Computación 1 - 2022

Gráficos en 2D y 3D

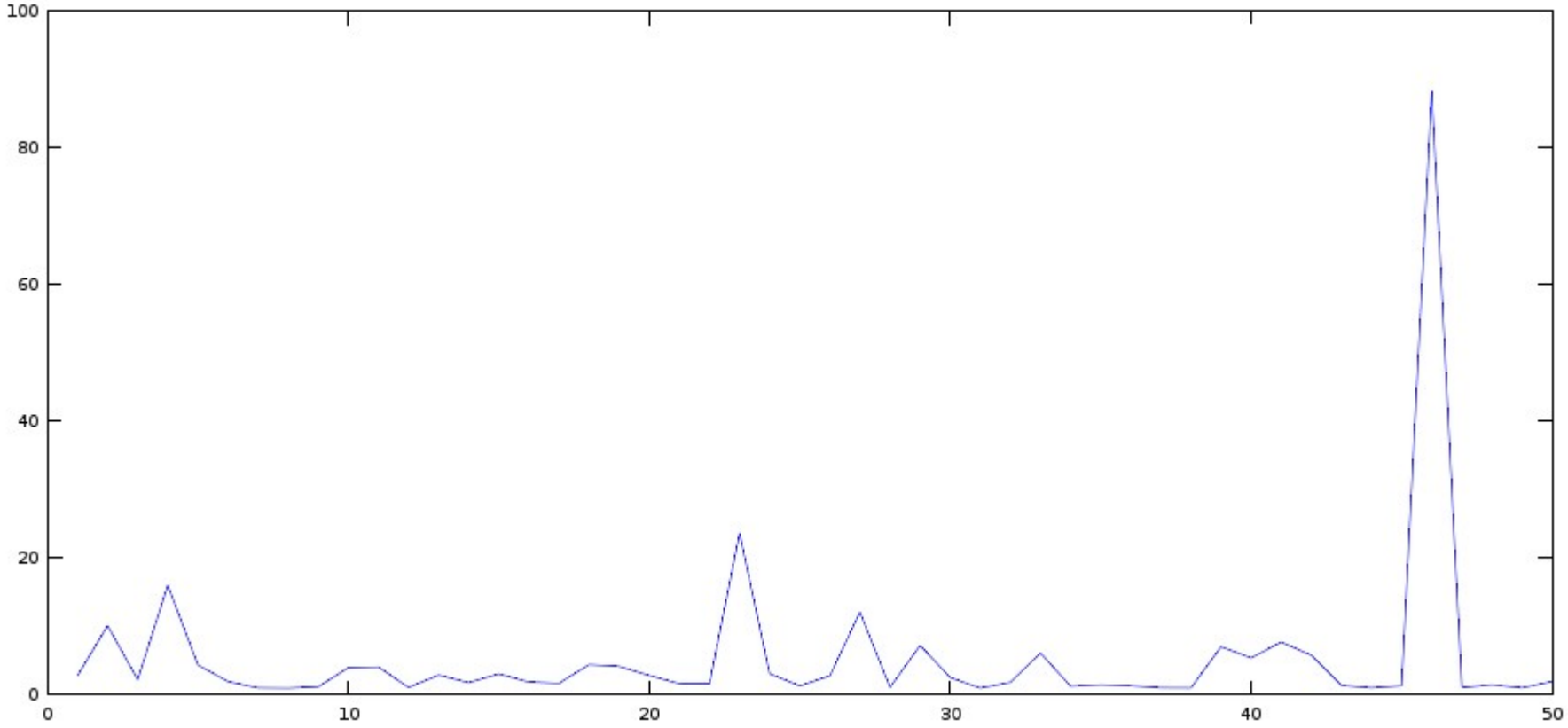
Necesidades

- Visualizar tendencias, patrones, frecuencias, trayectorias o cambios que son difíciles de encontrar en un conjunto de datos.
- ¡Una imagen vale más que mil palabras!
- Los gráficos ayudan a la toma de decisiones.

Ejemplo: **Encontrar el máximo de un conjunto**

```
Columns 1 through 10:
   2.8175   10.1326    2.2346   16.0404    4.4026    1.9738    1.0462    1.0091    1.1983    3.9982

Columns 11 through 20:
   4.0988    1.1173    2.9033    1.8290    3.0760    1.9099    1.7212    4.4390    4.1799    2.8604

Columns 21 through 30:
   1.6685    1.6588   23.6487    3.1122    1.3537    2.8040   12.1043    1.1234    7.2549    2.5510

Columns 31 through 40:
   1.0208    1.8573    6.1357    1.2936    1.5113    1.3588    1.0568    1.0353    7.0616    5.4359

Columns 41 through 50:
   7.7418    5.8158    1.4088    1.0493    1.3506   88.2787    1.0783    1.5144    1.0504    1.9798
```

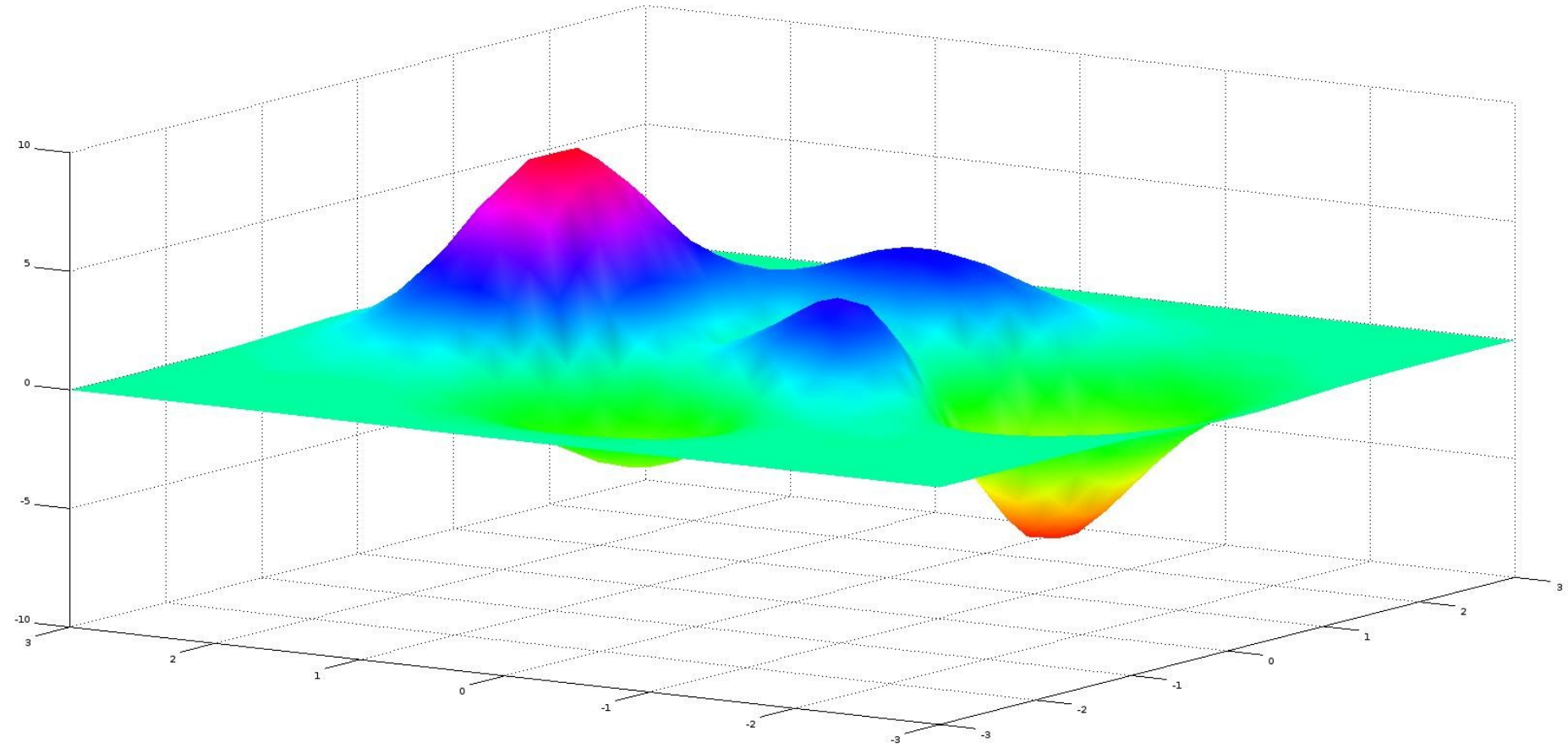
Ejemplo: **Encontrar el máximo de un conjunto**



