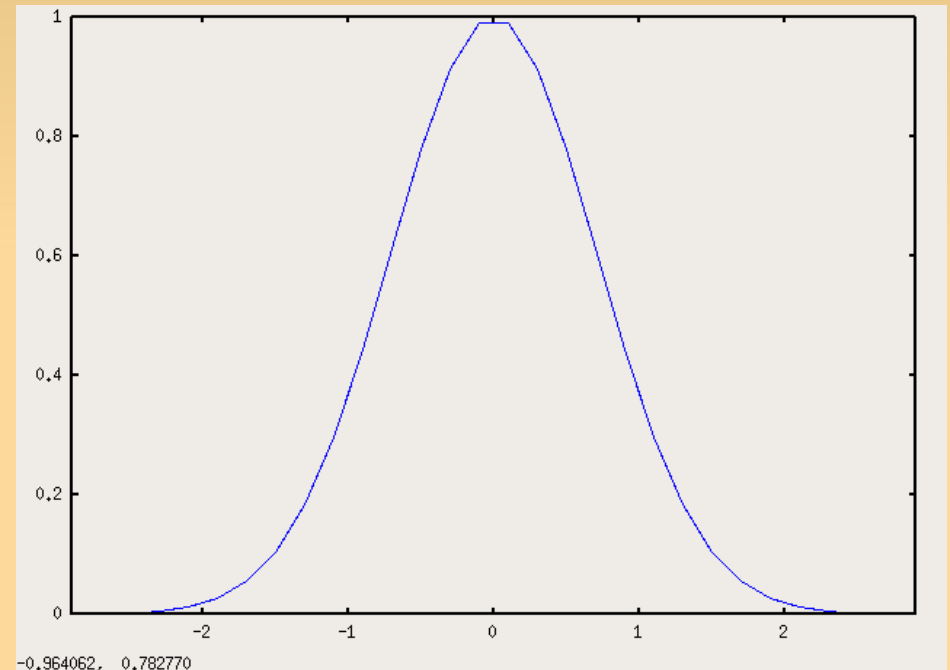


Gráficos

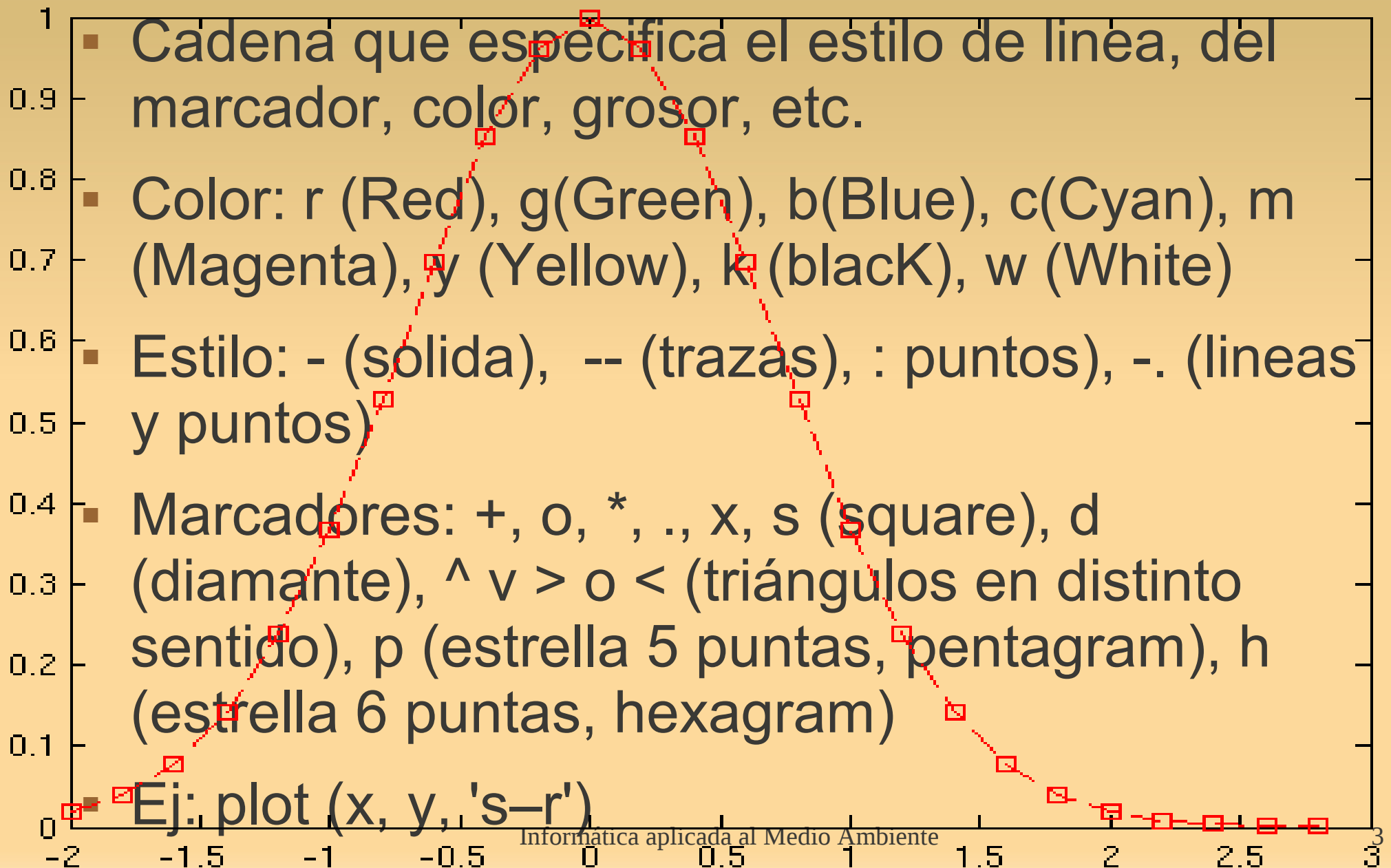
Informática aplicada al medio ambiente
curso 2010/2011

Gráficos en Matlab

- Comando básico:
`plot(x1, y1, x2, y2, x3, y3,..., 'opciones')`
- X e Y: Vectores de la misma longitud
- Opciones: Color, estilo, etc. del gráfico
- Ej.: `x= -2.9:0.2: 2.9;`
`y=exp(-x.*x); plot(x,y)`
- `plot(y)` → eje abcisas por omisión



