



CURSO DE SCILAB

INTRODUCCIÓN

PARTE II

Elaborado por: Prof. Enrique Eduardo Gavorskis Souto

e.gavorskis@hotmail.es

Revisado por: Prof. Francisco M. Gonzalez-Longatt

fglongatt@ieee.org



CONTENIDO

- El programa
- Entornos de Trabajo
- Uso del Help
- Operaciones con Vectores y Matrices
- Operaciones con Funciones
- **Graficación**
- **Programación**
- **Debugger**

GRÁFICOS: 2D Y 3D

SCILAB dispone de varias funciones básicas para crear gráficos 2-D. Pero la principal función es la siguiente:

- **plot2d(x,y)** crea un gráfico a partir de vectores y/o columnas de matrices, con escalas sobre ambos ejes.
- **style** Esta opción sirve para definir como será dibujada la curva.
- **logflag** Dicha opción se emplea para seleccionar el tipo de escala sobre los ejes.
- **rect** Esta opción puede ser usada para fijar los rangos de los ejes mínimos requeridos para la gráfica. Si esta opción es especificada, los valores asociados deberán ser dados como un vector con números reales de cuatro entradas: [xmin,ymin,xmax,ymax].
- **frameflag** Esta opción puede ser usada para controlar el cómputo actual de los rangos de las coordenadas para los valores mínimos requeridos. Los rangos en uso pueden tener requerimientos mayores o menores.
- **nax, leg**

STYLE

y	yellow	.	point	-	solid
m	magenta	o	circle	--	dashed
c	cyan	x	x-mark	-.	dashdot
r	red	+	plus	p	pentagram
g	green	*	star	^	triangle (up)
b	blue	s	square	<	triangle (left)
w	white	d	diamond	>	triangle (right)
k	black	:	dotted	v	triangle (down)
		h	hexagram		

GRÁFICOS: 2D Y 3D

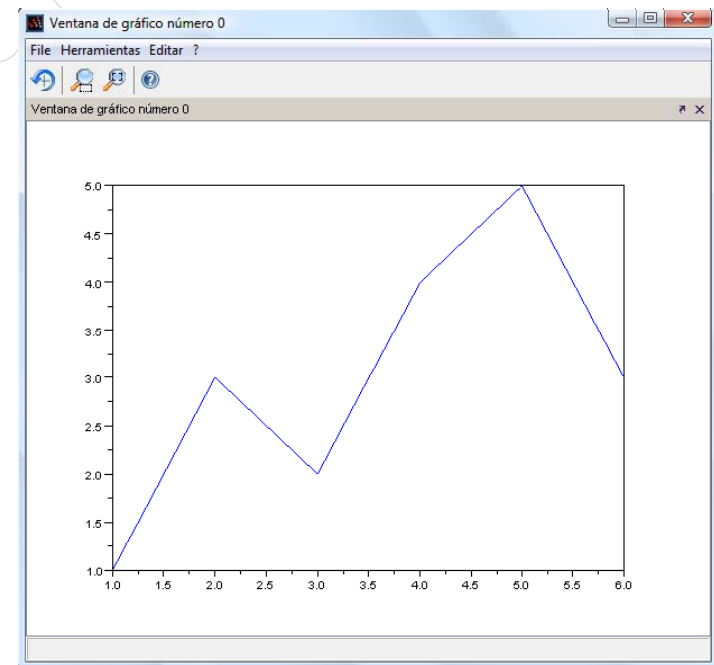
Existen funciones orientadas a añadir títulos al gráfico, a los ejes, a dibujar una cuadrícula auxiliar, etc.

- `title('título')` añade un título al dibujo
- `xlabel('Nombre del Eje x')` añade una etiqueta al eje de abscisas.
- `ylabel('Nombre del Eje y')` idem al eje de ordenadas.
- `xgrid` activa una cuadrícula en el dibujo.

GRÁFICOS: 2D Y 3D

plot es la función clave de todos los gráficos 2-D en MATLAB. Ya se ha dicho que el elemento básico de los gráficos bidimensionales es el **vector**.

