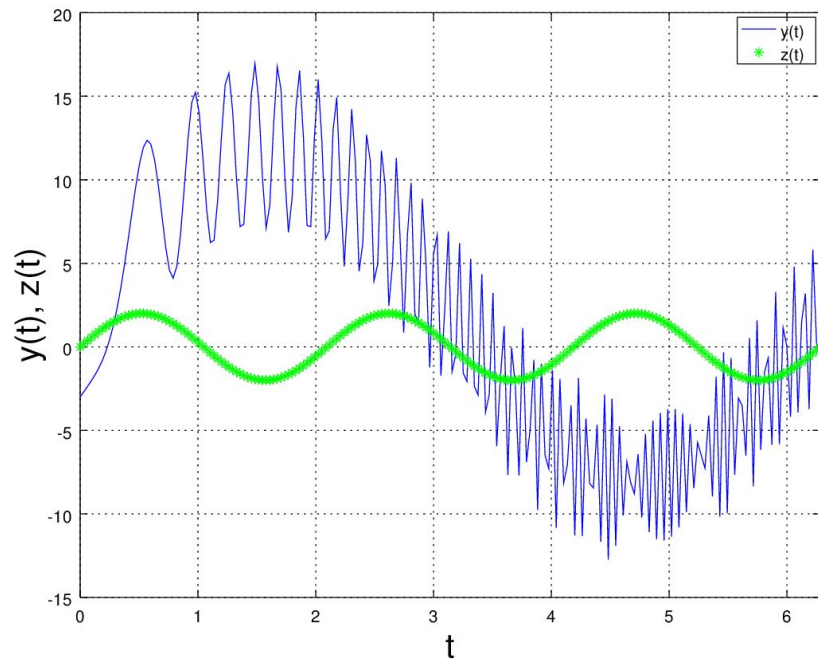
$$z(t) = 2 \sin(3x) \quad (3)$$

$t \in [0, 2\pi]$, con sus respectivos ejes, con rejilla, con colores y marcadores diferentes para cada función. Agregue una leyenda a cada curva.



Solución

```
1 t = linspace(0, 2*pi, 200);  
2 y = 10*sin(t) - 5*cos(10*t.^2) + 2;  
3 z = 2*sin(3*t);  
4 plot(t, y);  
5 hold on;  
6 plot(t, z, "g*");  
7 xlabel("t", "fontsize", 20);  
8 ylabel("y(t), z(t)", "fontsize", 20);  
9 grid on  
10 legend("y(t)", "z(t)");  
11 xlim([0 2*pi]);
```