Opciones



Objetos gráficos

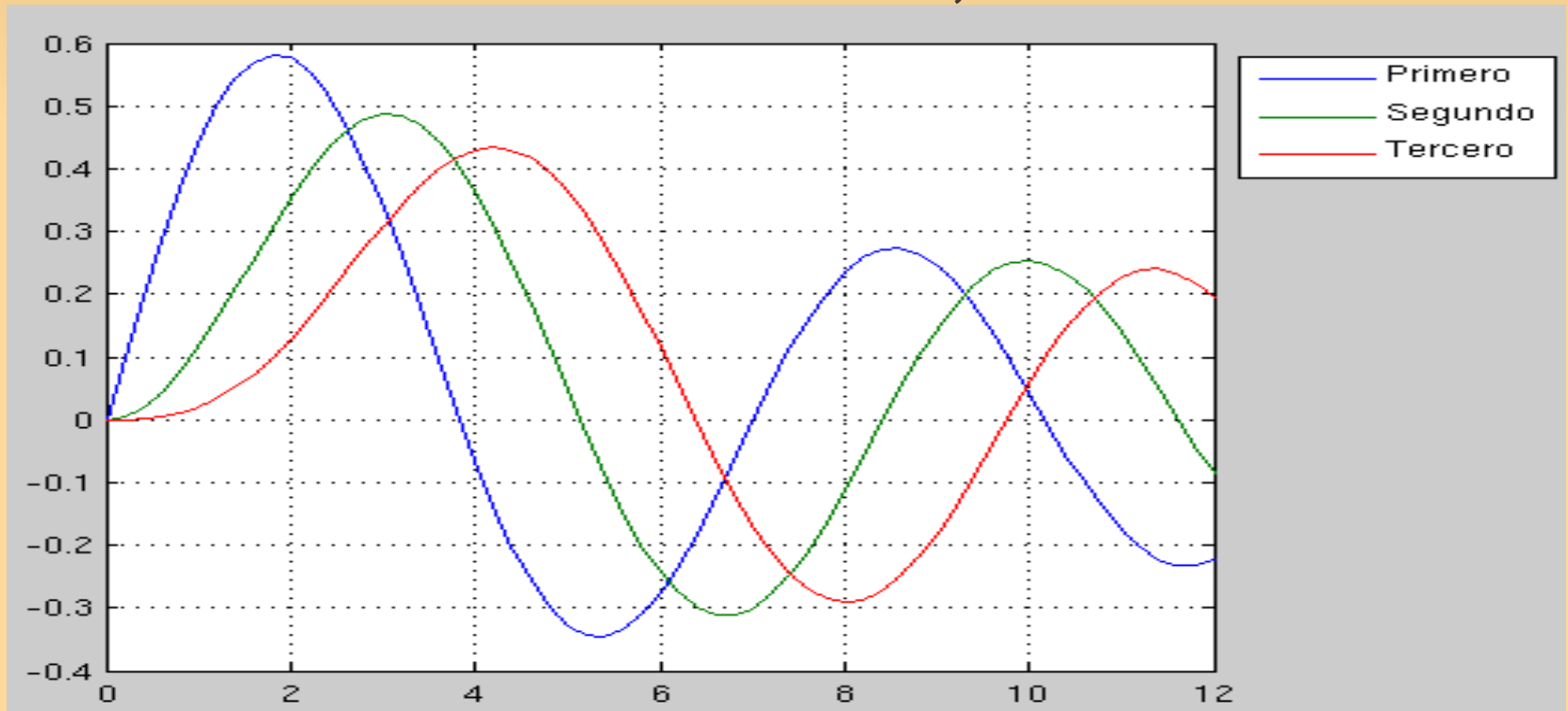
- Podemos añadir objetos al gráfico actual:
 - `xlabel('etiqueta del eje x')`
 - `ylabel('etiqueta eje y')`
 - `title('título del gráfico')`
 - `text (x, y, 'texto')`
 - Si x e y son vectores, el texto se repite
 - Si texto es una matrix de cadenas, de la misma dimensión que x e y, se situa cada texto en una posición
 - `gtext('texto')`: la posición se indica con el ratón.
(solo en Matlab)

Objetos gráficos

- Leyenda:
 - `legend('leyenda 1', 'leyenda 2', ...)` → añade la leyenda del eje, por cada serie de datos mostrada
 - `legend(..., 'location', posicion)`
 - `legend off | toggle`
 - `grid on | off | minor`
 - Rejilla del gráfico
- Posición
 - North = center top
 - South = center bottom
 - east = right center
 - west = left center
 - northeast = right top (por omisión)
 - Northwest = left top
 - Southeast = right bottom
 - southwest = left bottom
 - Best = Menor conflicto
 - +Outside: Margen
 - Ej: BestOutside

Ejemplo

- `x = 0:2:12; plot (x, bessel(1, x), x, bessel(2, x), x, bessel(3,x)); legen('Primero', 'Segundo', 'Tercero', 'Location', 'NorthEastOutside'); grid on`
- Soluciones diferencial de Bessel, variando el grado



Control de ejes

- `axis`: controla el escalado y apariencia de los ejes
- `axis ([xmin xmax ymin ymax])`
 - Fija los límites de los ejes x e y
- `axis equal` → mismo ratio de aspecto x e y
- `axis normal` ó `auto` → valor por omisión
- `axis square` → Ajusta los ejes para que el gráfico sea cuadrado (o cúbico en 3D)
- `axis off` | `on` → Oculta o muestra los ejes (lineas, marcas y etiquetas asociadas)

Creación de gráficos

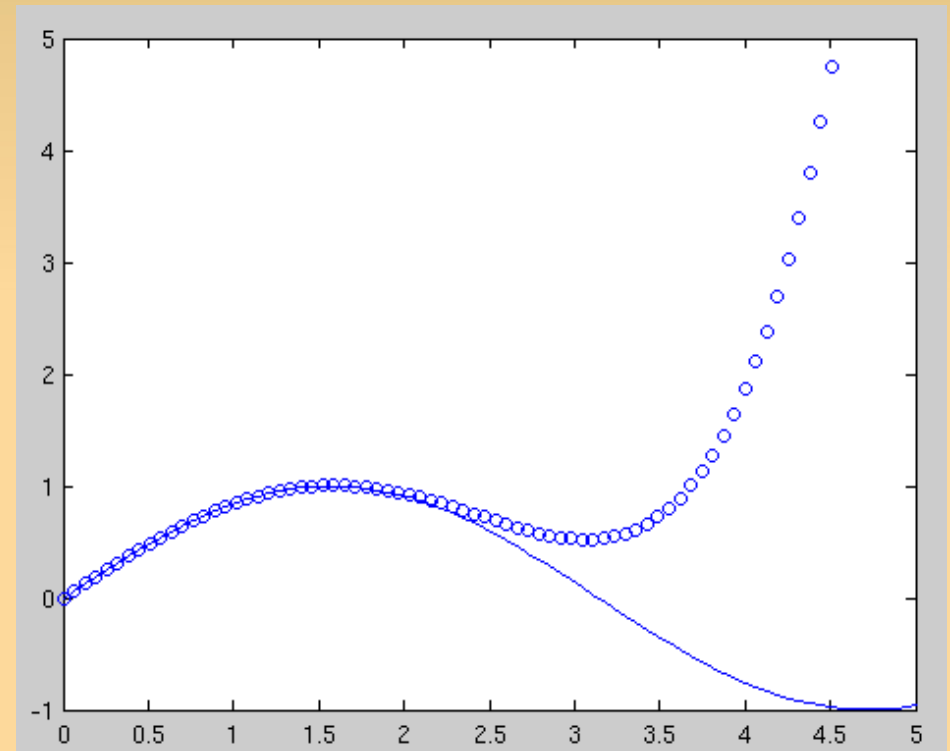
- Pasos
 - Cargar los datos
 - Procesar datos
 - Usar función de creación del gráfico (ej.: plot)
 - Situar límites de los ejes, marcas, mallas, textos, etc.