Ejemplo: **comprender los valores de una matriz**

6.6713e-001 1.8839e+000 4.7741e+000 1.0777e+001 2.1394e+001 3.6471e+001 5.0583e+001
1.3615e+000 3.8340e+000 9.6617e+000 2.1597e+001 4.2155e+001 6.9640e+001 9.0076e+001
2.5254e+000 7.1228e+000 1.7932e+001 3.9905e+001 7.7097e+001 1.2454e+002 1.5203e+002
4.2186e+000 1.2008e+001 3.0443e+001 6.8078e+001 1.3182e+002 2.1240e+002 2.5513e+002
6.2063e+000 1.8068e+001 4.6737e+001 1.0655e+002 2.1060e+002 3.4844e+002 4.3922e+002
7.5942e+000 2.3267e+001 6.2895e+001 1.4951e+002 3.0917e+002 5.4240e+002 7.5890e+002
6.3417e+000 2.2602e+001 6.8123e+001 1.7707e+002 3.9821e+002 7.6741e+002 1.2302e+003
-1.1225e+000 6.8887e+000 4.1208e+001 1.4570e+002 3.9997e+002 9.0680e+002 1.7199e+003
-1.9702e+001 -3.6995e+001 -4.9776e+001 -1.4926e+001 1.7475e+002 7.1218e+002 1.8363e+003
-5.4300e+001 -1.2273e+002 -2.3984e+002 -3.8692e+002 -4.5374e+002 -1.6297e+002 9.5564e+002
-1.0726e+002 -2.5761e+002 -5.4981e+002 -1.0249e+003 -1.6170e+003 -2.0097e+003 -1.5137e+003
-1.7555e+002 -4.3457e+002 -9.6566e+002 -1.9065e+003 -3.2925e+003 -4.8399e+003 -5.7083e+003
-2.4947e+002 -6.2846e+002 -1.4280e+003 -2.9054e+003 -5.2395e+003 -8.2460e+003 -1.1023e+004
-3.1455e+002 -8.0081e+002 -1.8439e+003 -3.8169e+003 -7.0483e+003 -1.1486e+004 -1.6241e+004
-3.5622e+002 -9.1279e+002 -2.1186e+003 -4.4305e+003 -8.2937e+003 -1.3777e+004 -2.0056e+004
-3.6521e+002 -9.3958e+002 -2.1915e+003 -4.6112e+003 -8.7022e+003 -1.4618e+004 -2.1628e+004
-3.4066e+002 -8.7864e+002 -2.0556e+003 -4.3417e+003 -8.2338e+003 -1.3923e+004 -2.0795e+004
-2.8977e+002 -7.4835e+002 -1.7535e+003 -3.7105e+003 -7.0529e+003 -1.1961e+004 -1.7935e+004
-2.2450e+002 -5.7955e+002 -1.3569e+003 -2.8679e+003 -5.4404e+003 -9.1957e+003 -1.3709e+004
-1.5754e+002 -4.0507e+002 -9.4320e+002 -1.9778e+003 -3.7084e+003 -6.1557e+003 -8.9070e+003
-9.8907e+001 -2.5140e+002 -5.7608e+002 -1.1802e+003 -2.1360e+003 -3.3464e+003 -4.3581e+003
-5.4262e+001 -1.3408e+002 -2.9484e+002 -5.6663e+002 -9.1945e+002 -1.1565e+003 -7.7683e+002
-2.4739e+001 -5.6841e+001 -1.1076e+002 -1.6815e+002 -1.3785e+002 2.2964e+002 1.4433e+003
-8.1050e+000 -1.4173e+001 -1.1749e+001 3.8443e+001 2.4657e+002 8.5999e+002 2.3362e+003
-5.1250e-001 4.1789e+000 2.7246e+001 1.0911e+002 3.4831e+002 9.4937e+002 2.2727e+003
1.8783e+000 8.7246e+000 3.2725e+001 1.0544e+002 2.9951e+002 7.6005e+002 1.7364e+003
1.9284e+000 7.3675e+000 2.4829e+001 7.4832e+001 2.0326e+002 4.9992e+002 1.1169e+003
1.2963e+000 4.6465e+000 1.5003e+001 4.3870e+001 1.1657e+002 2.8208e+002 6.2266e+002
7.0011e-001 2.4357e+000 7.6959e+000 2.2143e+001 5.8120e+001 1.3934e+002 3.0541e+002
3.2235e-001 1.1033e+000 3.4431e+000 9.8129e+000 2.5567e+001 6.0947e+001 1.3300e+002

Ejemplo: **comprender los valores de una matriz**



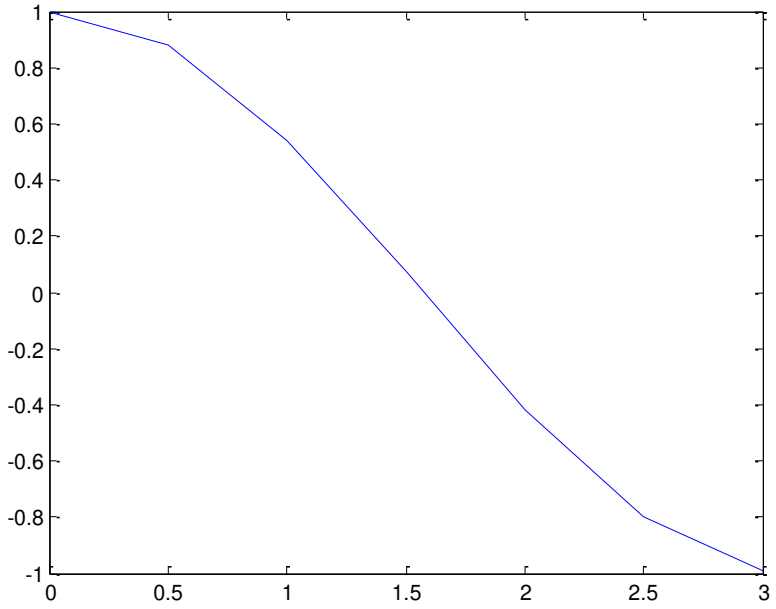
Comentarios

- Los gráficos representados en una computadora son discretos.
- A partir de una curva (matemática, materialmente inexistente), se toman puntos representantes y un programa se encarga de realizar una **interpolación**.
- El resultado de la **interpolación** se asemeja en mayor o en menor medida a la curva original.
- Existen varias formas de realizar **interpolaciones**.
- Para representar curvas, en Octave/Matlab utilizamos una interpolación lineal para aproximar el gráfico a la curva original.