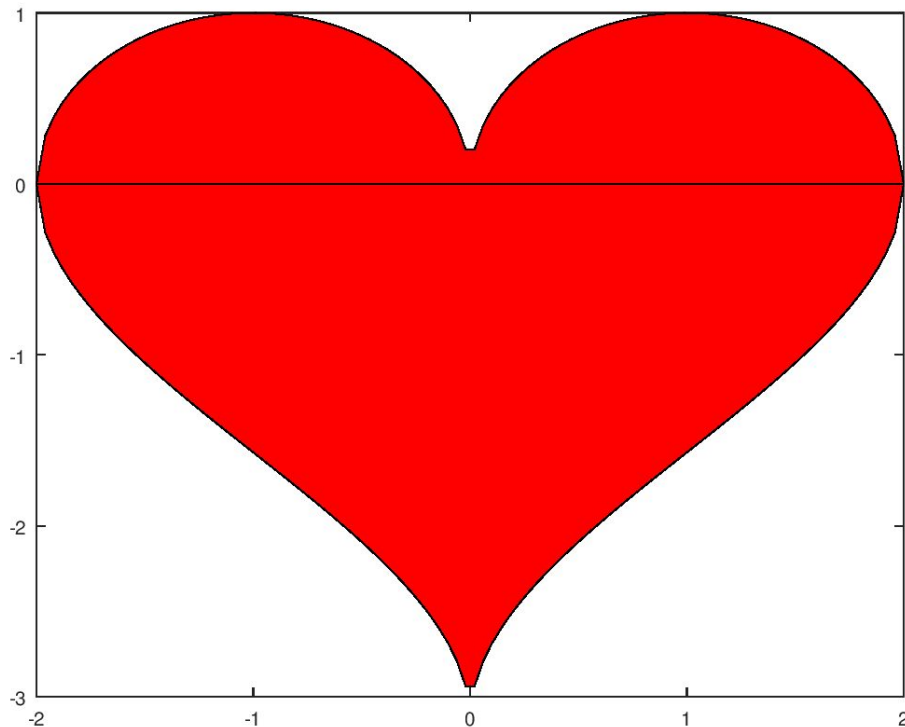
Ejercicios

Recree la siguiente imagen

$$-2 \leq x \leq 2$$

$$f(x) = \sqrt{1 - (|x| - 1)^2}$$

$$g(x) = \arccos(1 - |x|) - \pi$$



Solución

```
>> x = linspace(-2,2);  
>> f_ = sqrt(1 - (abs(x) - 1).^2);  
>> g_ = acos(1 - abs(x)) - pi;  
>> fill(x, f_, "r")  
>> hold on;  
>> fill(x, g_, "r")  
>> grid on  
>> |
```

Gráficas 3D

Se pueden crear gráficas en tres dimensiones, de forma análoga a como lo hicimos para las gráficas en dos dimensiones. La única diferencia es que ahora pasamos los argumentos en tripletes, y usamos el comando **>> plot3(X, Y, Z)** en lugar del comando `plot(X,Y)`.

Gráficas 3D

Normalmente, las gráficas tridimensionales representan a una función en términos de otras dos variables $z=z(x,y)$

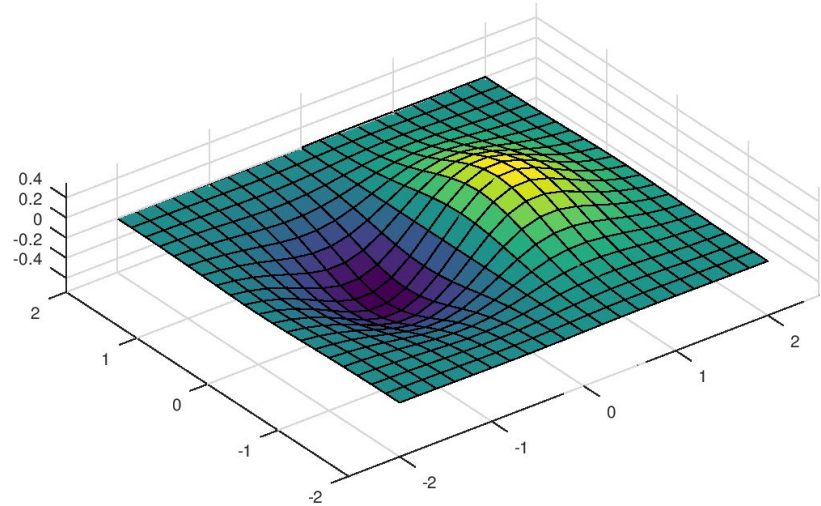
Para evaluar z , primero cree un conjunto de puntos (x,y) en el dominio de la función usando **meshgrid**

Y luego, cree una gráfica de la superficie, usando el comando **>>surf(X, Y, Z)**

Los ejes y el título se asignan de forma similar a como lo vimos anteriormente para gráficas bidimensionales.