Superposición de gráficos

- Pares de vectores en el mismo comando de creación del gráfico
- Usando hold on / hold off
 - El gráfico se crea por etapas
 - Es útil cuando los datos a dibujar no están disponibles al mismo tiempo

Ejemplo

- Aproximación de Taylor para la función seno.
- `x=linspace(0, 2*pi, 100)`
- `y1=sin(x)`
- `plot(x, y1)`
- `hold on`
- `y2= x- (x.^3)/6 + (x.^5)/120`
- `plot(x, y2, 'o')`
- `axis ([0 5 -1 5])`
- `hold off`



Superposición de gráficos

- Usando line (x, y, 'parámetros', 'valor')
- Parámetros
 - color
 - linestyle
 - Marker

%Ejemplo de hold

```
x=linspace(0, 2*pi, 100)
```

```
y1=sin(x)
```

```
plot(x, y1)
```

```
y2= x- (x.^3)/6 + (x.^5)/120
```

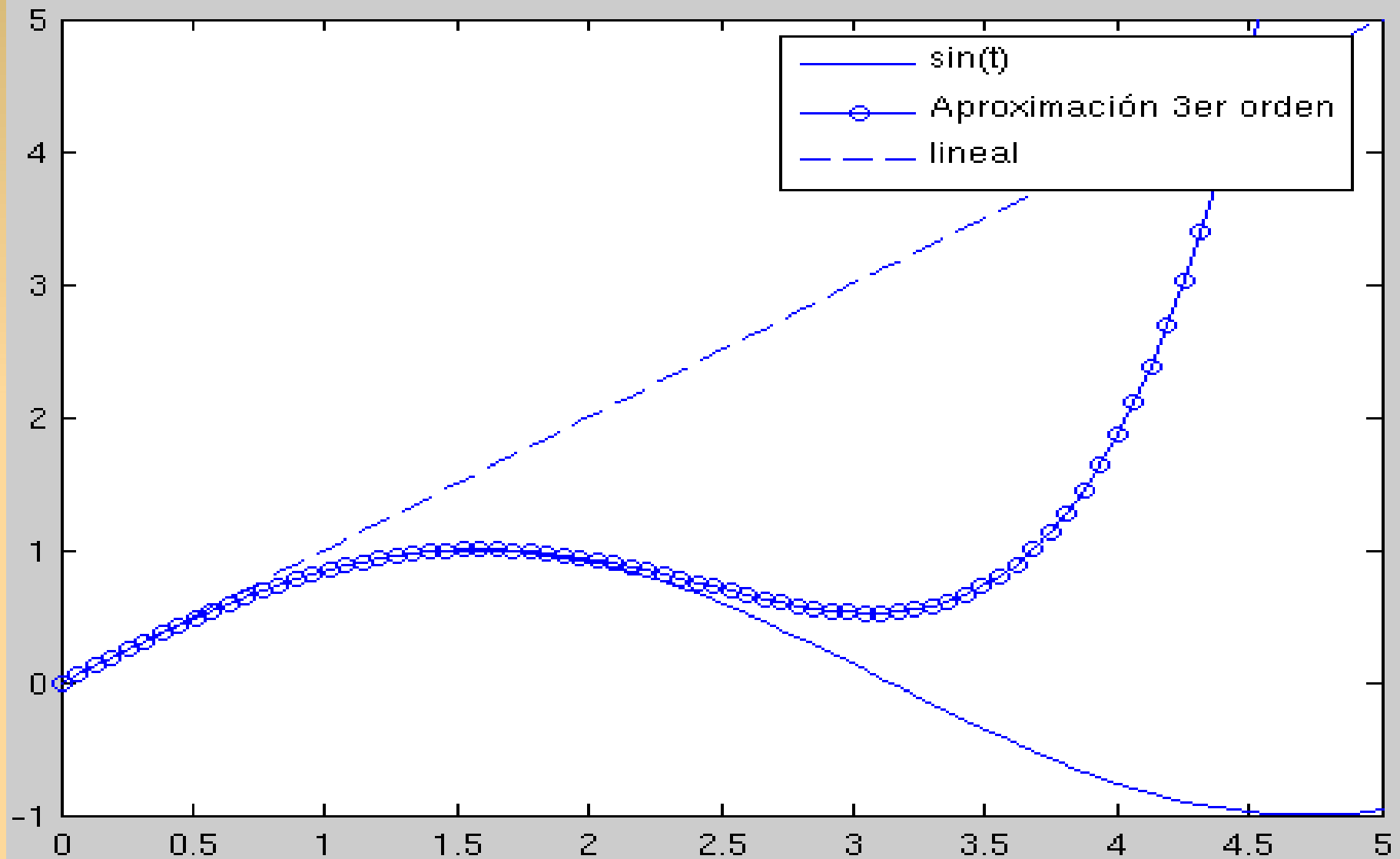
```
line(x, y2, 'marker', 'o')
```

```
line(x, x, 'linestyle', '--')
```

```
axis ([ 0 5 -1 5])
```

```
legend('sin(t)', 'Aproximación 3er  
orden', 'lineal')
```

