

Gráficos 2D

- Sentenza *plot*: varias formas
- Representa os elementos do vector *x* fronte ao índice ou nº de elemento: *plot(x)*
- Se *x* e *y* son vectores co mesmo nº de elementos, representa *y* fronte a *x*: *plot(x, y)*
- Indicando propiedades do gráfico:

plot(x, y, 'r-')*

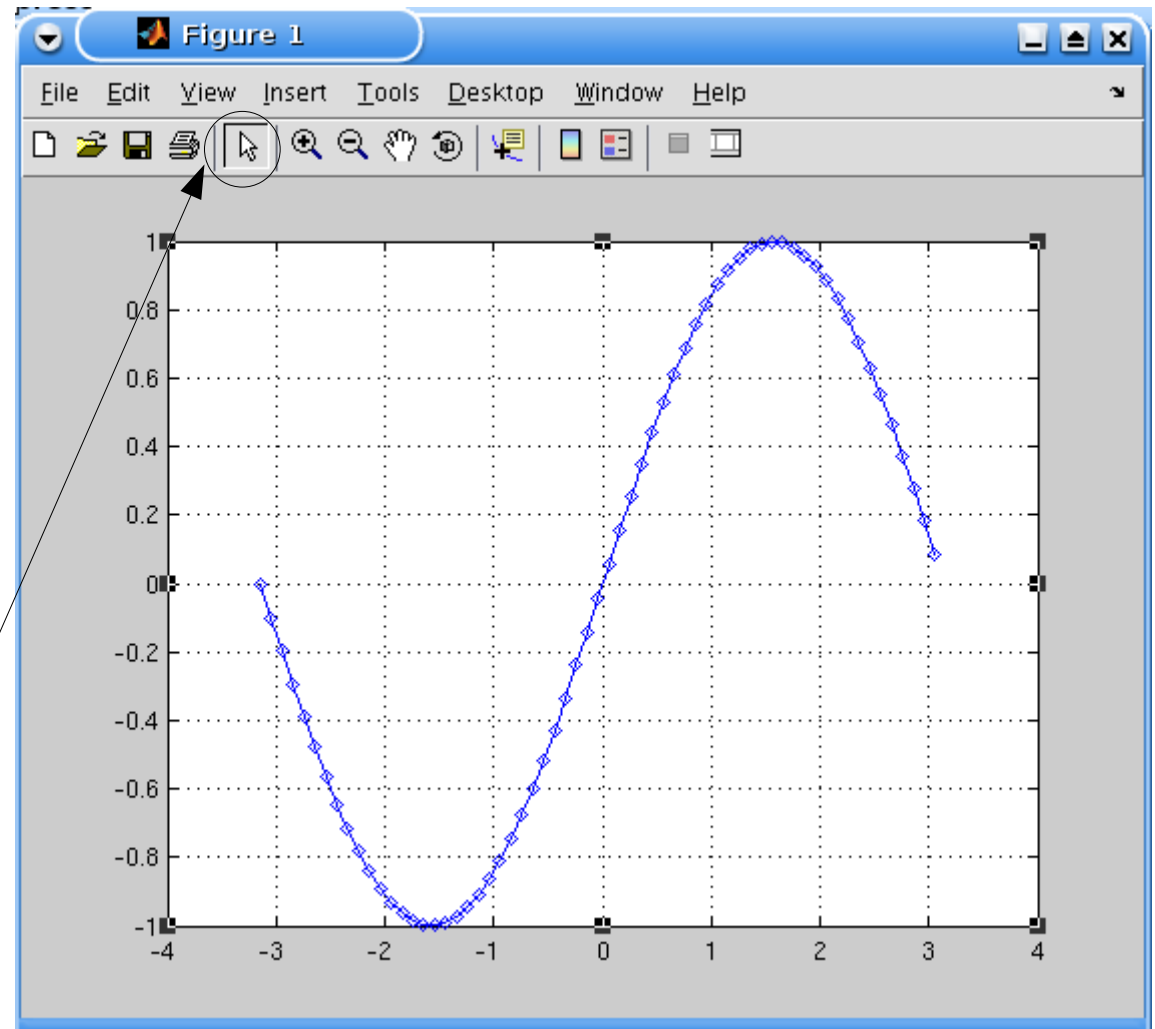
r=red, *-*= liña, ***=punto con forma de asterisco

- Se *a* é unha matriz, *plot(a)* representa cada columna como unha gráfica distinta (e cores distintas)
- Podes representar varias gráficas no mesmo *plot*:

plot(x,sin(x),'bs-',x,cos(x),'or-',x,x.^2,'vg-')

Ventá de figuras

- Non é necesario nin eficiente saber tódalas opcións (cores, tipos de liña e punto, propiedades dos eixos, etc.). Pódense modificar na ventá do plot



Exportación de figuras a ficheiros de imaxe

- Manualmente: gardar ou imprimir en ventá de figura.
- Automáticamente dende ventá de comandos ou dende programa: función *print*.
- A arquivo en formato encapsulado postscript (EPS):
print('-depasc', 'arquivo.eps');
- Formato portable network graphics (PNG):
print('-dpng', 'arquivo.png');
- Formato PDF: *print('-dpdf', 'arquivo.pdf');*
- Formato TIFF: *print('-dtiff', 'arquivo.tif')*
- Formato JPEG: *print('-djpeg', 'arquivo.jpg')*

Representación de funciones (expresión analítica)

- Función *ezplot*:

ezplot('función', [min, max])

ezplot('función'): en $[-2\pi, 2\pi]$

Ex: *ezplot('exp(-x^2)*sin(x)', [0,2*pi])*, *ezplot('sin(x)')*

- Función *fplot*:

fplot(@(x) x.^2,[min max])