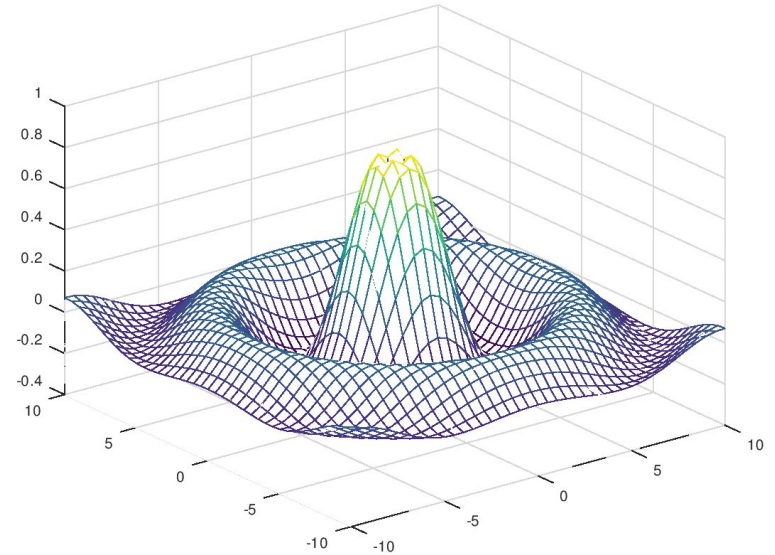
Ejemplo Gráfica 3D

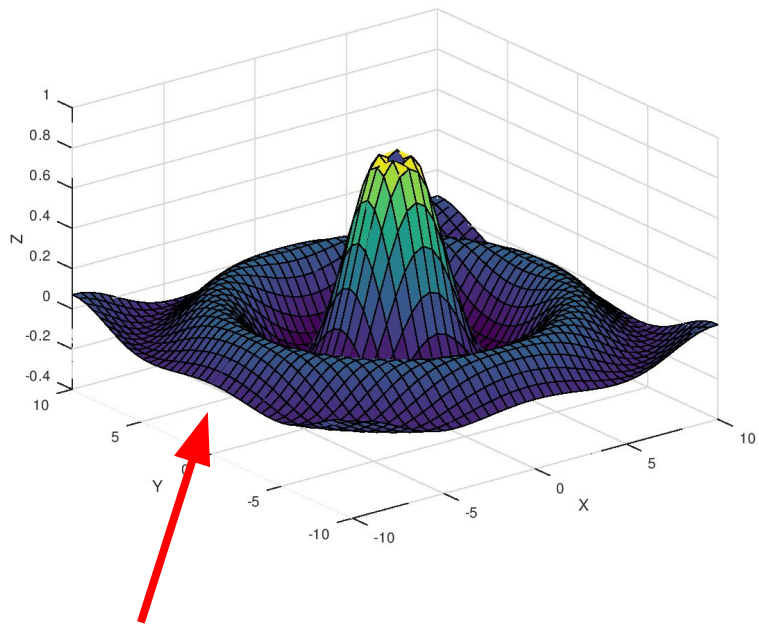
```
>> [X,Y] = meshgrid(-2:.2:2);  
>> Z = X .* exp(-X.^2 - Y.^2);  
>> surf(X,Y,Z)  
>> axis equal  
>> |
```



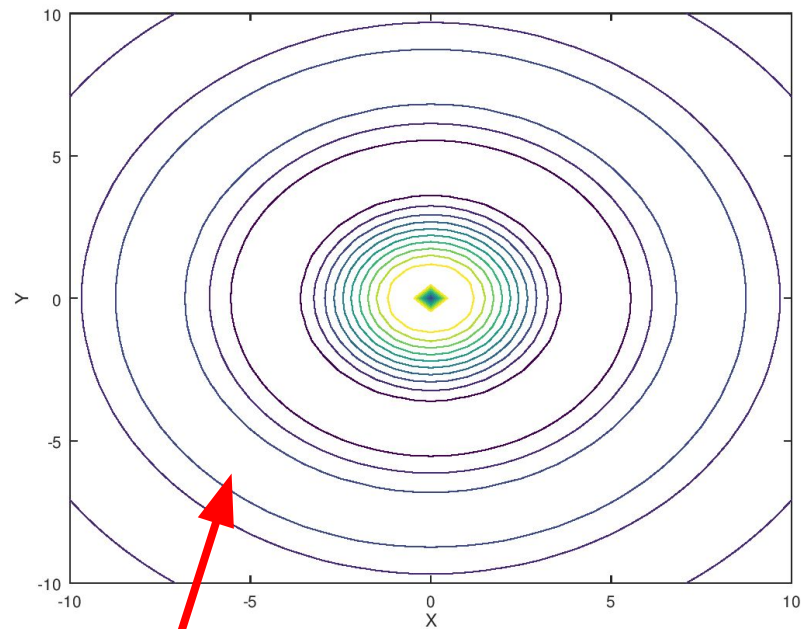
Ejemplo Gráfica 3D

```
>> X = -10:0.5:10;  
>> Y = -10:0.5:10;  
>> [X,Y] = meshgrid(X,Y);  
>> Z = sin(sqrt(X.^2 + Y.^2)) ./ sqrt(X.^2  
+ Y.^2 + 0.1);  
>> mesh(X,Y,Z)  
>> |
```