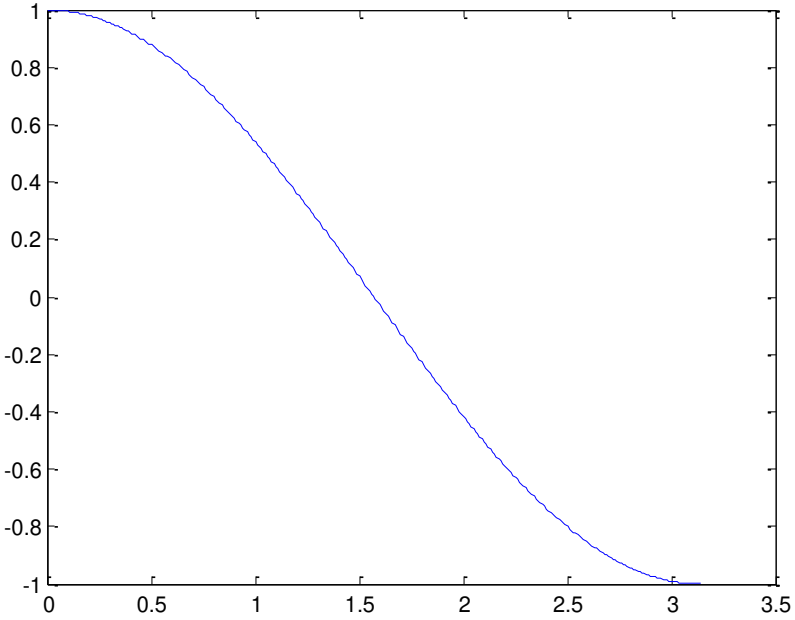
Comentario

- Cuantos más puntos, mejor será la aproximación
- Ejemplos:

f(x) = cos(x) para el conjunto x = [0:0.5:pi]



f(x) = cos(x) para el conjunto x = [0:0.01:pi]



Herramientas

- Operaciones para dibujar gráficos bidimensionales:

  `plot`

  Traza líneas.

  `semilogx, semilogy.`

  Traza líneas, donde un eje tiene escala logarítmica

  `loglog`

  Traza líneas, donde ambos ejes tienen escala logarítmica

  `contour`

  Traza isolíneas, curvas de nivel.

  `quiver`

  Despliega vectores de velocidad como flechas.

Herramientas

- Operaciones para controlar la ventana de dibujo:

  `title`

  Agrega un título.

  `legend`

  Agrega una leyenda.

  `xlabel`

  Etiqueta el eje x.

  `ylabel`

  Etiqueta el eje y.

  `grid`

  Activar o descativar la grilla.

  `hold`

  Permite determinar si el gráfico se descartará ante un nuevo trazo.

  `subplot`

  Divide la ventana de dibujo en celdas.

Herramientas

- Operaciones para cambiar la escala de los ejes:

  `axis`

  Cambia la escala de los ejes.

  axis([xmin xmax ymin ymax])

  axis('square')

  `get`

  Permite obtener el valor de una propiedad gráfica.

  `set`

  Permite definir el valor de una propiedad gráfica.

Traza de líneas usando `plot`

- Sintaxis:

```
plot(Y)
plot(X1, Y1, ...)
plot(X1, Y2, tipo_de_trazo)
```

Traza de líneas usando `plot`

- **`plot(Y)`**

  Traza las columnas de Y contra sus índices.

- **`plot(X1, Y1, ...)`**

  Traza las columnas de Yi contra las columnas de Xi.

- **`plot(X1, Y2,`** tipo_de_trazo**`)`**

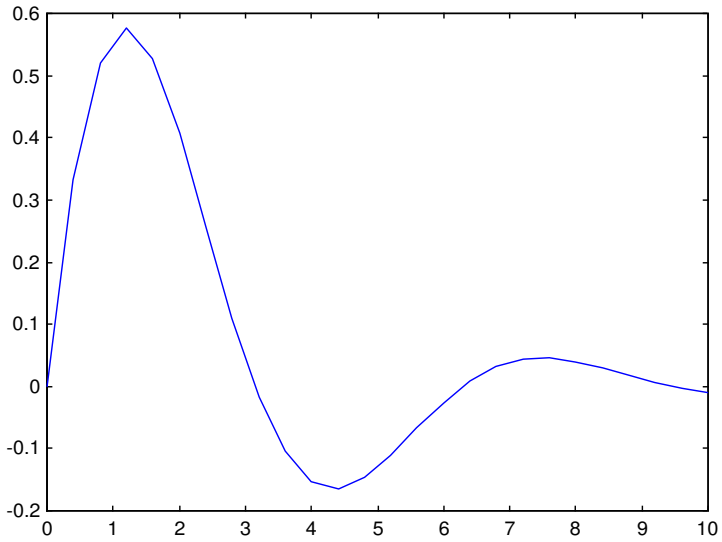
  Ítem anterior, pero permite definir el tipo de línea, la forma de los puntos y el color del trazo.

Traza de líneas usando `plot`

- Ejemplo:

$$f(x) = \sin(x)e^{-4x}$$

```
x = ( 0:0.4:10 );
y = sin(x) .* exp(-4 .* x);

plot(x,y)
```

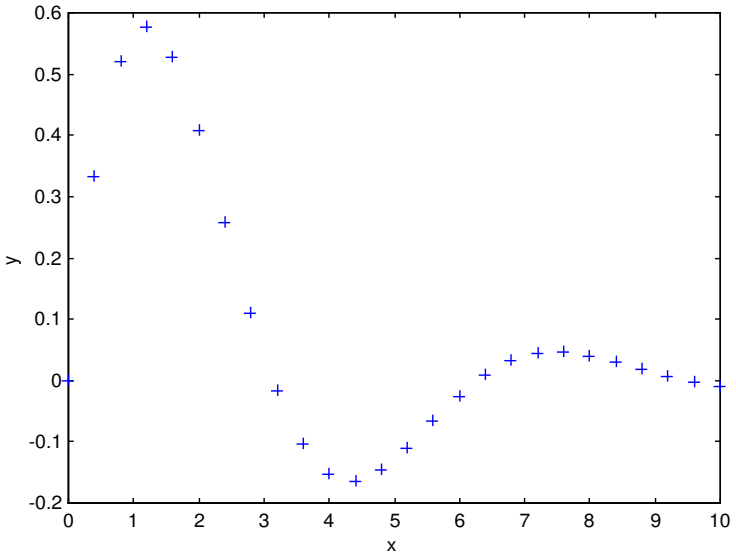


Traza de líneas usando `plot`

- Ejemplo:

$$f(x) = \sin(x)e^{-4x}$$

```
x = ( 0:0.4:10 );
y = sin(x) .* exp(-4 .* x);

plot(x,y,'+')
```



Traza de líneas usando `plot`

- Tipos de trazo: forma del punto

Especificador	Forma de punto
+	Signo de más
o	Círculo
*	Asterisco
.	Punto
x	Cruz
'square' o s	Cuadrado
'diamond' o d	Diamante
^	Triángulo apuntando hacia arriba
v	Triángulo apuntando hacia abajo
>	Triángulo apuntando hacia la derecha
<	Triángulo apuntando hacia la izquierda
'pentagram' o p	Estrella de cinco puntos
'hexagram' o h	Estrella de seis puntos

Traza de líneas usando `plot`

- Tipos de trazo: tipo de línea

Especificador	Tipo de línea
-	Continua
--	Guionada
:	Punteada
-.	Guionada y punteada