- Función anónima con operaciones componente a componente: *.* ./ .^*

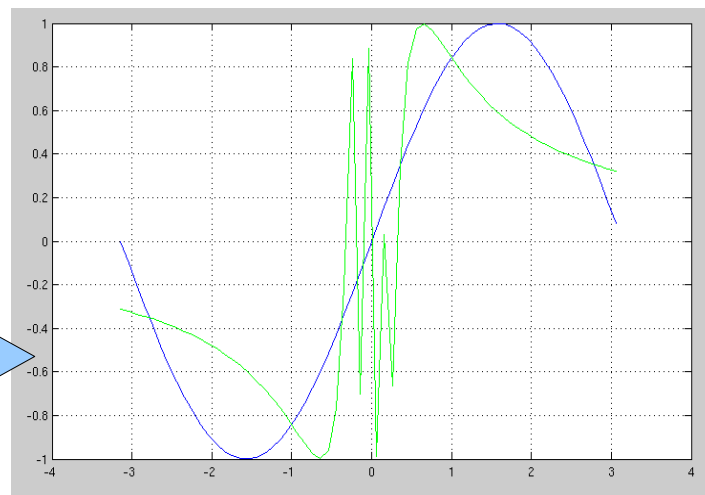
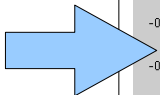
Comandos relacionados

- *xlabel('título eixo X')*, ídem con *ylabel*
- *title('título do gráfico')*
- *clf*: borra a gráfica actual
- *figure(2)*: crea unha nova ventá de figura
- *hold on*: permite representar unha gráfica mantendo a(s) anterior(es); *hold off*: desactiva isto
- *axis([xmin xmax ymin ymax])*: establece rangos en ambos eixos
- *axis equal*: igual lonxitude para a unidade en X e Y
- *axis square*: figura cadrada e non rectangular
- *grid on (off)*: pon (quita) o enreixado; *grid*: conmuta

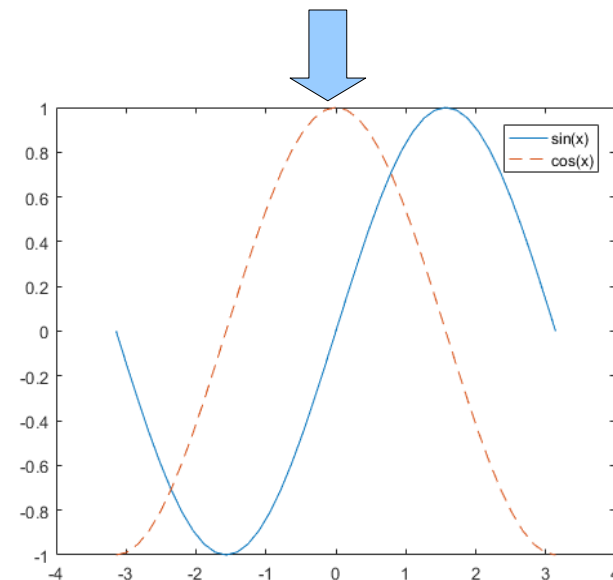
Múltiples gráficas na mesma ventá

- Con *hold on*:
- Comando *legend*:

```
clf  
x=-pi:0.1:pi;  
y=sin(x);  
z=sin(1./x);  
plot(x,y)  
grid on  
plot(x,z,'g')  
plot(x,y)  
grid on  
hold on  
plot(x,z,'g')
```



legend({'gráfica1' 'gráfica2'},
'location', 'northwest')
Outras *location*: east, southwest,...



Múltiples gráficas en distintas sub-ventás

- Crea unha matriz de figuras (2x2, 3x2, ...)

subplot(filas,columnas,actual)

- A 1ª execución de subplot crea a matriz (sen figuras) e crea a figura (1,1). Execútase unha vez por cada figura, e crea a figura actual ($1 \leq actual \leq filas*columnas$)

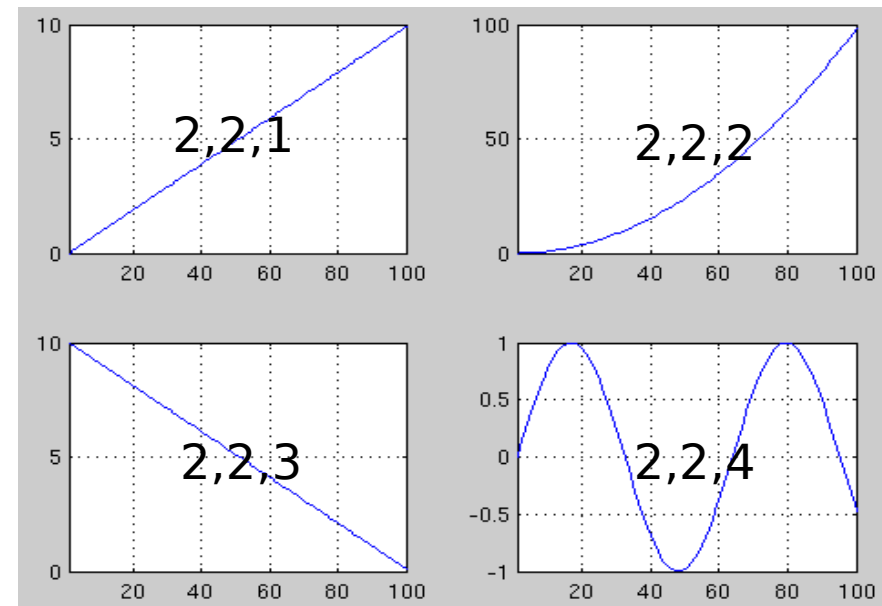
```
x=0:0.1:10;y=x.^2;z=10-x;  
t=sin(x);
```

```
subplot(2,2,1); plot(x)
```

```
subplot(2,2,2); plot(y)
```

```
subplot(2,2,3); plot(z)
```

```
subplot(2,2,4); plot(t)
```