Ejemplo



Ejes logarítmicos

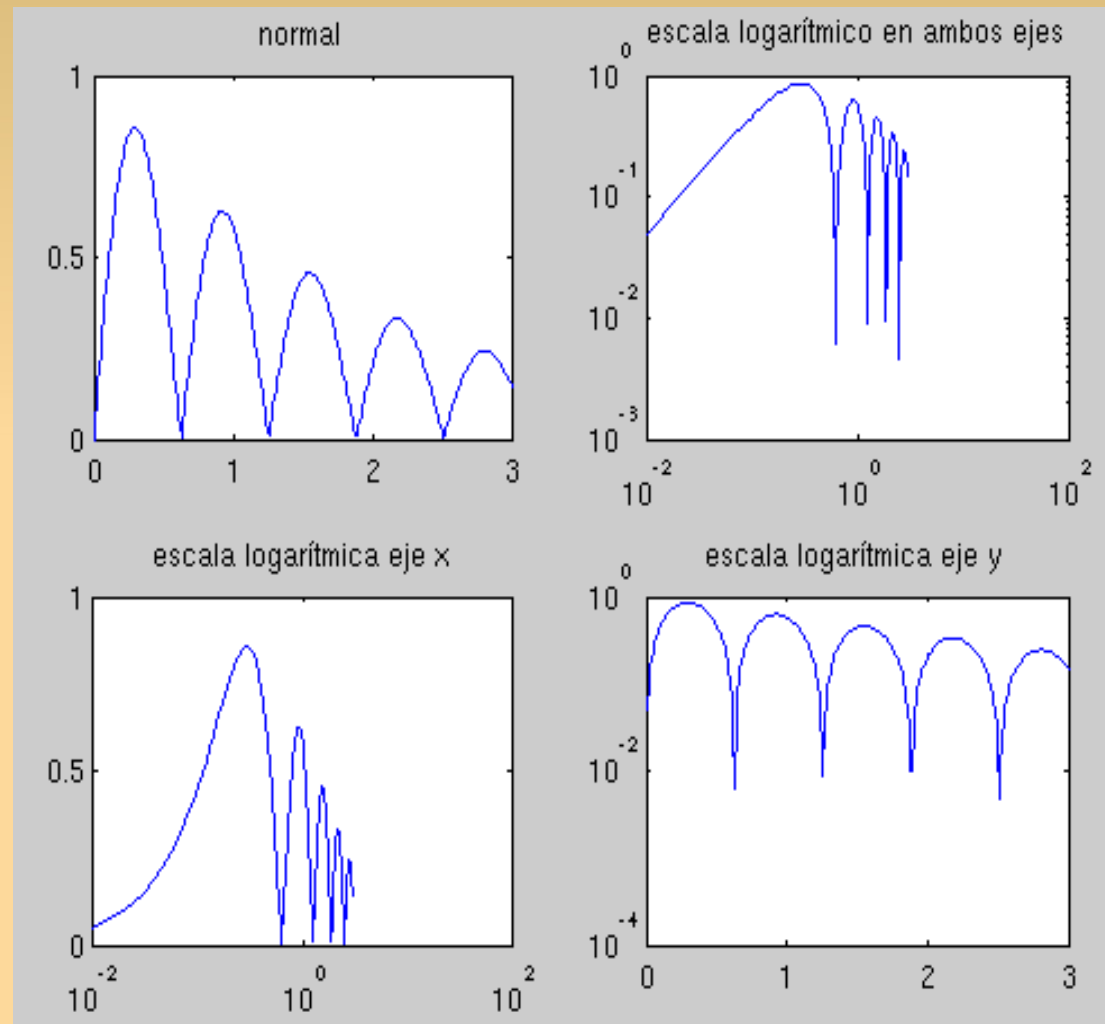
- `loglog (x, y)` : ambos ejes en escala logarítmica
- `semilogx(x, y)`: eje X logarítmico
- `semilogy(x, y)`: eje y logarítmico

Gráficos múltiples

- subplot (m, n, p)
 - Divide la ventana gráfica en MxN subventanas
 - Asigna la ventana p-ésima como la actual, donde se dibujará el gráfico

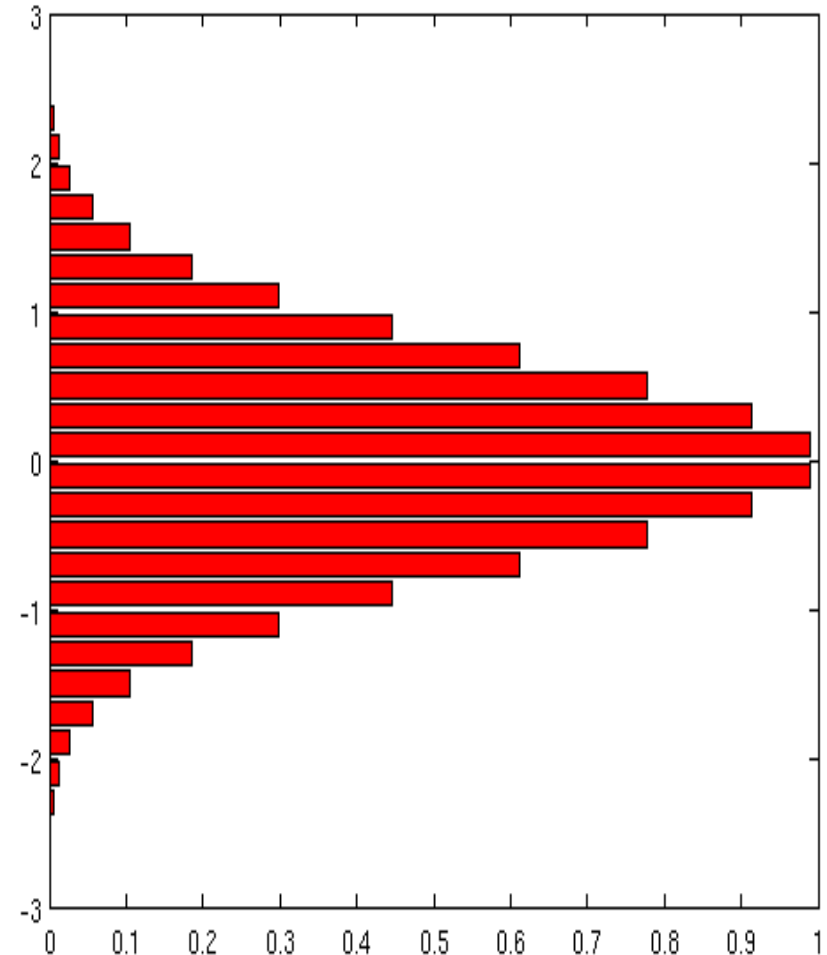
Ejemplo

```
X=[0:0.01:3];  
y=abs(exp(-0.5*x).*sin(5*x));  
subplot( 2, 2, 1); plot (x, y)  
title('normal'); hold on  
subplot (2, 2, 2); loglog (x, y)  
title ('escala logarítmico en ambos  
ejes')  
subplot(2, 2, 3); semilogx(x, y)  
title ('escala logarítmica eje x')  
subplot (2, 2, 4); semilogy(x, y)  
title('escala logarítmica eje y')  
hold off
```