Traza de líneas usando `plot`

- Tipos de trazo: color

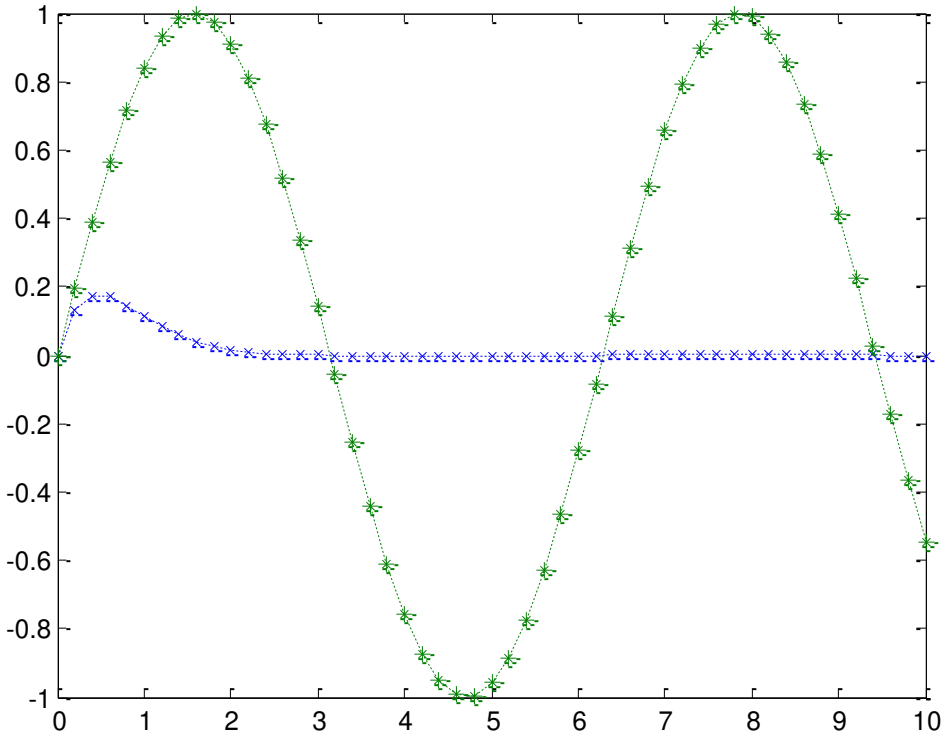
Especificador	Color
r	Rojo
g	Verde
b	Azul
c	Cian
m	Magenta
y	Amarillo
k	Negro
w	Blanco

Traza de líneas usando `plot`

- Ejemplo: $f(x) = \sin(x)e^{-4x}$

$$g(x) = \sin(x)$$

```
x = ( 0:0.4:10 );
f = sin(x) .* exp(-4 .* x);
g = sin(x);

plot(x,f,':x',x,g,'--*')
```

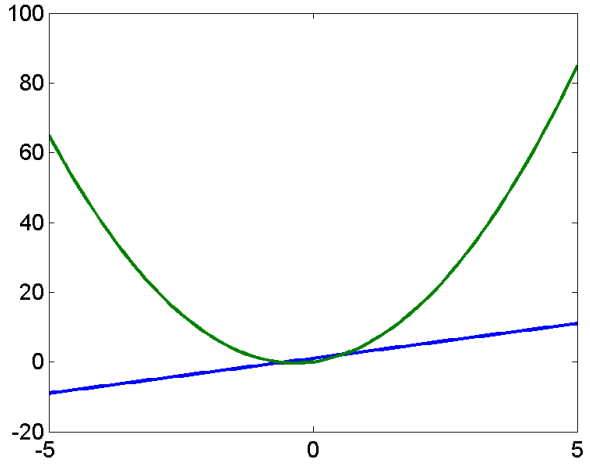


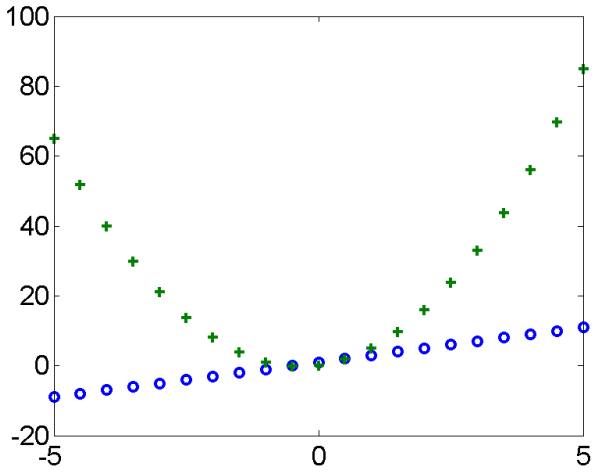
Traza de líneas usando `plot`

- Podemos mezclar marcas de puntos y tipos de línea en cualquier combinación.
- Esto es importante cuando el medio de visualización no permite colores.

Traza de líneas usando `plot`

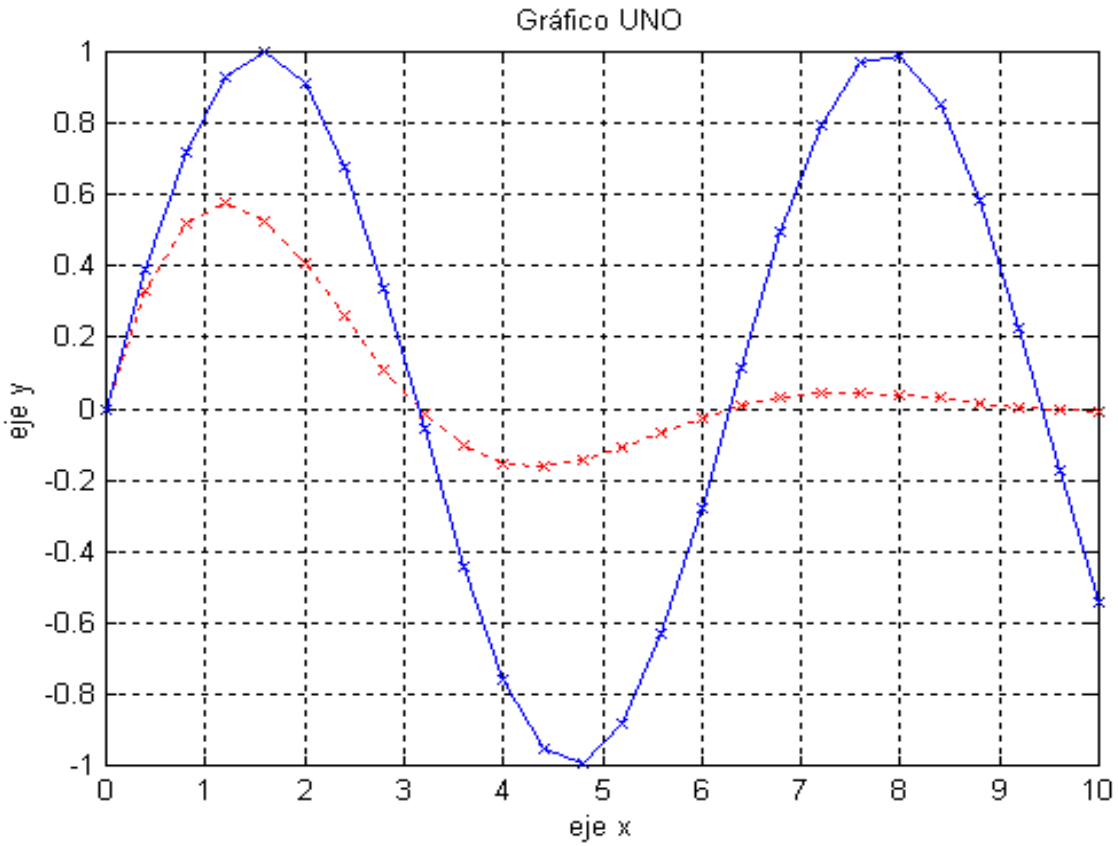
- Gráficas simultáneas.
  - Utilizando el formato por defecto:
    - **`plot(x1, y1, x2, y2)`**
  - Aplicando un formato propio:
    - **`plot(x1, y1, 'o', x2, y2, '+')`**
  - Usando hold on:
    - **`plot(x1, y1, 'o')`**
    - **`hold on`**
    - **`plot(x2, y2, '+')`**