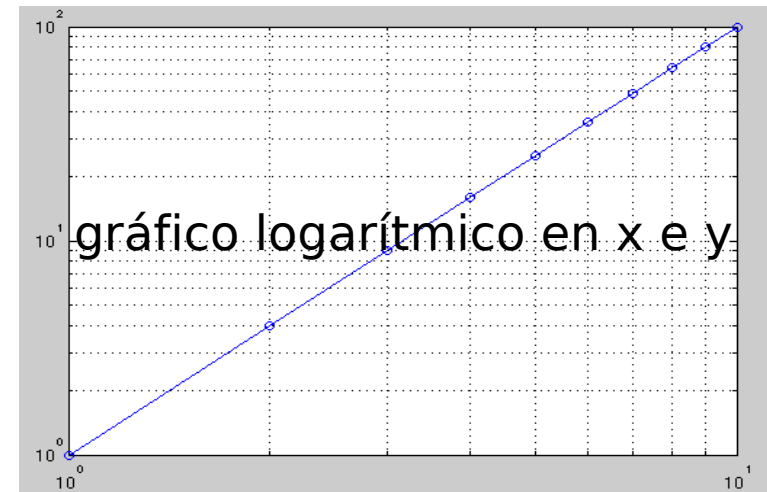
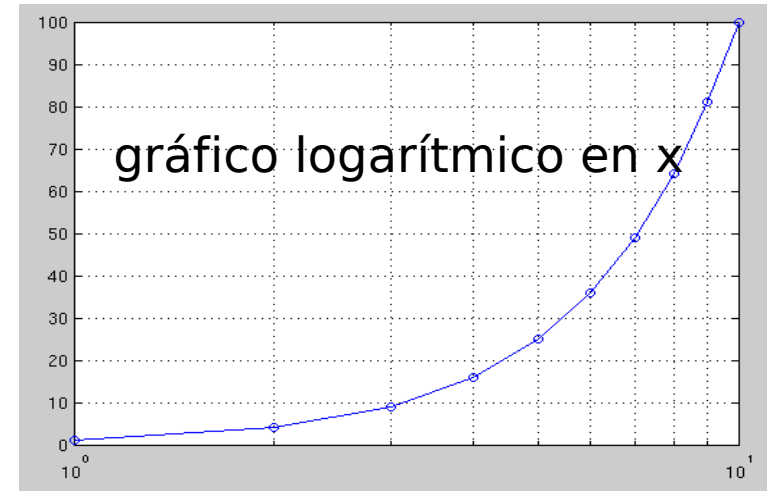
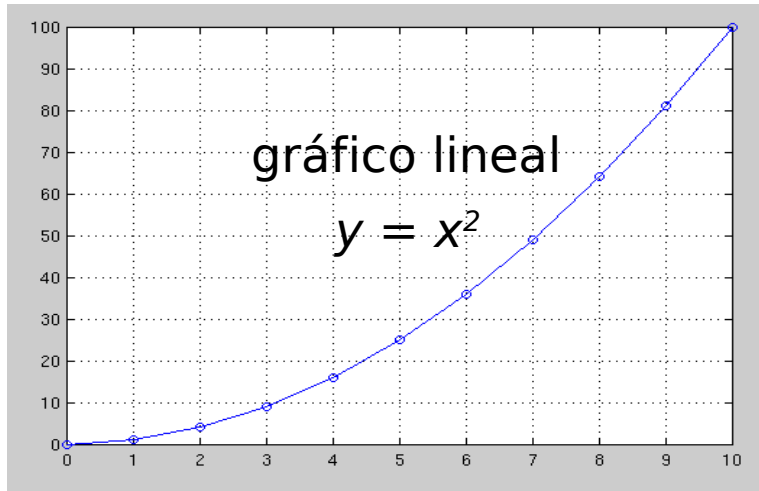
Outros comandos

- Engadir texto na figura: *text(x,y,'texto no gráfico en (x,y)');* *gtext('texto en posición indicada co rato');*

Gráficos logarítmicos

- Ás veces é bon usar unha escala logarítmica (isto é, representar $\log(x)$ no canto de y , ou o mesmo para y) xa que o rango de valores é moi grande.
- *semilogx(x, y)*: escala log só en x
- *semilogy(x, y)*: escala log só en y
- *loglog(x, y)*: escala log en x e y
- Os valores nulos ou negativos non se poden representar nestos gráficos.

Gráficos logarítmicos



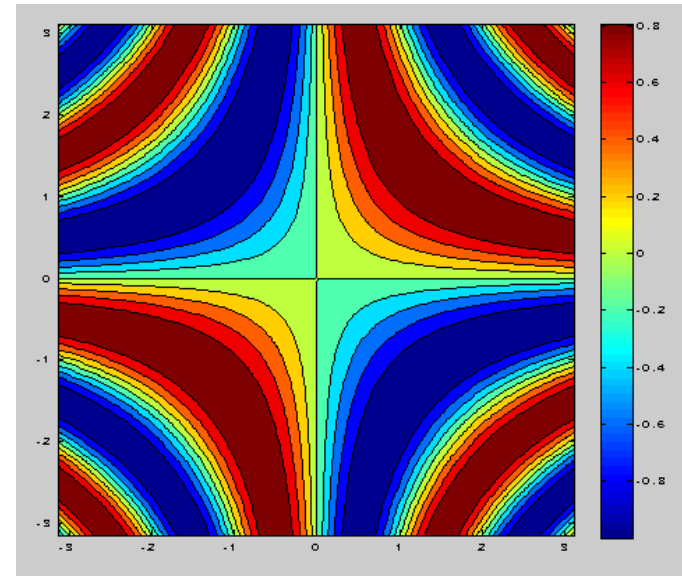
Mapa de calor

- Representa unha función $f(x,y)$ cun mapa de cores, que se pode cambiar coa función *colormap* (*help* para ver os mapas dispoñíbeis).