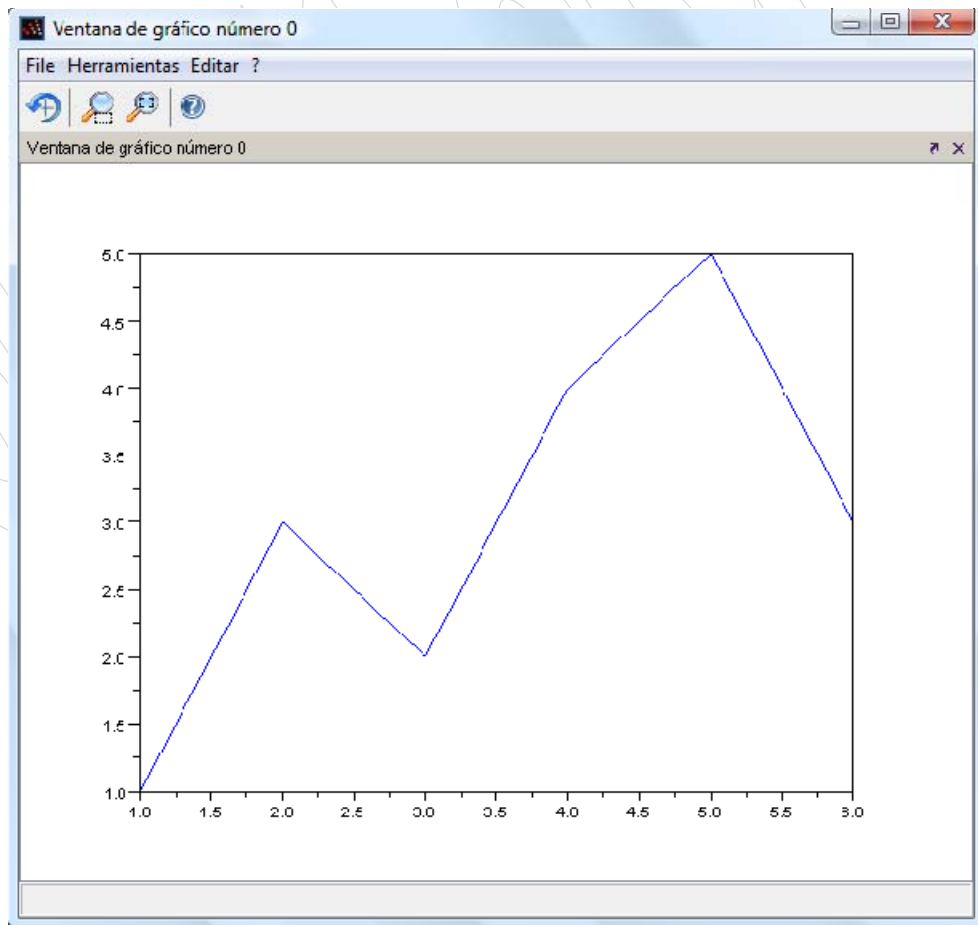
```
» x=[1 3 2 4 5 3]
x =
1 3 2 4 5 3
» plot(x)
```



GRÁFICOS: 2D Y 3D

La función `plot()`, no hace otra cosa que dibujar vectores.

```
» x=[1 3 2 4 5 3]
x =
1 3 2 4 5 3
» plot(x)
```