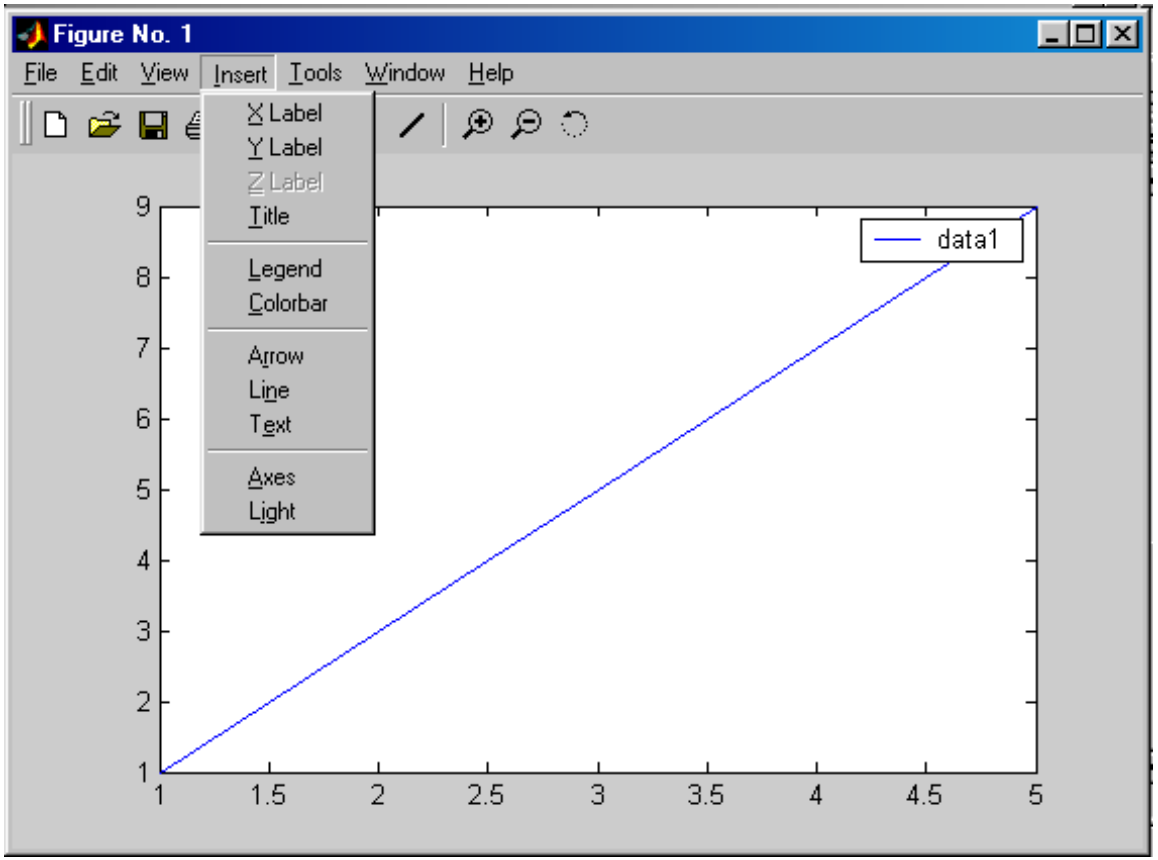
Grilla, etiqueta, título

```
plot ( x, y, ':xr')
grid on
xlabel('eje x');
ylabel('eje y');
hold on
plot ( x, y1, '-xb')
title('Gráfico UNO')
```



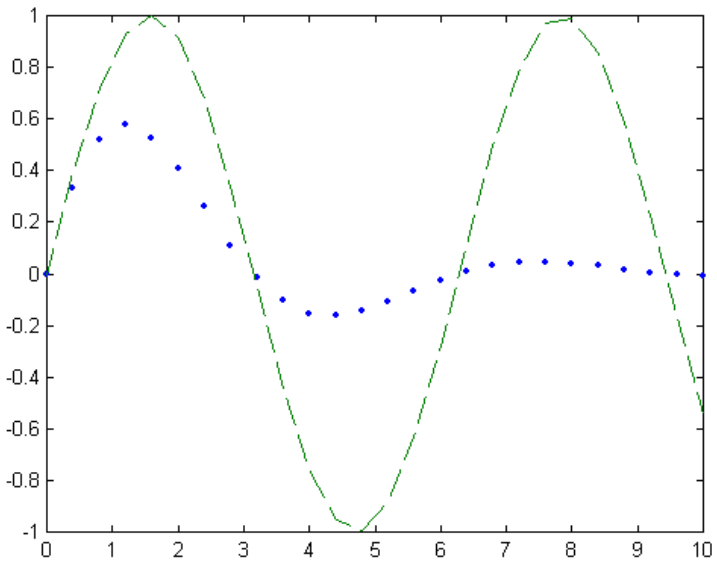
Grilla, etiqueta, título

```
plot ( x, y, '-b')
```

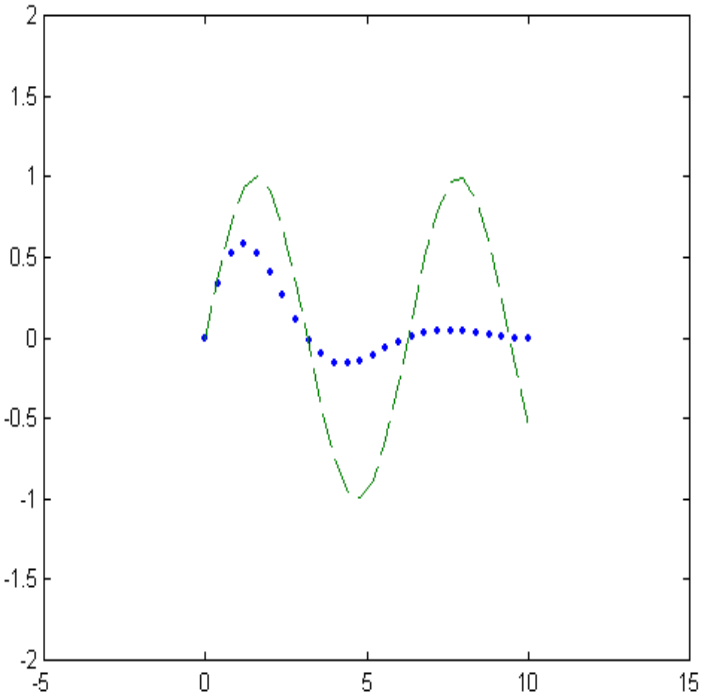


Escala de los ejes

- Escala seleccionada por Octave/Matlab



- Escala con `axis([-5 15 -2 2])`



Escala de los ejes

- Se puede hacer un forzar que la escala en alguno de los ejes (o ambos) sea logarítmica.
- Para ello, se utilizan los comandos `loglog`, `semilogx` o `semilogy`.
- Su funcionamiento es similar a `plot`, con la diferencia de que utilizan una escala logarítmica para realizar el trazo.

División de la ventana con `subplot`

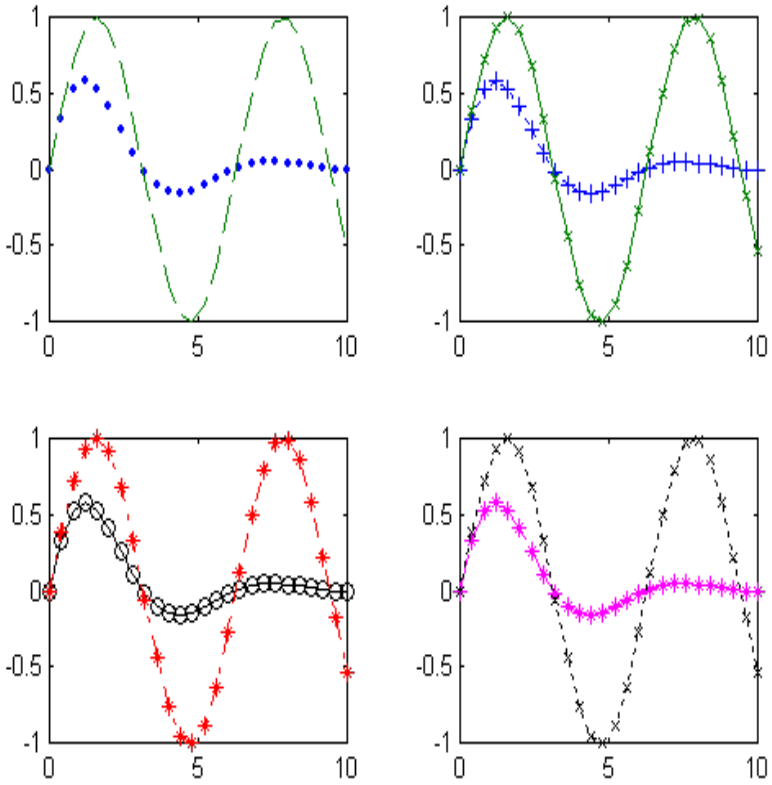
- El comando subplot permite dividir la ventana de dibujo en m*n celdas.
- A cada celda se le asigna un número: de izquierda a derecha y de arriba hacia abajo.
- El número permite definir sobre qué celda se trabajará.
- Sintaxis:

  `subplot(m,n,`número`)`

  Divide la ventana en m filas por n columnas, especificando qué número de celda se utilizará para dibujar el gráfico.