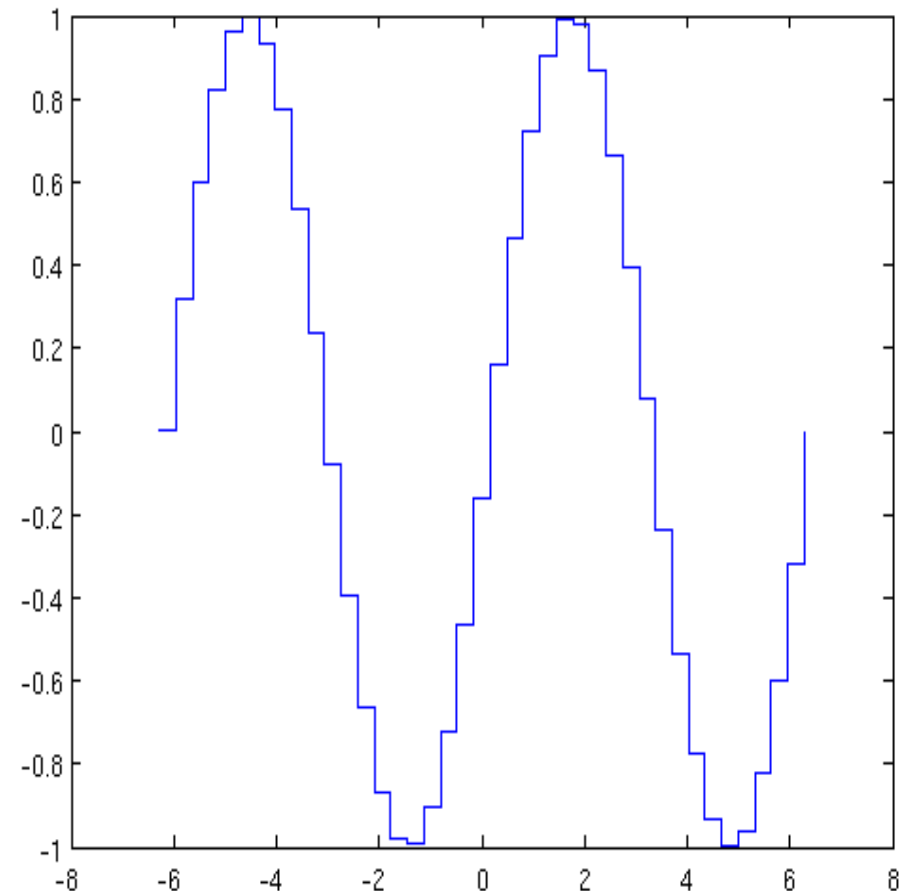
Gráficos especiales

- Gráfico de barras
 - `bar(x, y, 'opciones plot', 'tipo')`
 - Tipos:
 - Apilado: 'stacked'
 - Agrupado: 'grouped', valor por omisión
 - `barh`: En horizontal
 - Ejemplo
 - `x = -2.9:0.2:2.9;`
 - `barh(x,exp(-x.*x),'r')`



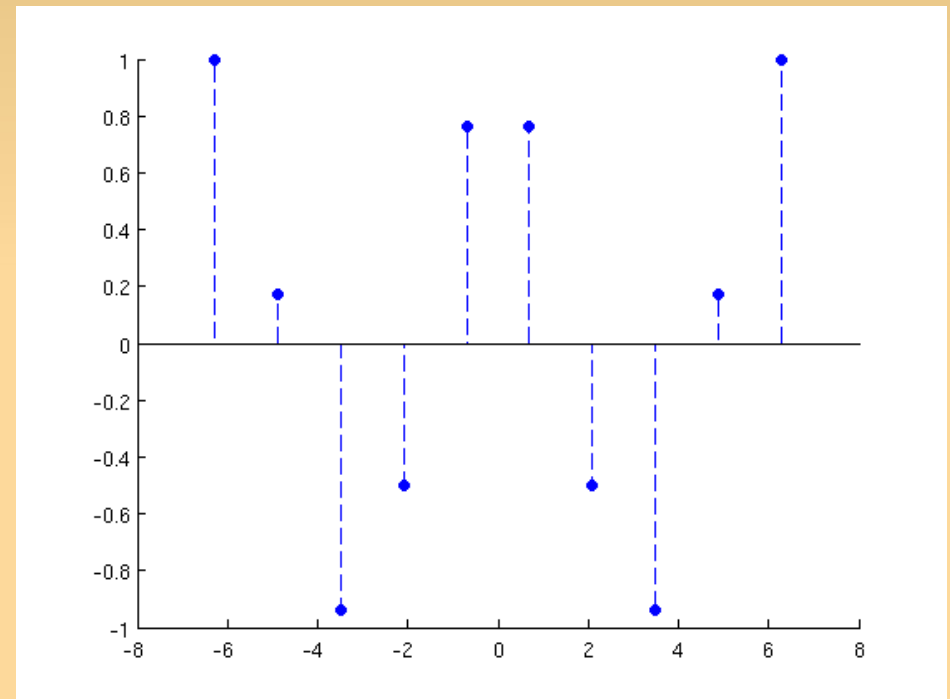
Gráficos especiales

- Gráfico de escaleras:
`stairs(x, y)`
 - Útil para series temporales o datos digitales
- Ejemplo:
 - `x = linspace(-2*pi, 2*pi, 40)`
 - `stairs(x, sin(x))`



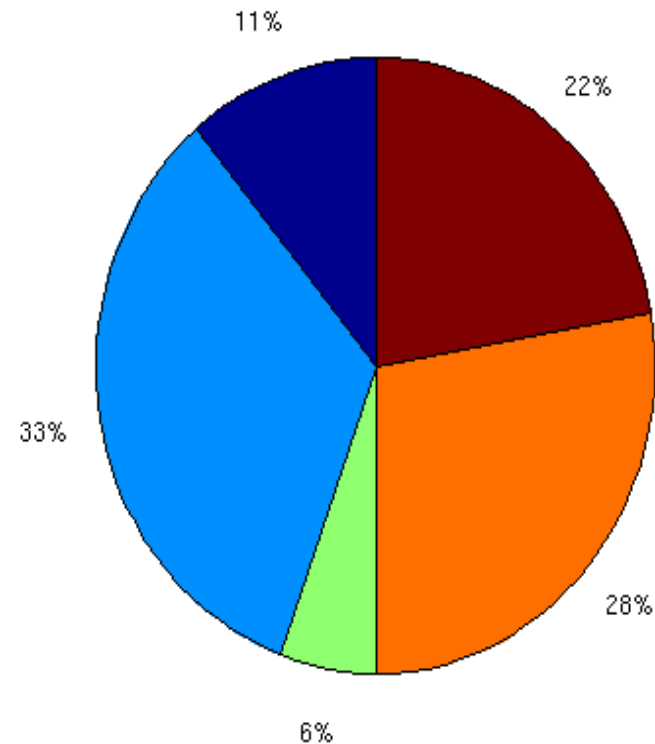
Gráficos especiales

- Gráfico de tallo o líneas verticales:
 - `stem(x, y, opciones)`
- Ejemplo
 - `t = linspace(-2*pi, 2*pi, 10)`
 - `h = stem(t, cos(t), 'fill', '--');`



Gráficos especiales

- Gráfico de tarta
 - `pie(X, opciones...)`
 - Representa los valores de una serie de datos, normalizados a $X/\text{sum}(X) \rightarrow$ porcentaje del total
- Ejemplo
 - `x = [1 3 0.5 2.5 2];`
 - `pie(x)`