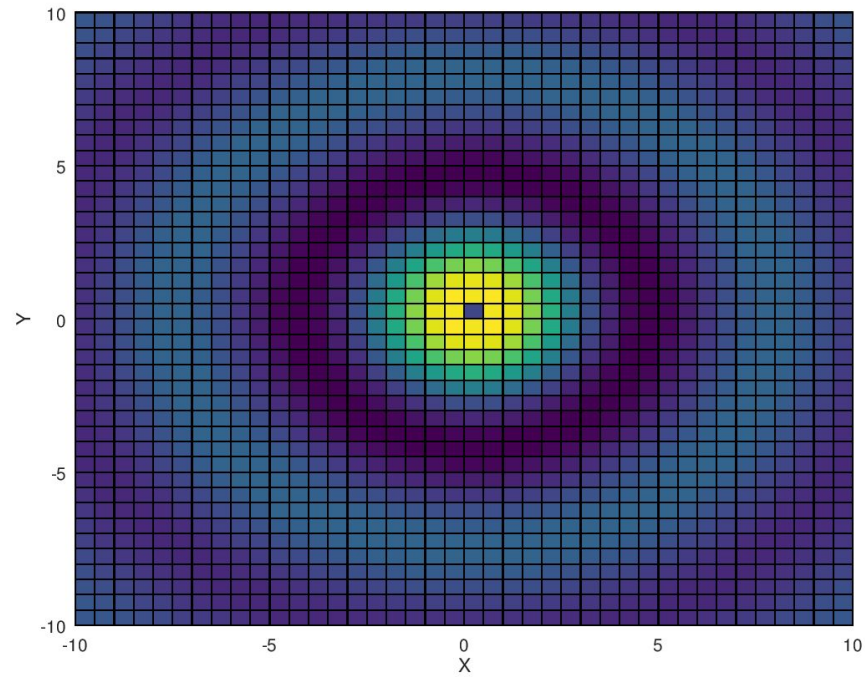




`surface(X,Y,Z)` Colorea la superficie.






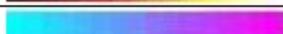
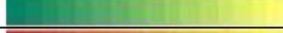


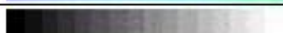
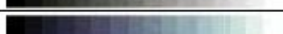
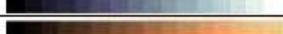
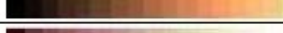
`contour(X,Y,Z)` Dibuja las líneas de contorno.



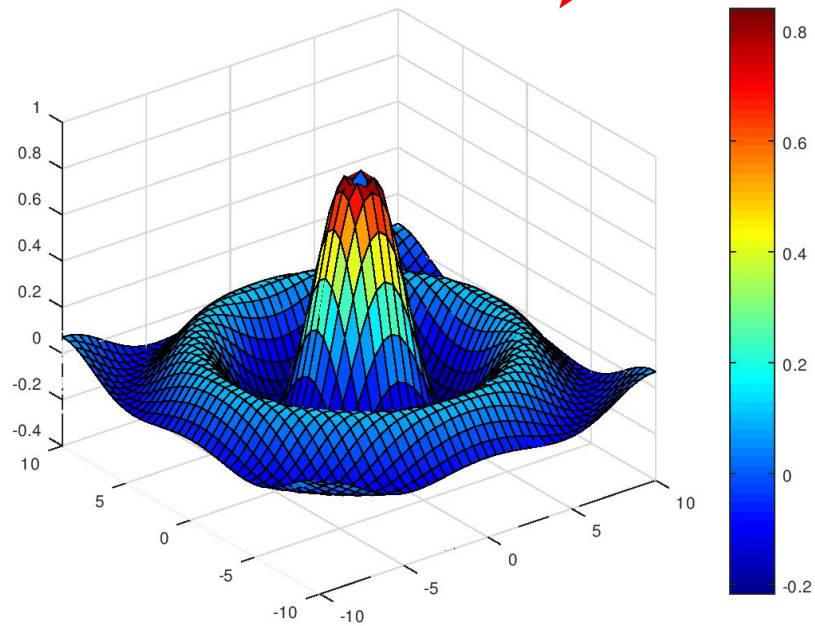
La función **pcolor(X,Y,Z)** transforma la altura a un conjunto de colores.