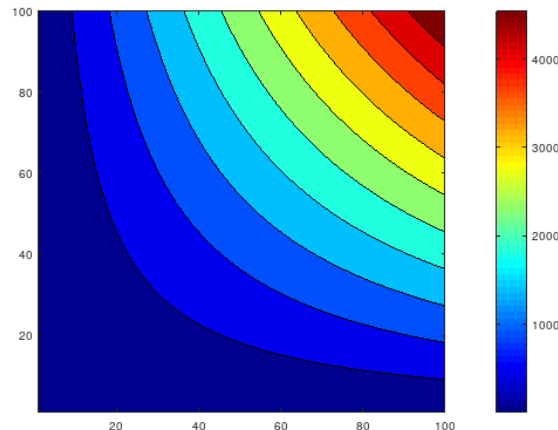
```
[x y]=meshgrid(-3:0.05:3);  
z=sin(x.*y);  
colormap jet  
contourf(x,y,z)  
colorbar
```

Selecciona o
mapa de cores

Engade a barra de
cores na dereita



- Tamén permite representar unha matriz:

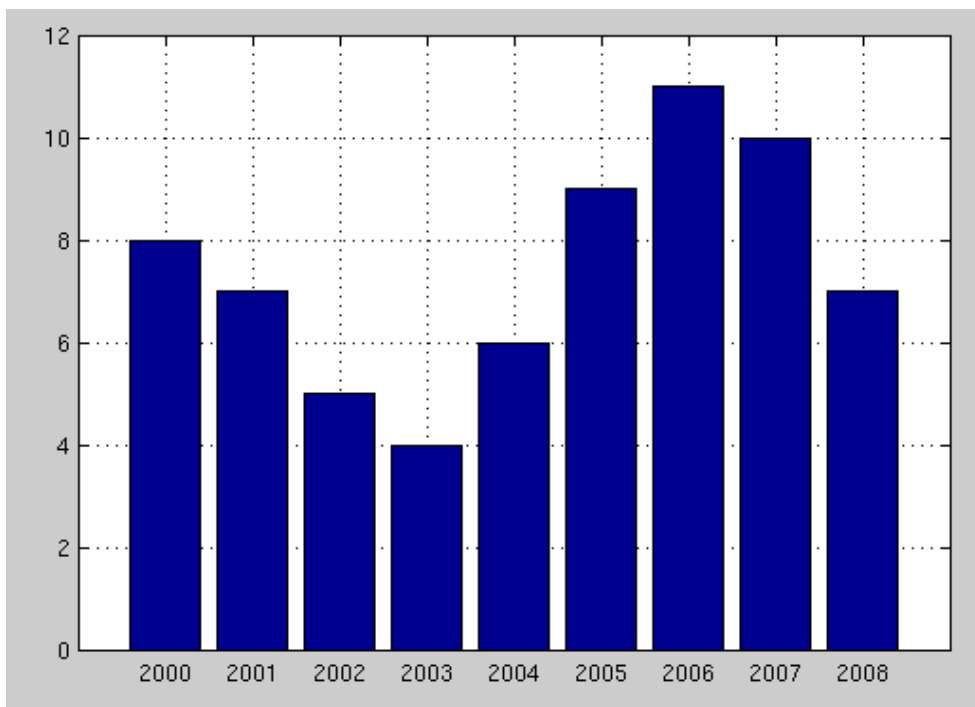


```
n=100;a=zeros(n);  
for i=1:n  
    for j=1:n  
        a(i,j)=i*j/2+1;  
    end  
end  
colormap jet;contourf(a);colorbar
```


Gráficos especiais

- Barras verticais:

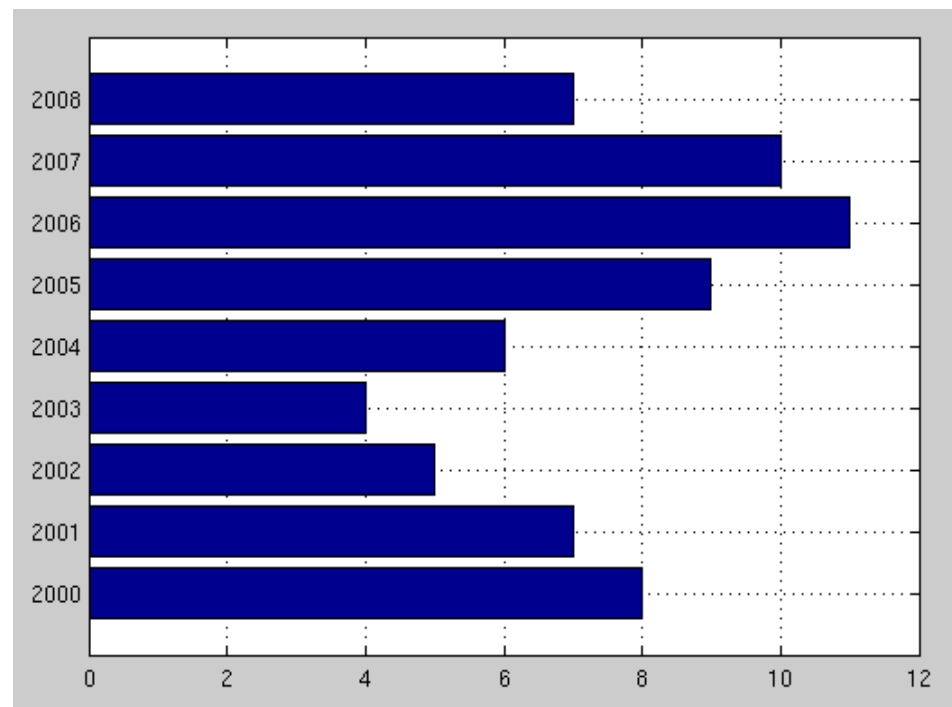
```
x = [2000:2008]; y = [8 7 5 4  
6 9 11 10 7];  
bar(x, y); % x crescente
```



Programación en Octave

- Barras horizontais:

```
x = [2000:2008]; y = [8 7  
5 4 6 9 11 10 7];  
barh(x, y); % x crescente
```