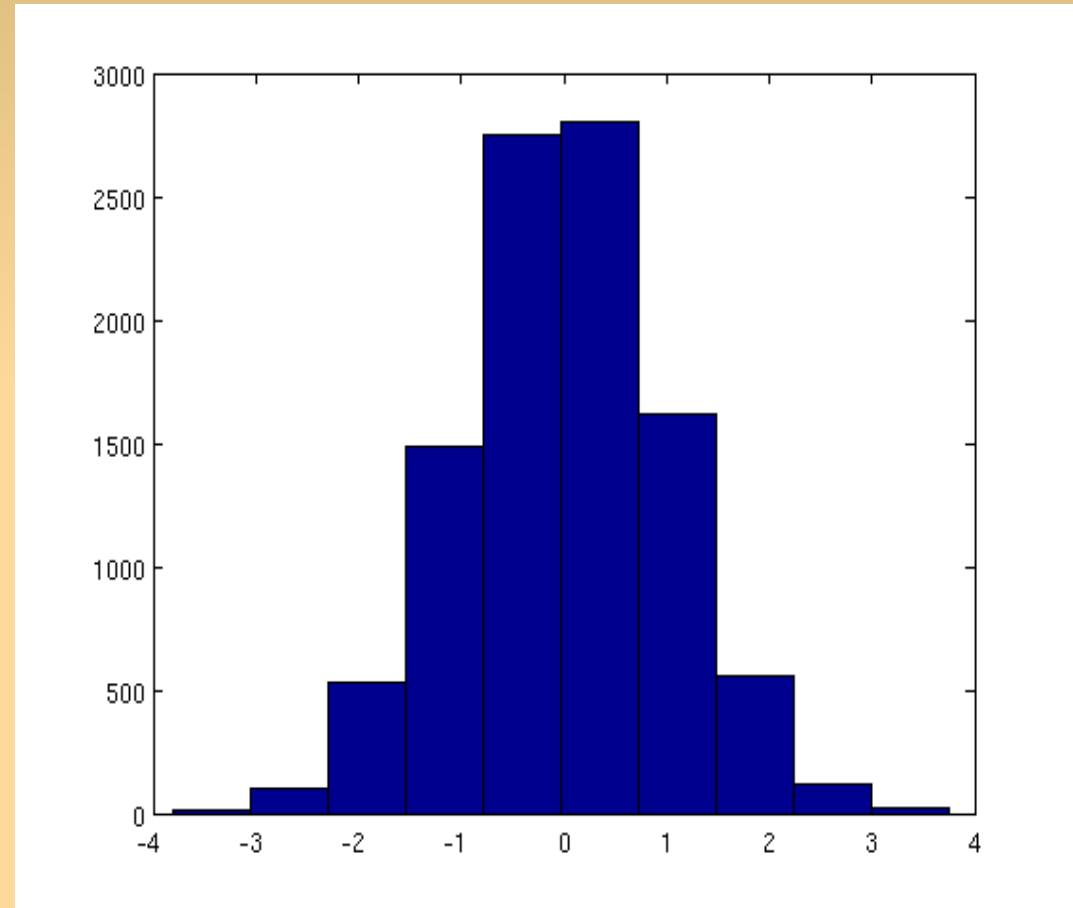


Histogramas (1/2)

- Muestran la distribución de una serie de datos, representando cuantos puntos hay en cada intervalo
- `hist(y)`
 - Representa el histograma mediante barras verticales
 - Ancho = rango
 - Altura = Puntos en el intervalo.
 - Por omisión 10 intervalos equidistantes

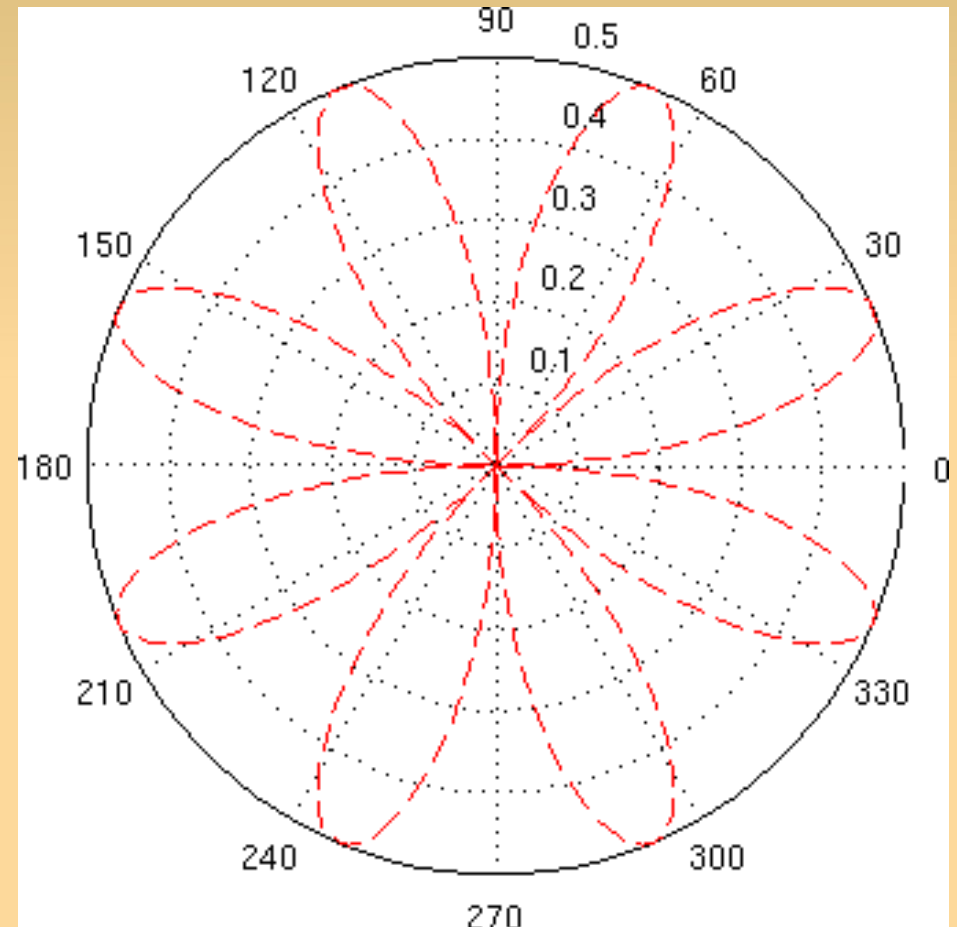
Histogramas (2/2)

- `hist(y, n_intervalos)` : fijamos el número de intervalos
- `hist(y, x)` : `x` = vector con ancho de cada intervalo
- Ejemplo:
 - `yn = randn(10000,1);`
 - `hist(yn)`



Gráficos en coordenadas polares

- `polar(angulos, radios)`
- Dibuja las coordenadas polares en un plano, indicando los angulos y el radio de cada punto
- Ejemplo:
 - `t = 0:.01:2*pi;`
 - `polar(t,sin(2*t).*cos(2*t),'--r')`