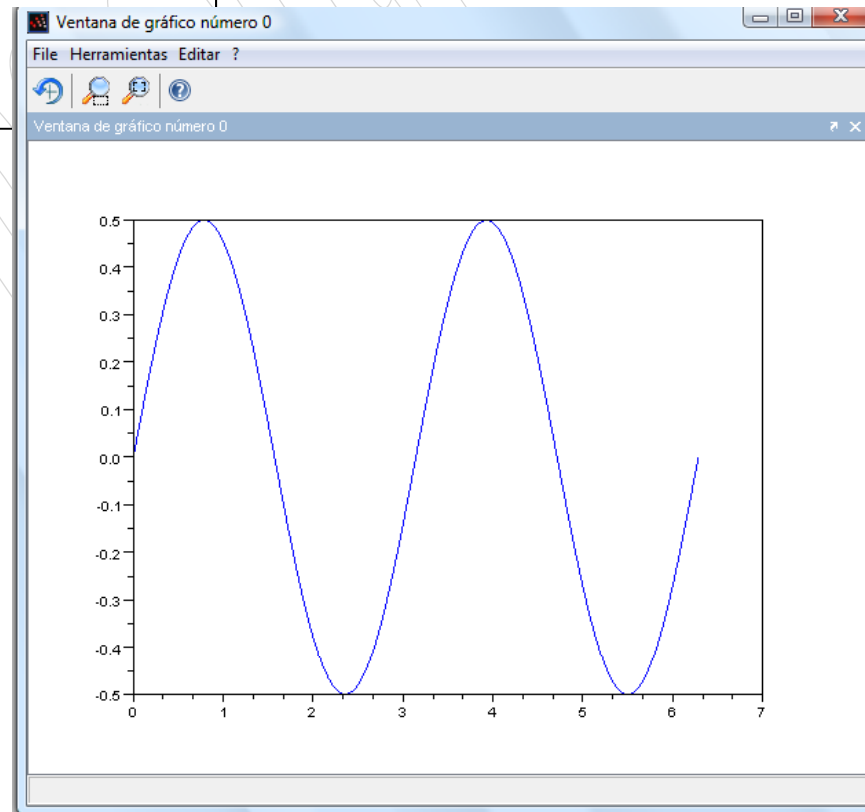


COMANDO PLOT

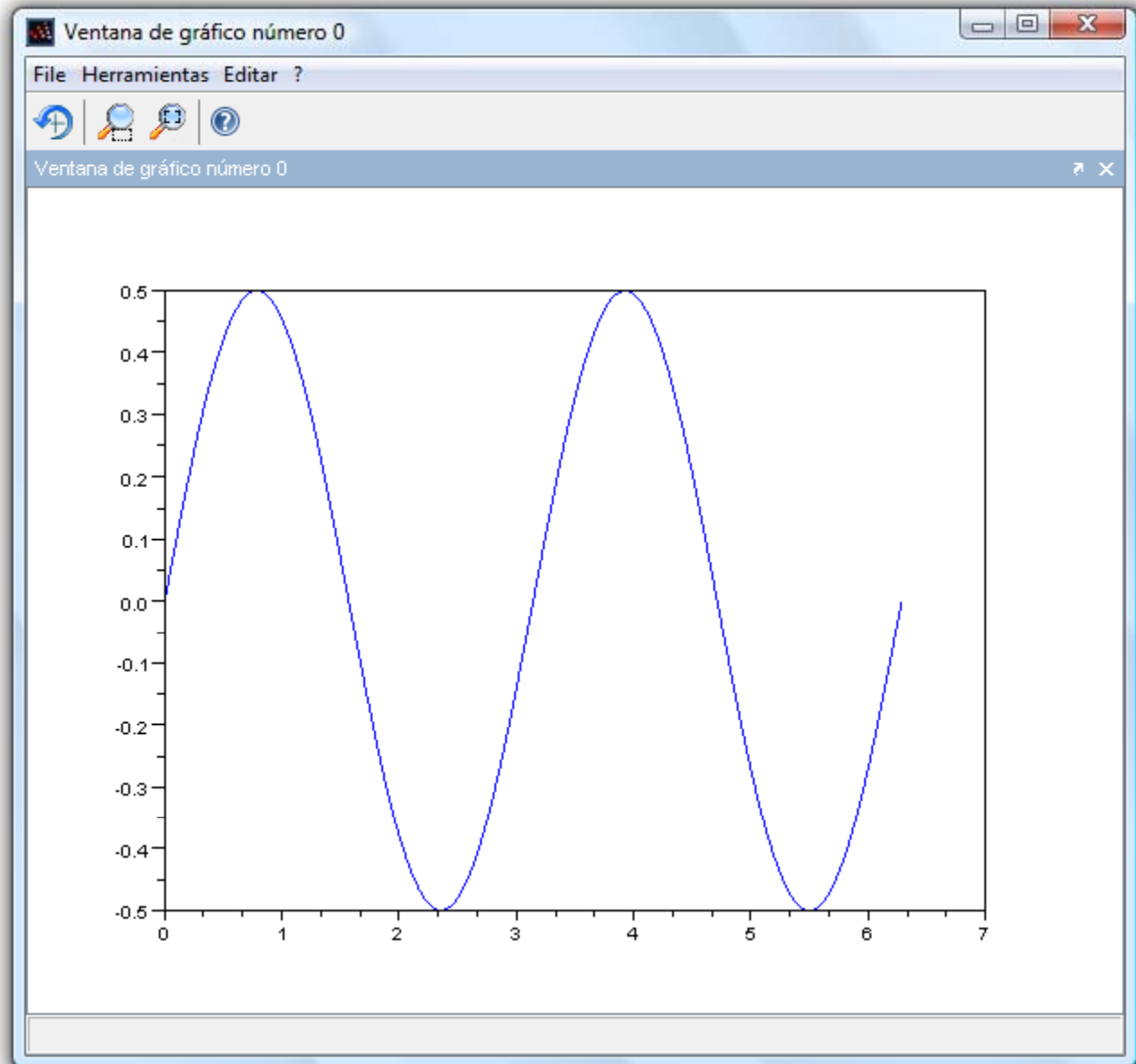
- Es el comando más utilizado para gráficos en 2D
- Representa gráficamente conjuntos de *arrays* de datos:
 - Elige automáticamente los ejes apropiados
 - Por defecto, conecta los datos mediante líneas rectas