División de la ventana con `subplot`

```
subplot( 2, 2, 1)
plot ( x, y, '.', x, y1, '--')
subplot( 2, 2, 2)
plot ( x, y, '+:', x, y1, 'x-')
subplot( 2, 2, 3)
plot ( x, y, 'o-k', x, y1, '*-.r')
subplot( 2, 2, 4)
plot ( x, y, '*--m', x, y1, 'x:k')
```

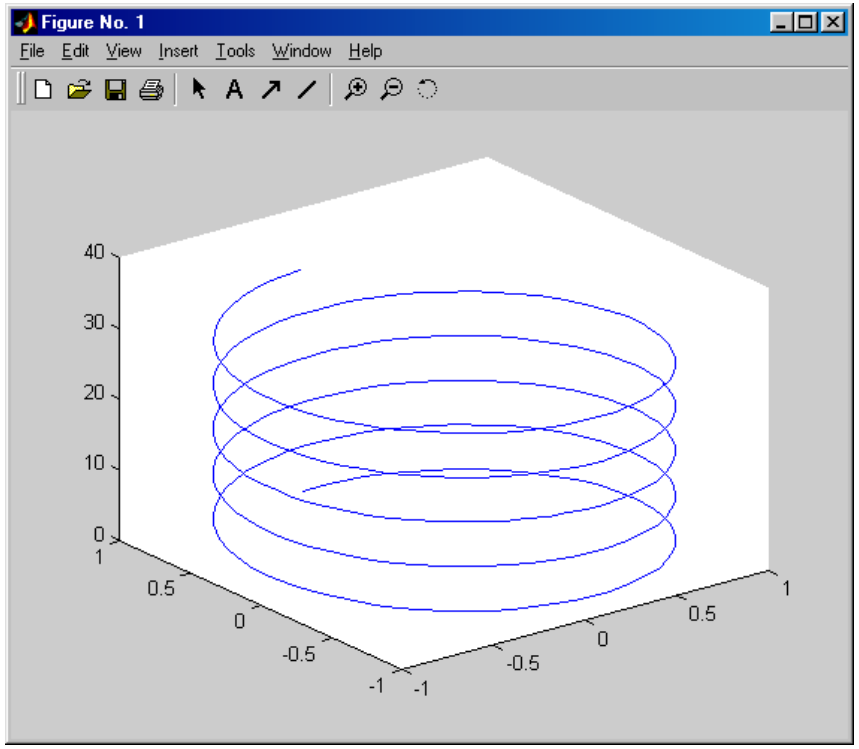


Otras opciones de gráficos

- Gráficas 3D
- Mallas (mesh)
- Superficies con texturas
- Efectos de iluminación, etc.

Ejemplo de plot 3D

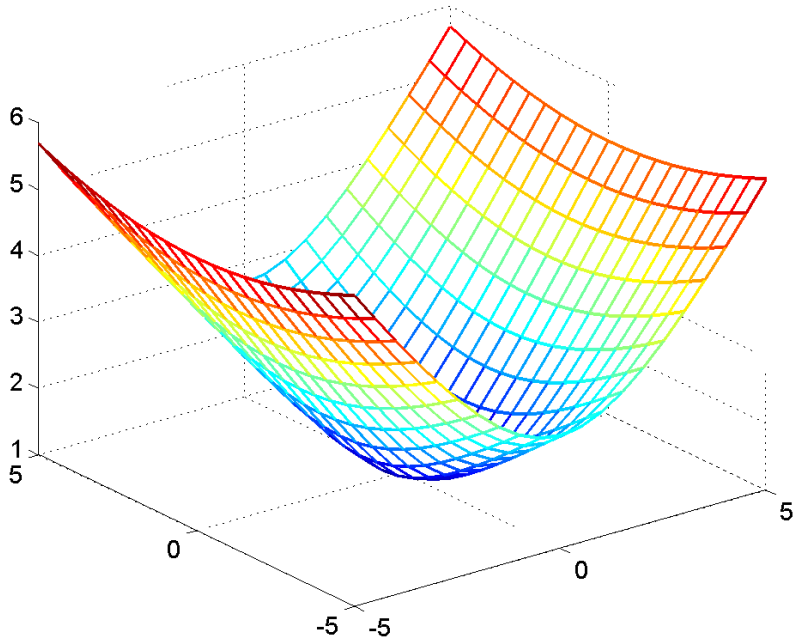
```
t = 0:.1:10*pi;
plot3 (sin(t), cos(t), t)
```



Ejemplo de mesh

```
x = [-5:.5:5];
y = x;
[X,Y] = meshgrid(x,y);
Z = sqrt(1+0.25*Y.^2+X.^2);
```

`mesh(X,Y,Z)`



Funcionamiento de meshgrid

- Transforma el dominio especificado por los vectores x e y en matrices X e Y.
- X e Y pueden ser utilizadas para evaluar funciones de dos variables y gráficas tridimensionales (usando mesh o surface).
- Las filas de la matriz X son copias del vector x.
- Las columnas de la matriz Y son copias del vector y.
- Ejemplo:

```
[X, Y] = meshgrid(1:3, 10:14)
```

```
X =

     1     2     3
     1     2     3
     1     2     3
     1     2     3
     1     2     3
```

```
Y =

    10    10    10
    11    11    11
    12    12    12
    13    13    13
    14    14    14
```

Funcionamiento de meshgrid

- Ejemplo:

```
[X, Y] = meshgrid(-3:3, -3:3)

X =

    -3    -2    -1     0     1     2     3
    -3    -2    -1     0     1     2     3
    -3    -2    -1     0     1     2     3
    -3    -2    -1     0     1     2     3
    -3    -2    -1     0     1     2     3
    -3    -2    -1     0     1     2     3
    -3    -2    -1     0     1     2     3

Y =

    -3    -3    -3    -3    -3    -3    -3
    -2    -2    -2    -2    -2    -2    -2
    -1    -1    -1    -1    -1    -1    -1
     0     0     0     0     0     0     0
     1     1     1     1     1     1     1
     2     2     2     2     2     2     2
     3     3     3     3     3     3     3
```