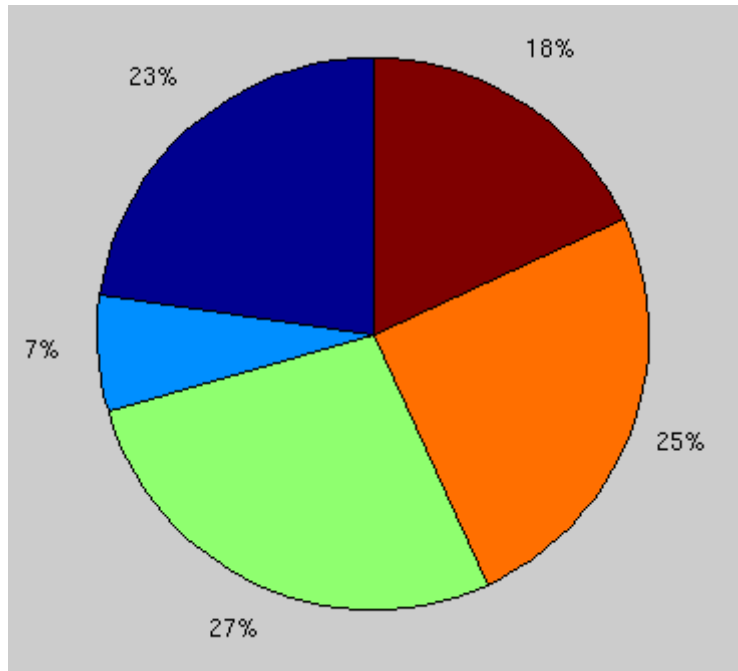


Gráficos 2D

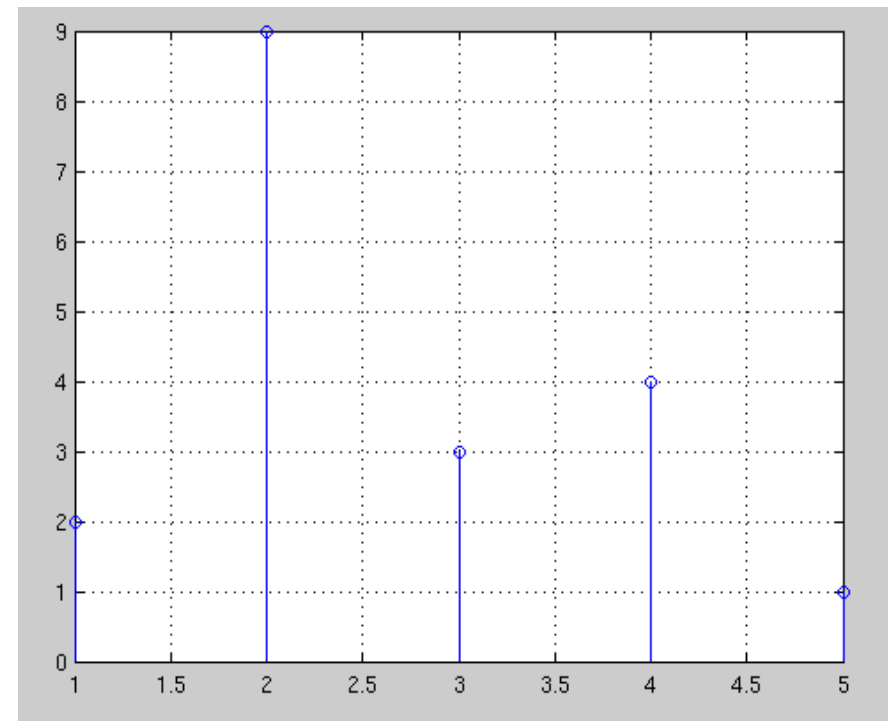
Gráficos especiais

- Tartas:

`x=[10 3 12 11 8];`
`pie(x)` % calcula os
porcentaxes; sentido
antihorario

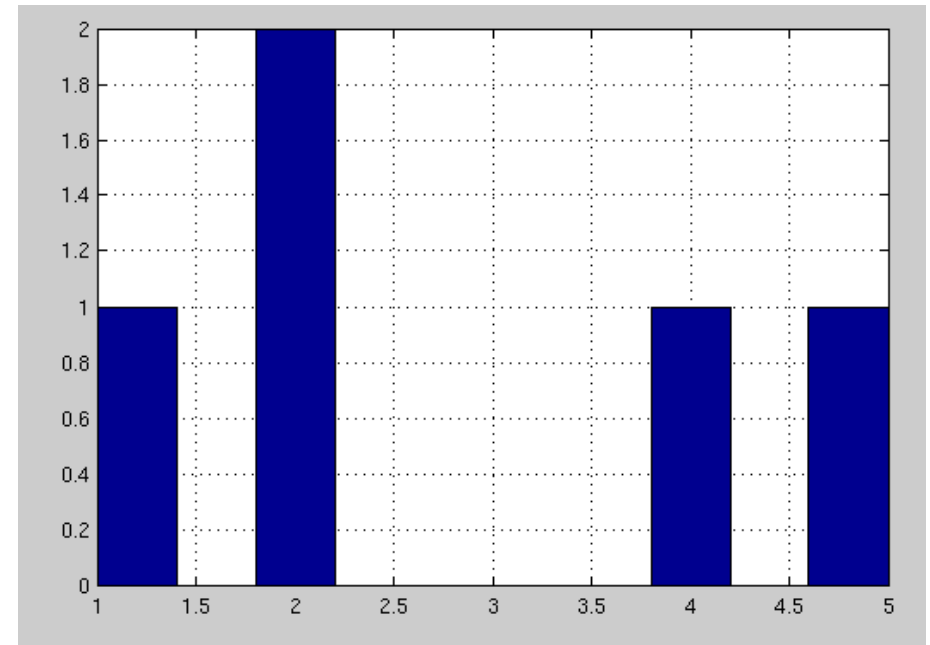


- Diagrama de troncos:
`x=[1:5];y=[2 9 3 4 1];`
`stem(x,y)` %x crescente



Histogramas (I)