Comandos de interés: Visualización en 3D

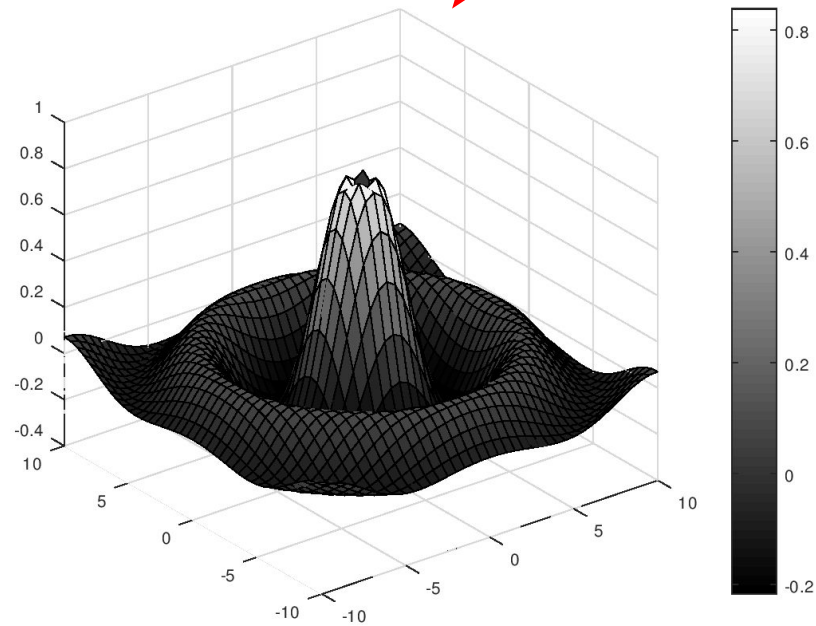
- **>>colorbar** añade la barra de color, es útil para comparar de forma intuitiva la magnitud de la función.
- **>>colormap("Función")** Puede seleccionarse la escala de colores de interés, algunas de ellas son:

Función	Colores
Jet	
HSV	
Hot	
Cool	
Spring	
Summer	
Autumn	
Winter	
Gray	
Bone	
Copper	
Pink	
Lines	

colormap("jet")
colorbar



colormap("gray")
colorbar



Colores de interés

Color	Nombre corto	Rojo/Verde/Azul
Negro	k	[0 0 0]
Blanco	w	[1 1 1]
Rojo	r	[1 0 0]
Verde	g	[0 1 0]
Azul	b	[0 0 1]
Amarillo	y	[1 1 0]
Magenta	m	[1 0 1]

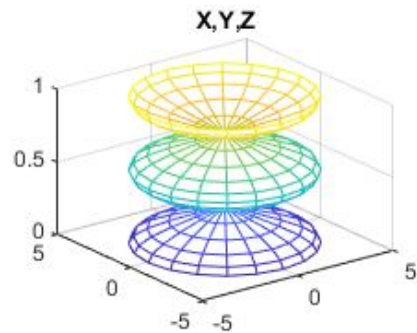
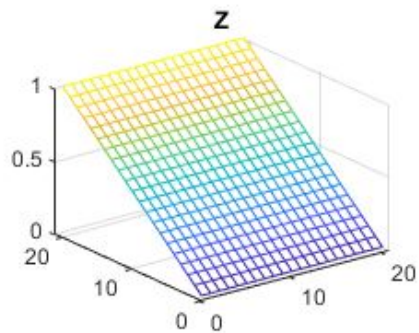
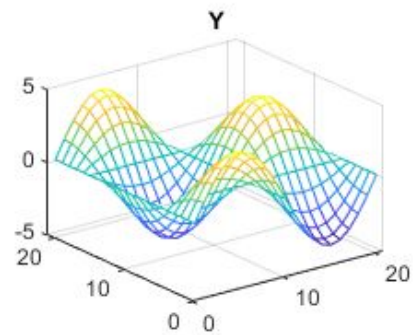
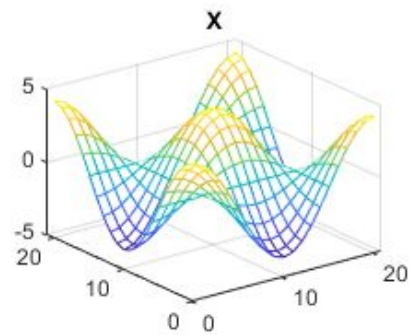
El comando subplot(m,n)

Tal y como se mostró en la sección de gráficas 2D, en gráficas 3D también puede dividirse el lienzo, para mostrar varias gráficas a la vez, cada una independiente, con sus propios labels y títulos.


```

t = 0:pi/10:2*pi;
[X,Y,Z] = cylinder(4*cos(t));
subplot(2,2,1); mesh(X); title('X');
subplot(2,2,2); mesh(Y); title('Y');
subplot(2,2,3); mesh(Z); title('Z');
subplot(2,2,4); mesh(X,Y,Z); title('X,Y,Z');