Funcionamiento de meshgrid

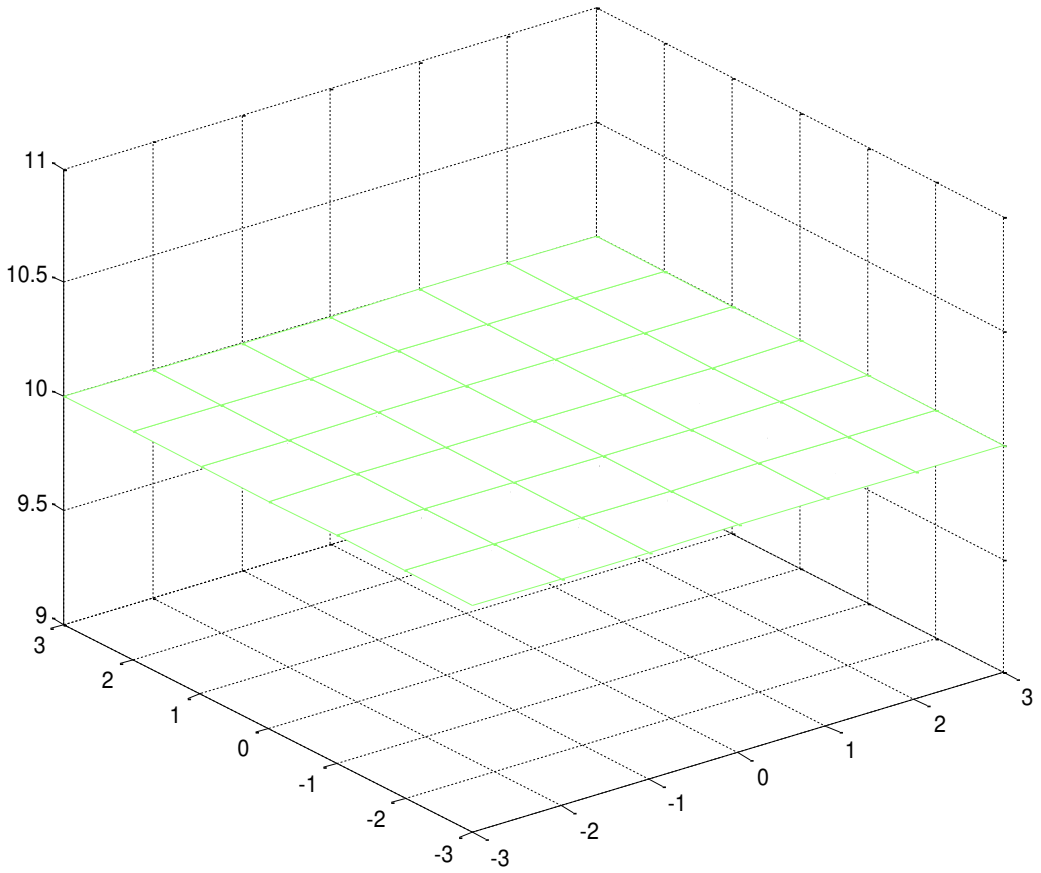
- Plano Z = 10:

```
>> Z = ones(7,7)*10

Z =

    10    10    10    10    10    10    10
    10    10    10    10    10    10    10
    10    10    10    10    10    10    10
    10    10    10    10    10    10    10
    10    10    10    10    10    10    10
    10    10    10    10    10    10    10
    10    10    10    10    10    10    10
```

Funcionamiento de meshgrid

- Gráfico:

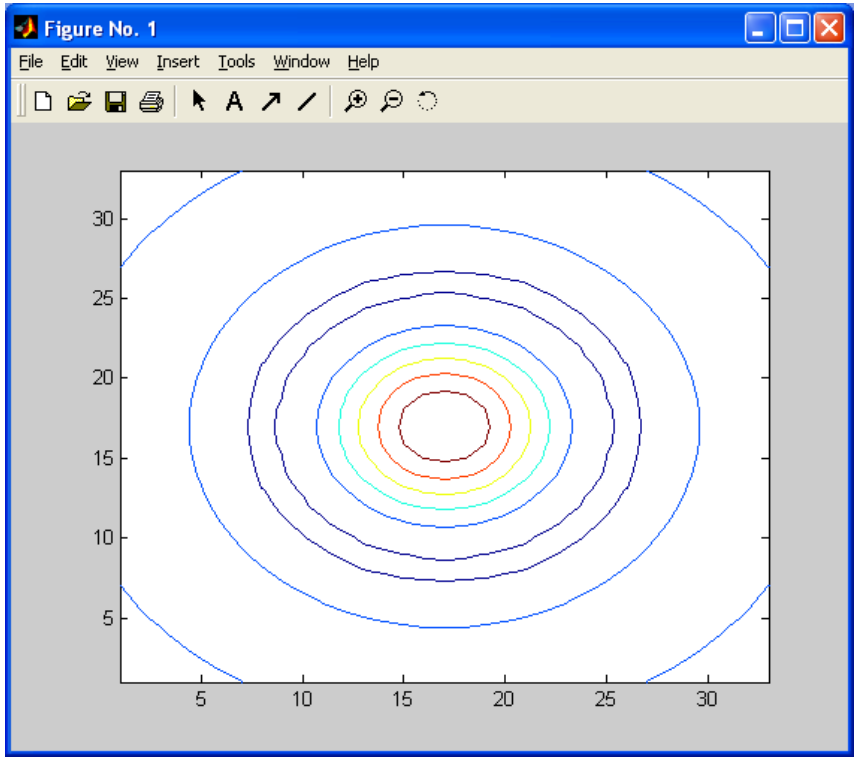
```
>> mesh(X,Y,Z)
```



Gráficos: contour

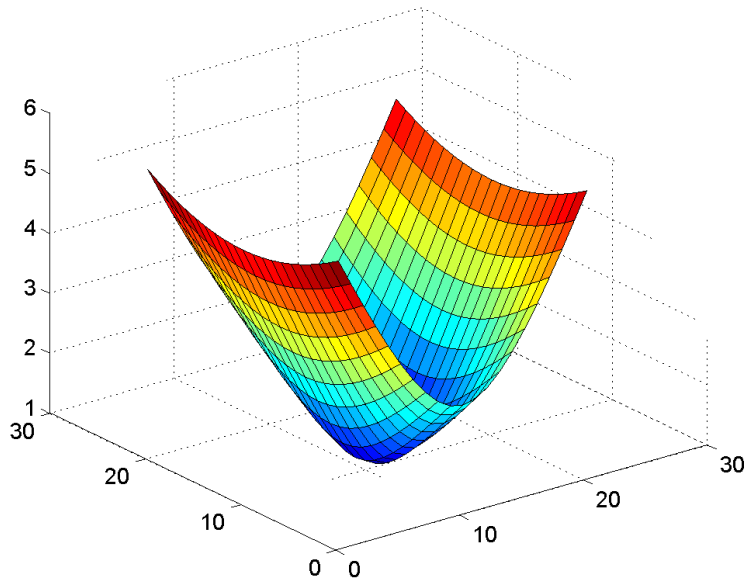
- Variantes:

```
contour( Z )
contour(X,Y,Z)
contour(Z,n)
contour(X,Y,Z,n)
```



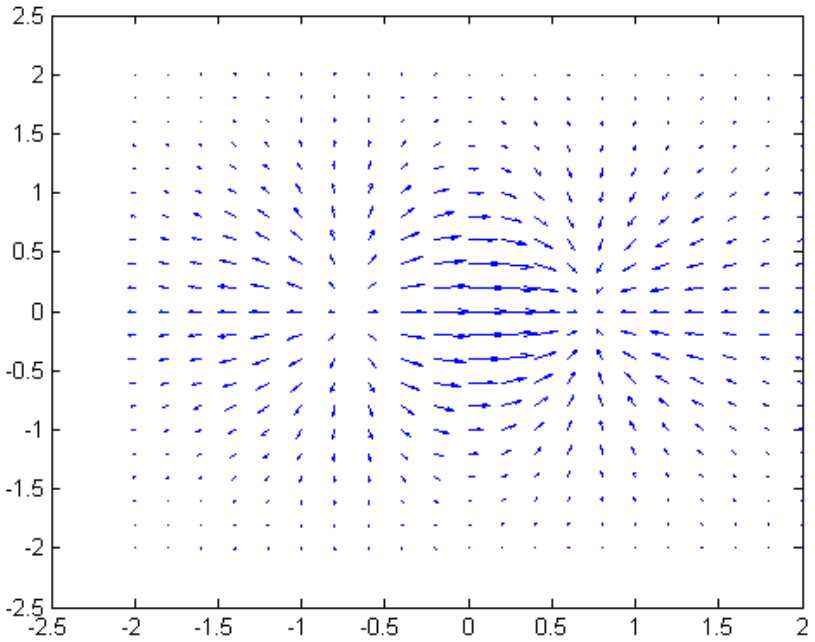
Gráficos: surf

```
x = [-5:.5:5];
y = x;
[X,Y] = meshgrid(x,y);
Z = sqrt(1+0.25*Y.^2+X.^2);
surf(Z)
```