- Cada barra vertical está asociada a un intervalo de los datos
- A altura de cada barra representa o nº de datos que se atopan no seu intervalo asociado
- Divide o rango dos elementos do vector en intervalos (10, por defecto)
- Ex: $y=[1 \ 2 \ 4 \ 5 \ 2];$
hist(y)
- Con nº de intervalos:
hist(y, 5)



Histogramas (II)

- Usando 2 vectores, para X e Y:

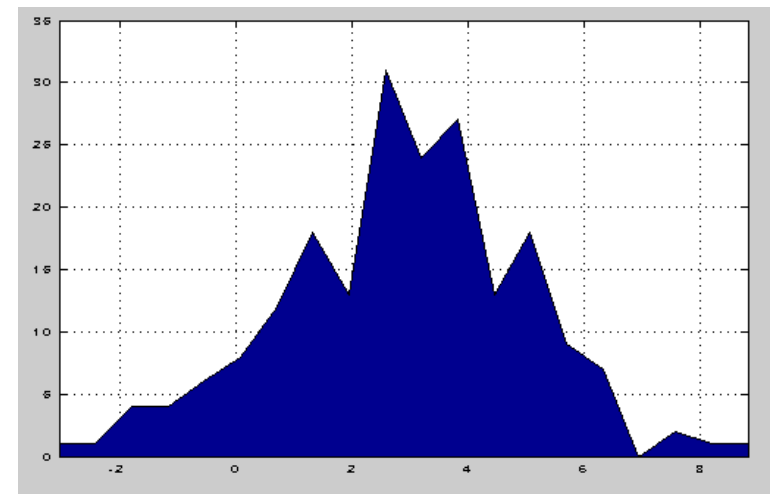
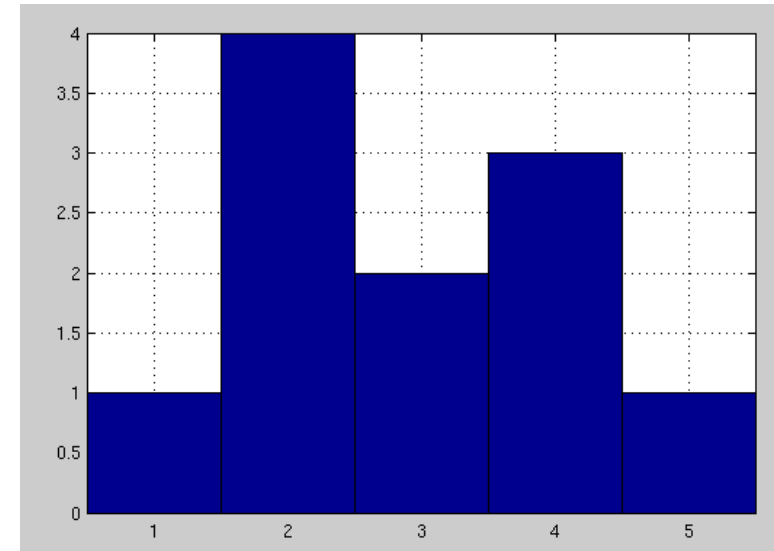
```
x=[1 2 3 4 5];y=[1 2 3 2 3 4 4 2 5 4 2];
```

```
hist(y, x)
```

- O 2º vector debe ser crecente (se non, erro)
- Para que o histograma sexa cunha liña continua:

```
med=3;desv=2;n=20;  
x=normrnd(med,desv,1,200);  
t=linspace(min(x),max(x),n);  
area(t,hist(x,n));grid on
```

Números aleatorios seguindo unha distribución gausiana

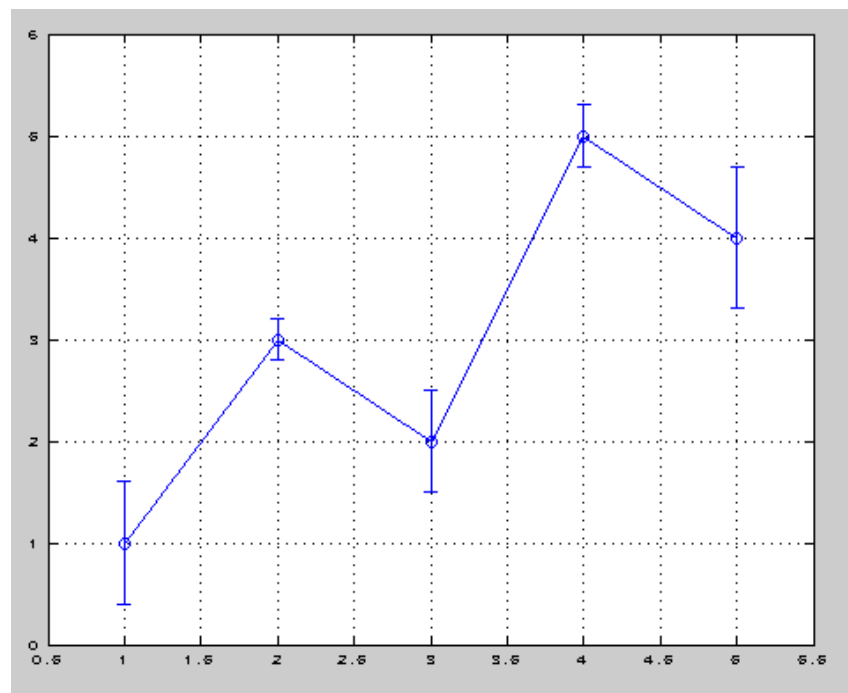


Gráficas con barras de erros

- Gráficas con barras de erros (p.e. para desviaciones típicas de valores promedio):

errorbar(datos, erro, opciones)

Ex: $x = [1 \ 3 \ 2 \ 5 \ 4];$
 $erro = [0.6 \ 0.2 \ 0.5 \ 0.3 \ 0.7];$
errorbar(x, erro, 'o-')