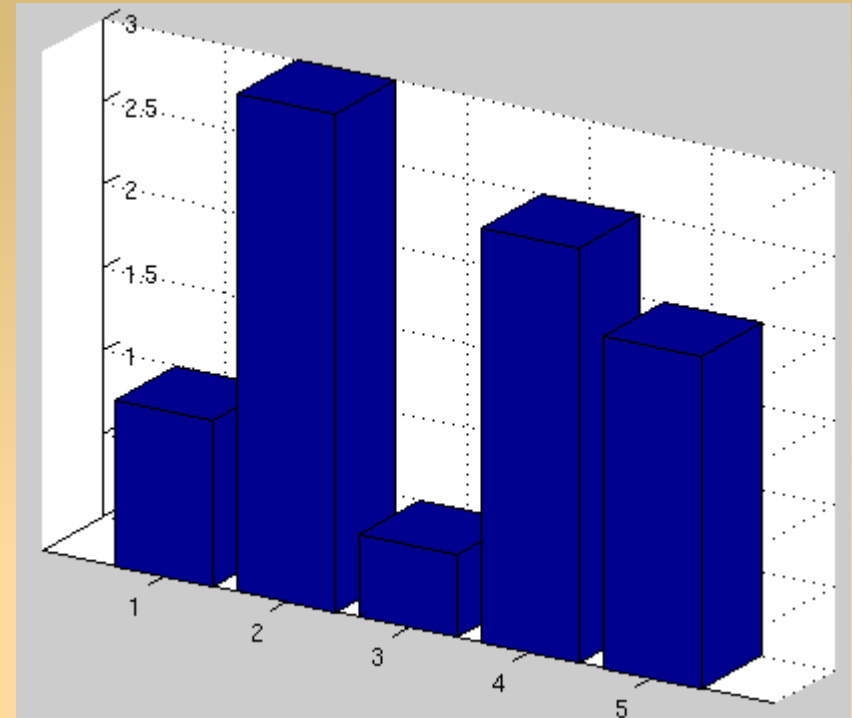


Gráficos 3D

- Representación en 3D, datos 1D o 2D

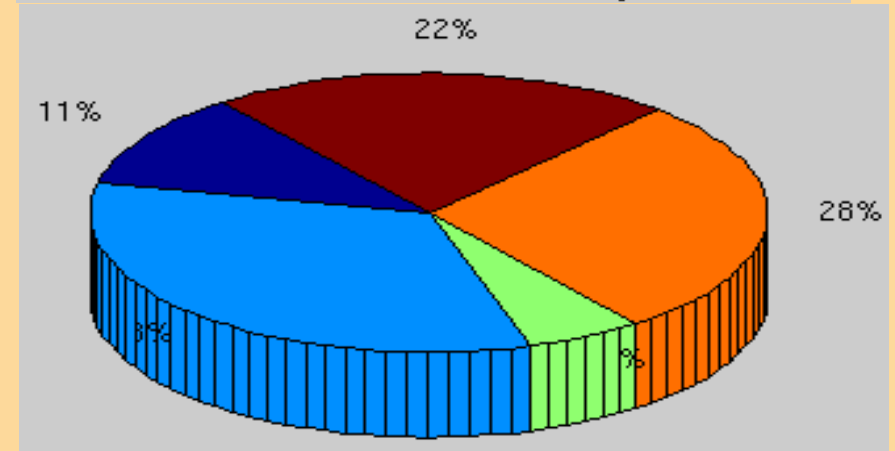
- Barras 3D

- $x = [1 \ 3 \ 0.5 \ 2.5 \ 2]$
 - `bar3(x)`



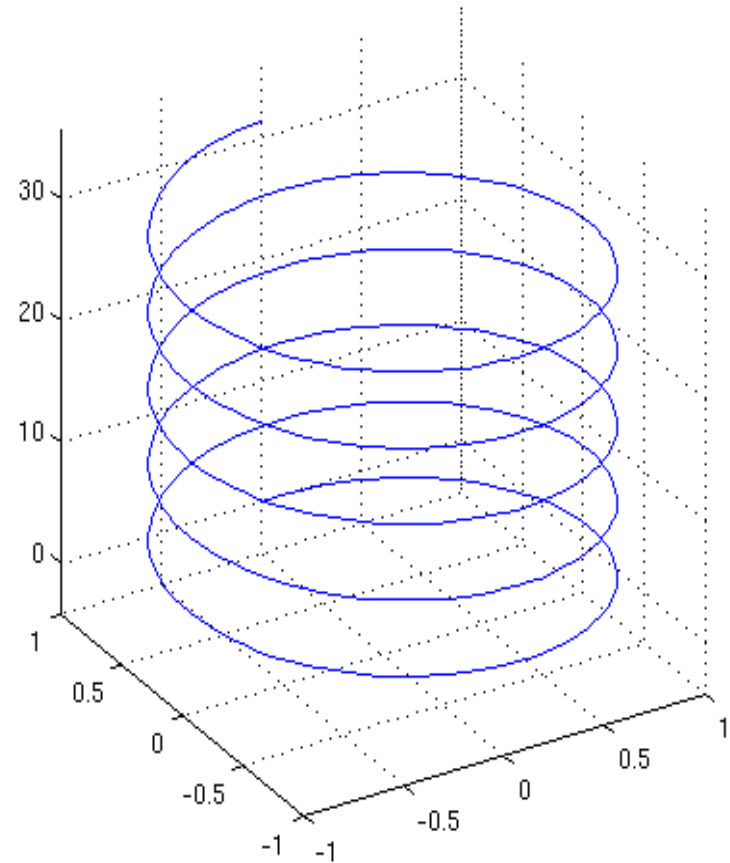
- Tarta 3D

- $x = [1 \ 3 \ 0.5 \ 2.5 \ 2]$
 - `pie3(x)`



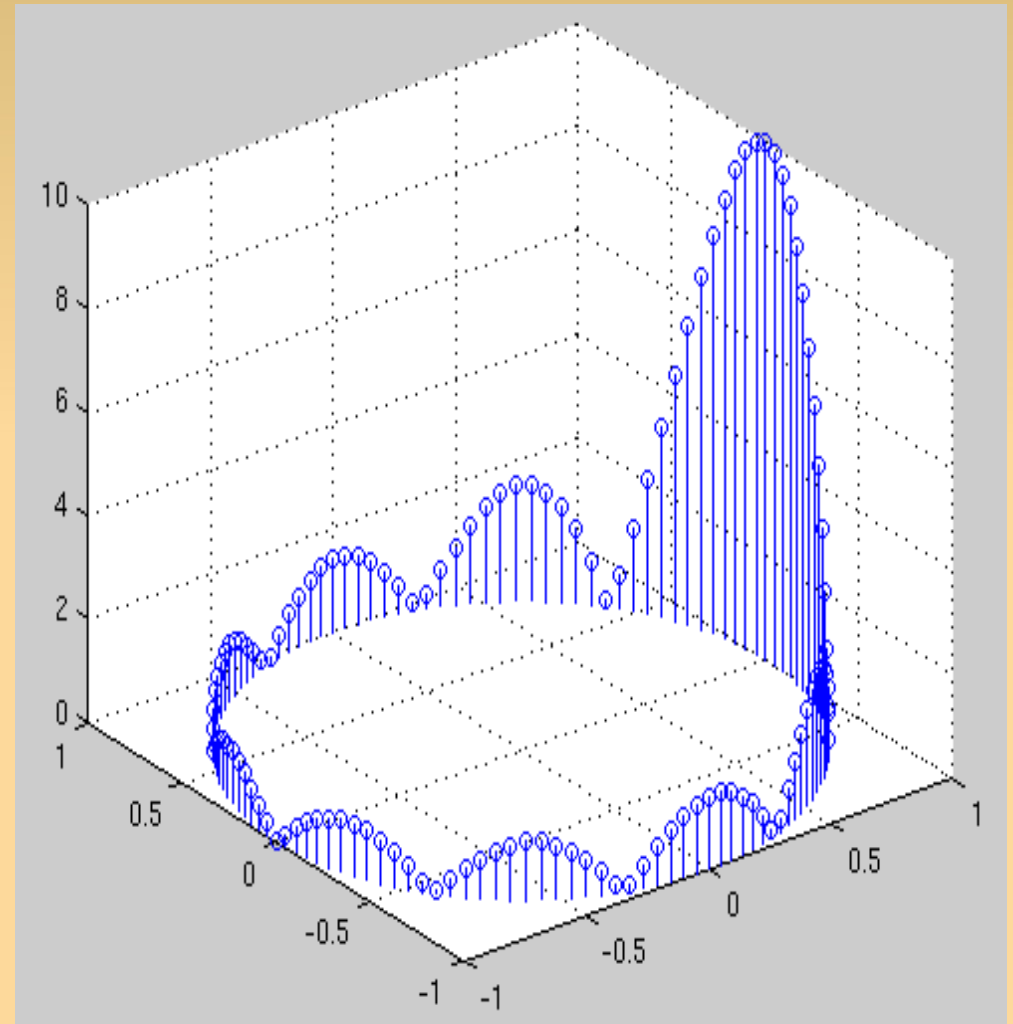
Gráficos 3D

- Gráficos de líneas:
 - `plot3(x, y, z):`
Equivalente a `plot`,
para dibujar líneas en
3D
- Ejemplo
 - `t = 0:pi/50:10*pi;`
 - `plot3(sin(t),cos(t),t)`
 - `axis square; grid on`



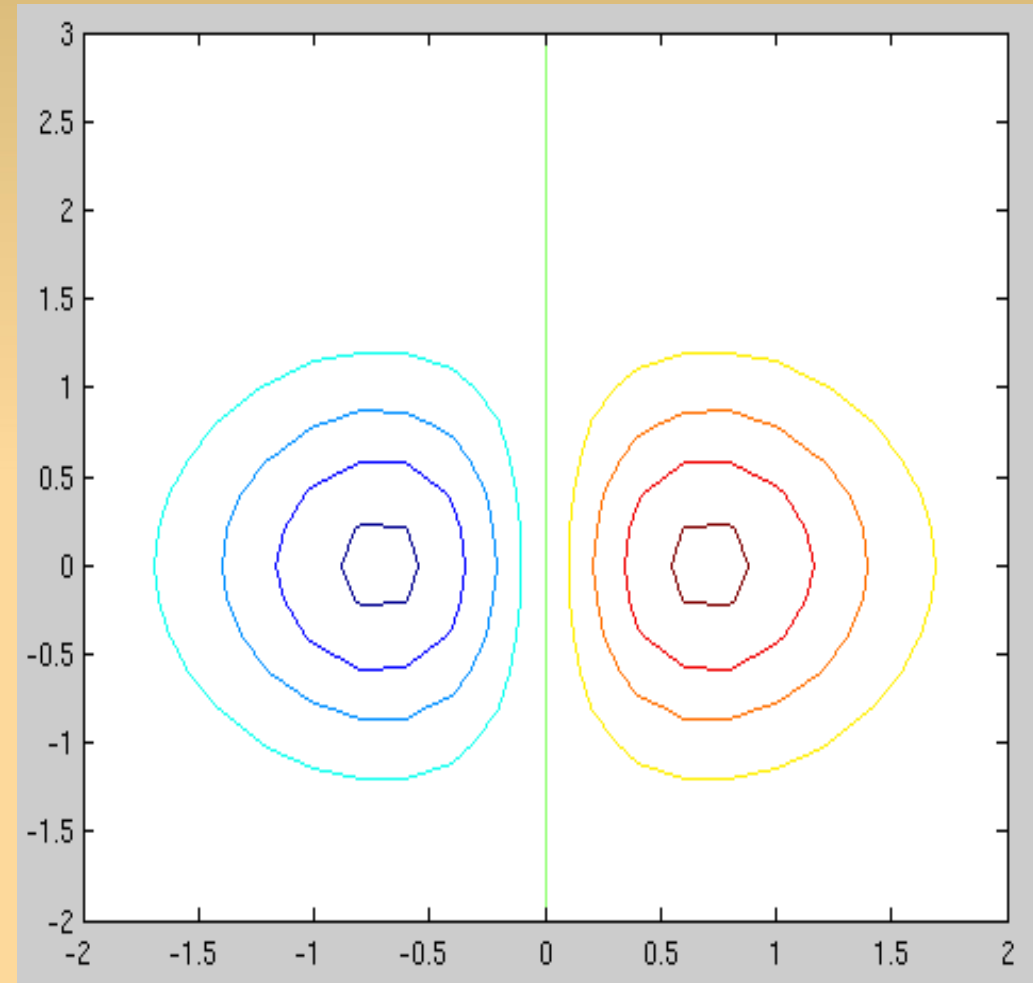
Gráficos 3D

- Gráfico de tallos 3D
- Ejemplo:
transformada rápida
de fourier (fft)
 - $th = (0:127)/128*2*pi;$
 - $x = \cos(th);$
 - $y = \sin(th);$
 - $f =$
 $abs(fft(ones(10,1),128$
 $));$
 - $stem3(x,y,f')$



Contornos

- `contour(x, y, z)`
 - Permiten dibujar isolineas en 2D
 - Z = matriz $N \times M$, representa la altura
- Ejemplo
 - `[X,Y] = meshgrid(-2::2:2,-2::2:3);`
 - `Z = X.*exp(-X.^2-Y.^2);`
 - `contour(X,Y,Z);`



Superficies

- `mesh(x, y, z)`
 - Representan la altura de una serie de puntos en una rejilla
- `meshc` → muestra contorno debajo
- Ejemplo
 - `[X,Y] = meshgrid(-3:125:3);`
 - `Z = peaks(X,Y);`
 - `meshc(X,Y,Z)`