EJEMPLO

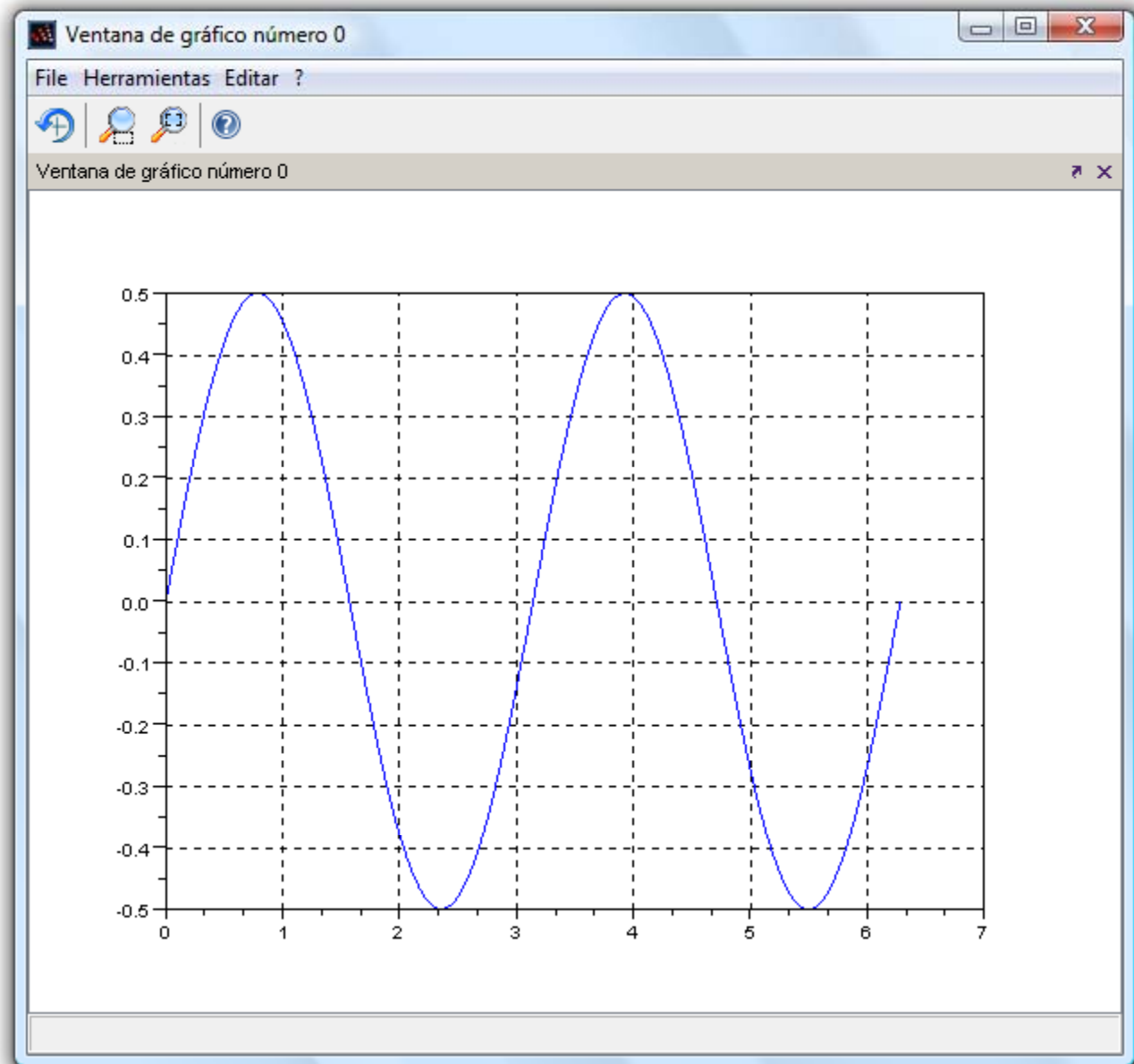
```
» x=0:%pi/90:2*%pi;  
» y=sin(x).*cos(x);  
» plot(x,y)  
  
» xgrid  
» xlabel('eje x (en radianes)')  
» ylabel('eje y')  
» title('y=sen(x)*cos(x)')
```