Curvas en coordenadas polares

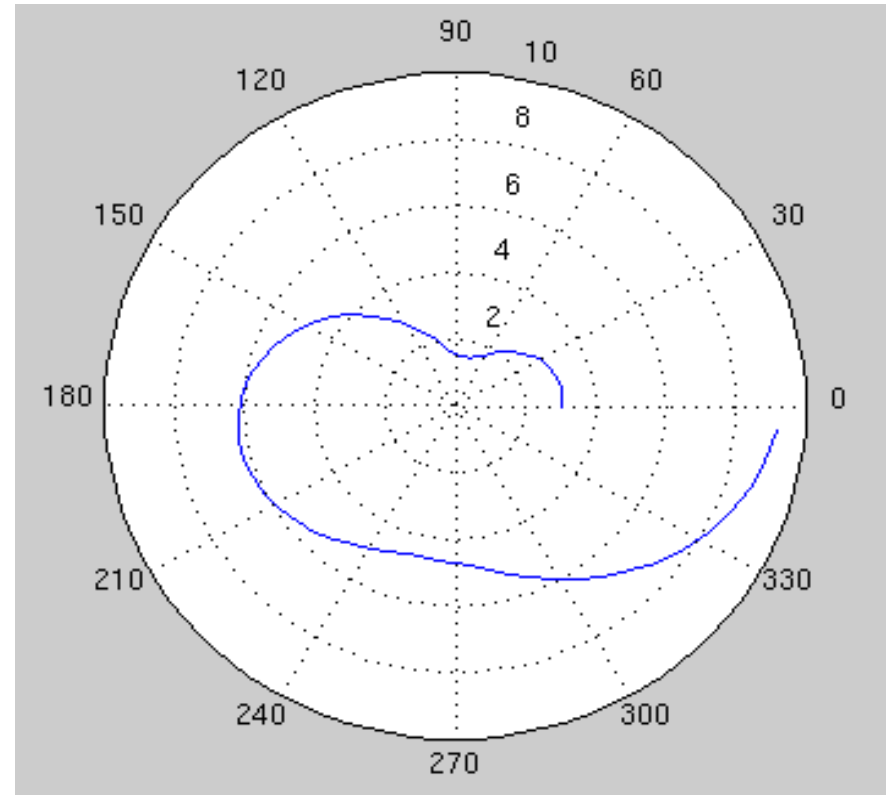
- Ecuación da curva en coordenadas polares: *radio* = *radio(ángulo)*: $\rho = \rho(\theta)$

- Ex: *polar(ángulo, radio)*

ángulo: vector cos valores do ángulo; *radio*: vector cos valores do radio

```
t=0:0.1:2*pi;
```

```
r=3*cos(t).^2+t;polar(t,r)
```



- Función *ezpolar*($\rho(\theta)$, [θ_{min} θ_{max}])

ezpolar('3*cos(t)^2 + t', [0 2*pi])

Máis funcións gráficas

- Gráficas para matrices:

image(a): non escala

imagesc(a): escala

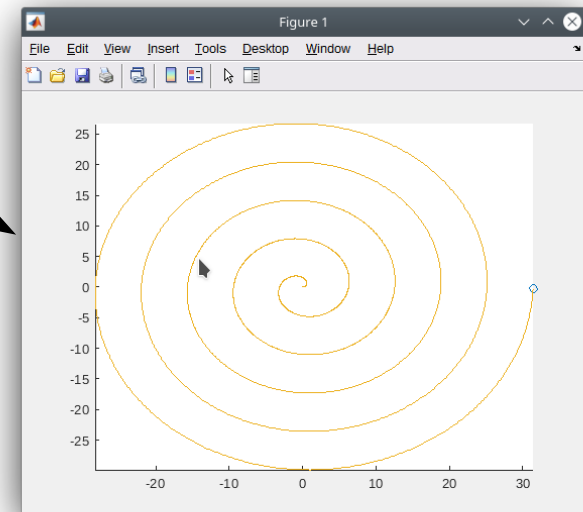
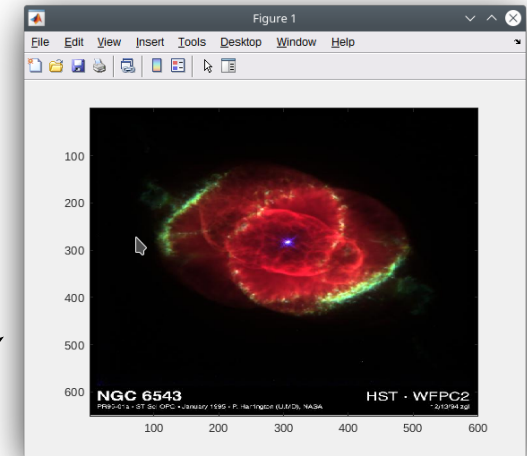
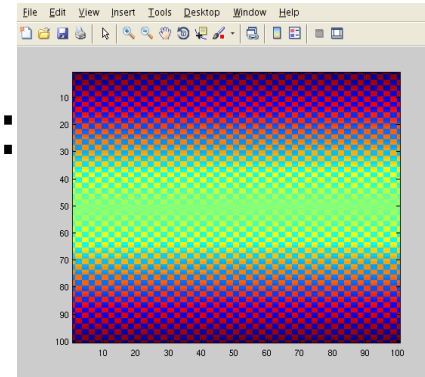
pcolor(a)

- Ex: *a=magic(10); image(a)*

x=imread('ngc6543a.jpg');image(x)