```



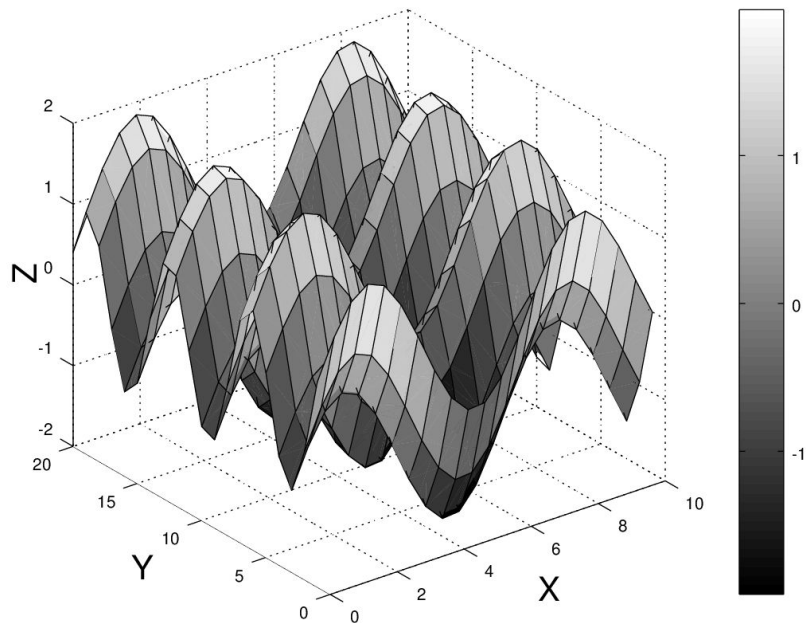
Ejercicios Gráficas en 3D

Grafique la siguiente función:

$$z(x, y) = \sin(x) + \cos(y)$$

Muéstrelo en una superficie, con el respectivo nombre a los ejes. Utilice una escala de grises. Además, incluya una barra de colores para visualizar la magnitud de la función de forma intuitiva.

Solución



```
1 [x, y] = meshgrid(0:0.5:10, 1:20);  
2 z = sin(x) + cos(y);  
3 surf(x,y,z);  
4 colormap("gray");  
5 colorbar  
6 xlabel("X", "fontsize", 20);  
7 ylabel("Y", "fontsize", 20);  
8 zlabel("Z", "fontsize", 20);  
9 axes equal  
10  
11
```

El comando isosurface

El comando isosurface, es capaz de extraer datos de un volumen y seleccionar en qué puntos la función se mantiene constante (Es una superficie que representa a los puntos que representen un valor constante).

El comando `>> [f,v] = isosurface(...)` retorna las “caras (f)” y los “vértices (v)” en arreglos separados.

Estos argumentos pueden pasarse a la función “patch(...)” que es un objeto “parche” en MATLAB.

Función patch(...)

Un objeto Graphics Patch está compuesto por uno o más polígonos que pueden o no estar conectados. Los remiendos son útiles para modelar objetos del mundo real tales como aviones o automóviles. Diferente a la función de superficie que son cuadrículas útiles para mostrar topografías planas (funciones de dos variables).

Syntax

```
patch(X,Y,C)
patch(X,Y,Z,C)
patch(FV)
patch(... 'PropertyName',PropertyValue...)
patch('PropertyName',PropertyValue...) PN/PV pairs only
handle = patch(...)
```

Syntax

```
fv = isosurface(X,Y,Z,V,isovalue)
fv = isosurface(V,isovalue)
fvc = isosurface(...,colors)
fv = isosurface(...,'noshare')
fv = isosurface(...,'verbose')
[f,v] = isosurface(...)
[f,v,c] = isosurface(...)
isosurface(...)
```

Ejemplo ilustrativo isosurface y patch

```
1 x_range = linspace(-2,2,300);
2 y_range = linspace(-2,2,300);
3 z_range = linspace(-2,2,300);
4 [X_space, Y_space, Z_space] = meshgrid( x_range, y_range, z_range );
5
6 f = (Y_space.^2 + 9/4*X_space.^2 + Z_space.^2-1).^3 ...
7     - Y_space.^2.*Z_space.^3 ...
8     - 9/80*X_space.^2.*Z_space.^3;
9
10
11 [faces,verts] = isosurface(X_space, Y_space, Z_space, f,.0);
12 patch('Vertices',verts,'Faces',faces, 'FaceColor', 'r', 'EdgeColor', 'none')
13 box on
14 view(49,23);
15 camlight
16 lighting gouraud
17
18
```

Video Informativo

<https://www.youtube.com/watch?v=eUzB0L0mSCI>

Can You Recover Sound From Images? - Veritasium