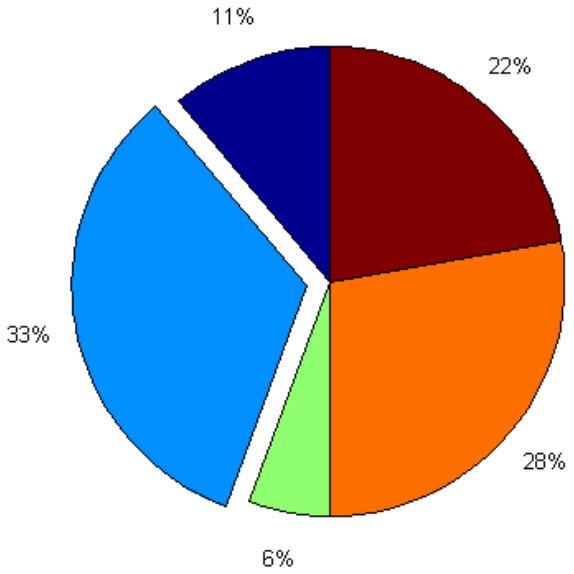
Gráficos: quiver

```
figure
[X,Y] = meshgrid(-2:.2:2);
Z = X.*exp(-X.^2 - Y.^2);
[DX,DY] = gradient(Z,.2,.2);
quiver(X,Y,DX,DY)
```



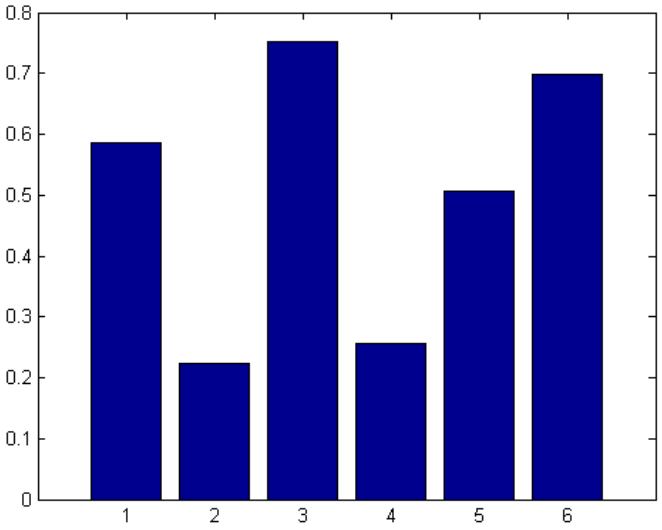
Gráficos: pie

```
x = [1 3 0.5 2.5 2];
explode = [0 1 0 0 0];
pie(x,explode)
colormap jet
```

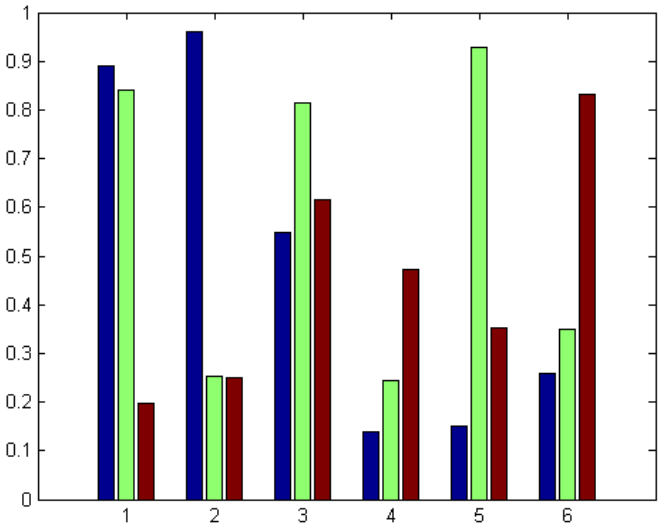


Gráficos: bar

```
A=rand(6,1);
bar(A)
```

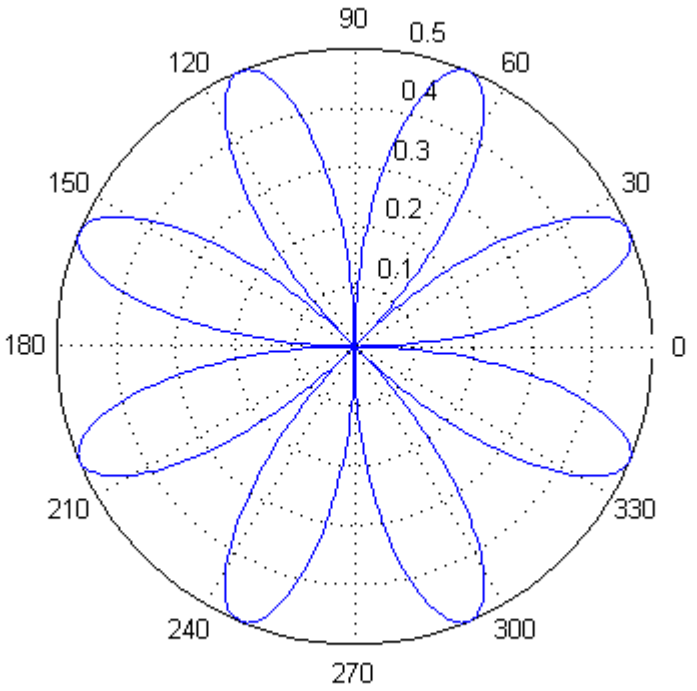



```
A=rand(6,3);
bar(A)
```



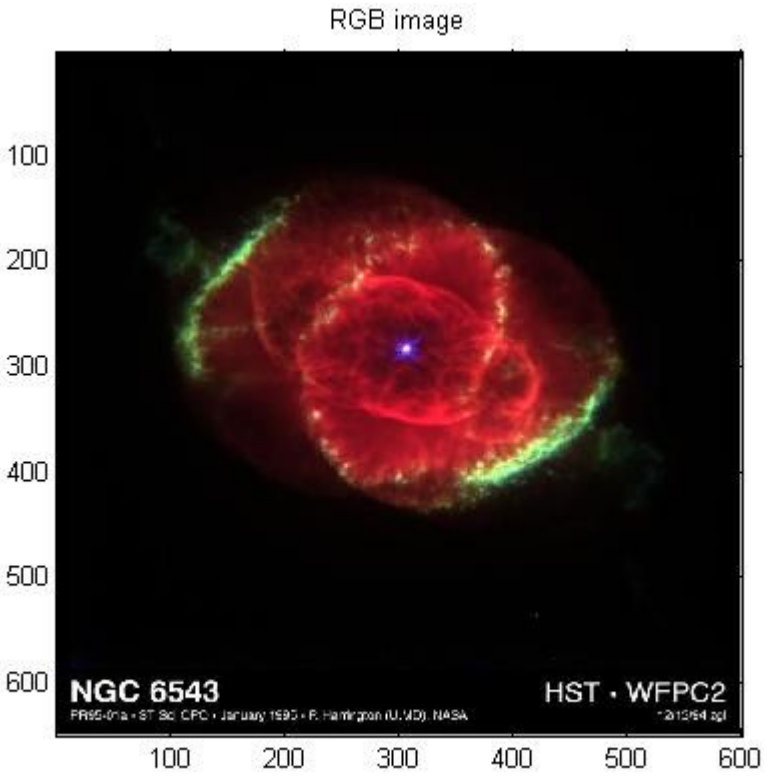
Gráficos: polar

```
figure
t = 0:.01:2*pi;
r=sin(2*t).*cos(2*t);
polar(t,r)
```



Gráficos: image

```
rgb =
   imread('ngc6543a.jpg');

image(rgb);

title('RGB image')
axis('square')
```



Gráficos: image

```
rgb = imread('ngc6543a.jpg');
im = mean(rgb,3);

image(im);

title('Intensity Heat Map')

colormap(hot(256))

axis('square')
```