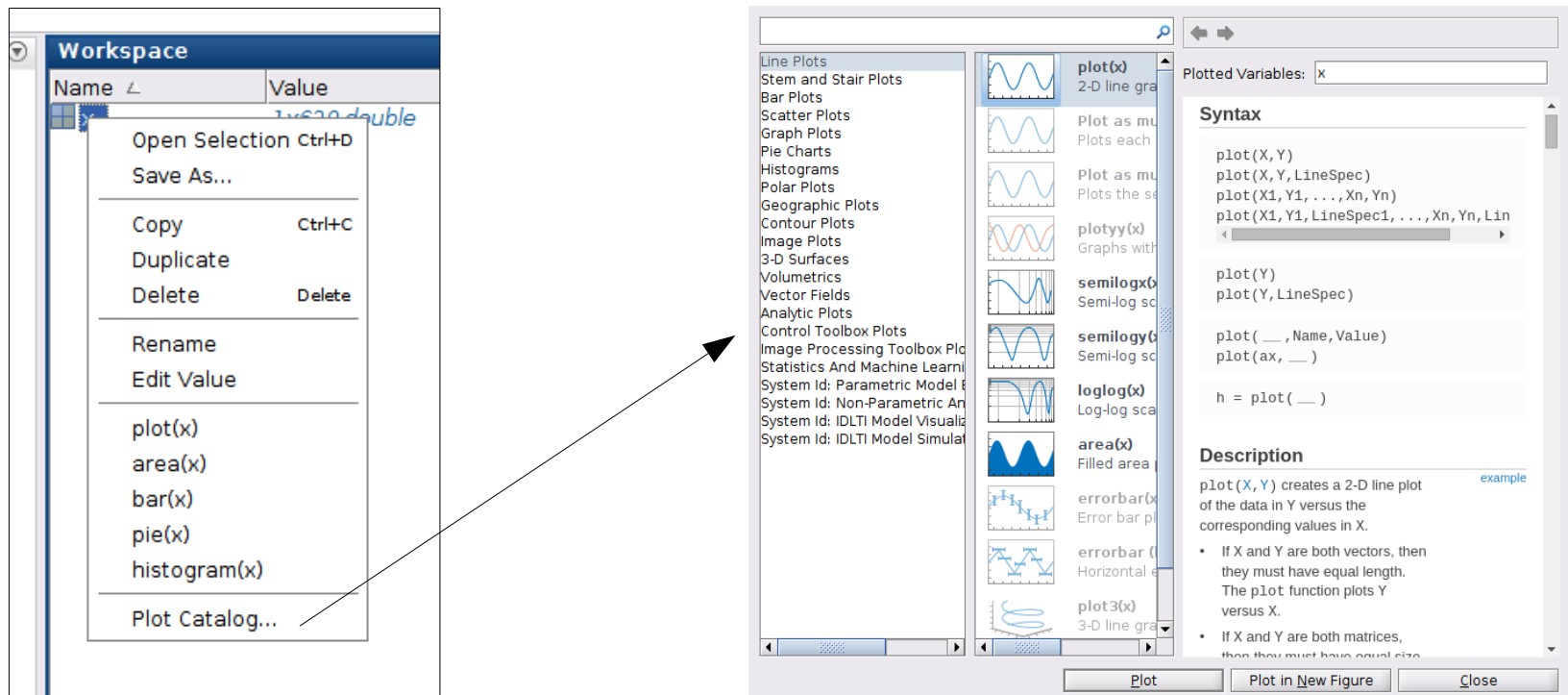
- Animacións: *comet(x)*, *comet(x,y)*

- Ex: *t=0:0.01:10*pi;*
*x=t.*cos(t);y=t.*sin(t);*
comet(x,y)

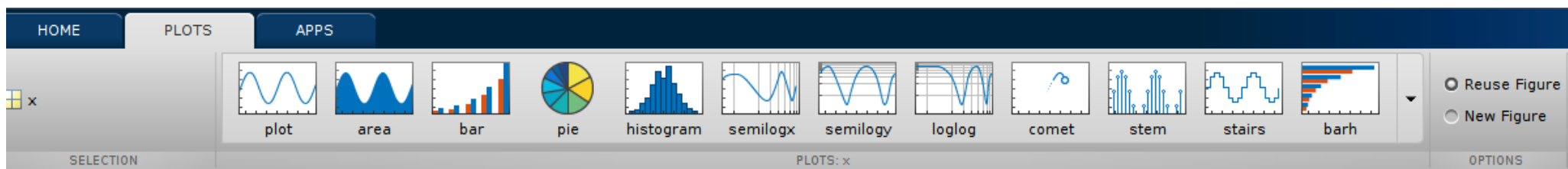


Asistente para gráficos

- Na ventá *workspace* pódese seleccionar un vector ou matriz e o tipo de gráfico para representalo



- Tamén na barra de menú *Plots*



Exercicios

Representa as seguintes curvas e superficies:

1) Espiral de Arquímedes (polares): $r(\theta) = a\theta$

2) Bruxa de Agnesi: $y = \frac{8a^3}{x^2 + 4a^2}$

3) $y = f(x) = \tan x$, en $[-\pi/2, \pi/2]$ (usa `ezplot()`)

4) $y = f(x) = e^{-x/2} \sin 20x$, en $[0, 10]$ (usa `fplot()`)

5) $y = \frac{x^2 - x + 1}{x^2 + x + 1}$, $x \in [-10, 10]$

6) Representa os datos da táboa xunto coa función que os modela

$$y = 320 \left[\left(\frac{x}{210} \right)^{0.16} + 1 \right]$$

x	y
7E-5	345
2E-4	362
0.05	419
0.8	454
4.2	485
215	633
3500	831

Soluciones aos exercicios

1) `t = 0:0.1:pi; r = t; polar(t, r);` (supoño $a = 1$)

2) (supoño $a = 1$) `fplot('8/(x^2 + 4)', [-10, 10])`

3) `ezplot('tan x', [-pi/2, pi/2])`

4) `fplot('exp(-x/2)*sin(20*x)', [0, 10])`

5) `fplot('(x^2 - x + 1)/(x^2 + x + 1)', [-10, 10])`

6) `a=load('datos.dat');`
`s = 320*((a(:, 1)/210).^0.16`
`+ 1);`
`semilogx(a(:, 1), a(:, 2), 'bo-',`
`a(:, 1), s, 'gs-')`