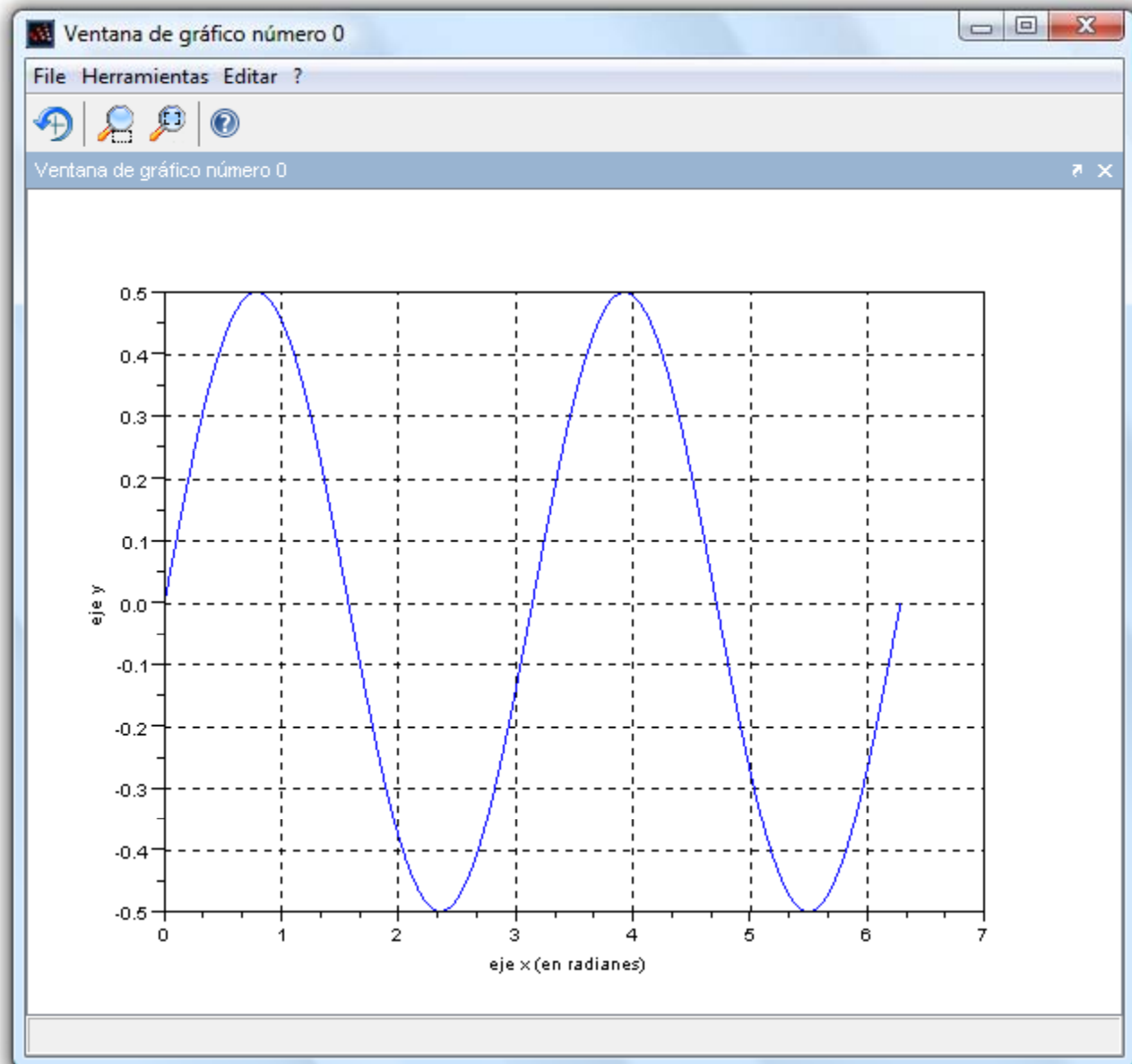
```
-->x=0:%pi/90:2*%pi;  
-->y=sin(x).*cos(x);  
-->plot(x,y)  
-->
```



```
-->x=0:pi/90:2*pi;  
-->y=sin(x).*cos(x);  
-->plot(x,y)  
-->xgrid  
-->
```



```
-->x=0:%pi/90:2*%pi;  
-->y=sin(x).*cos(x);  
-->plot(x,y)  
-->xgrid  
-->xlabel('eje x (en radianes)')  
-->ylabel('eje y')  
-->
```



OTROS COMANDOS UTILES

Determina las coordenadas del punto en donde se ubique el puntero del mouse

• » **xgetmouse**

Calcular las coordenadas de puntos sobre la curva

• » **xclick**

Insertar textos en una figura

• » **xstring()**

Escala isométricas

• » **isoview()**

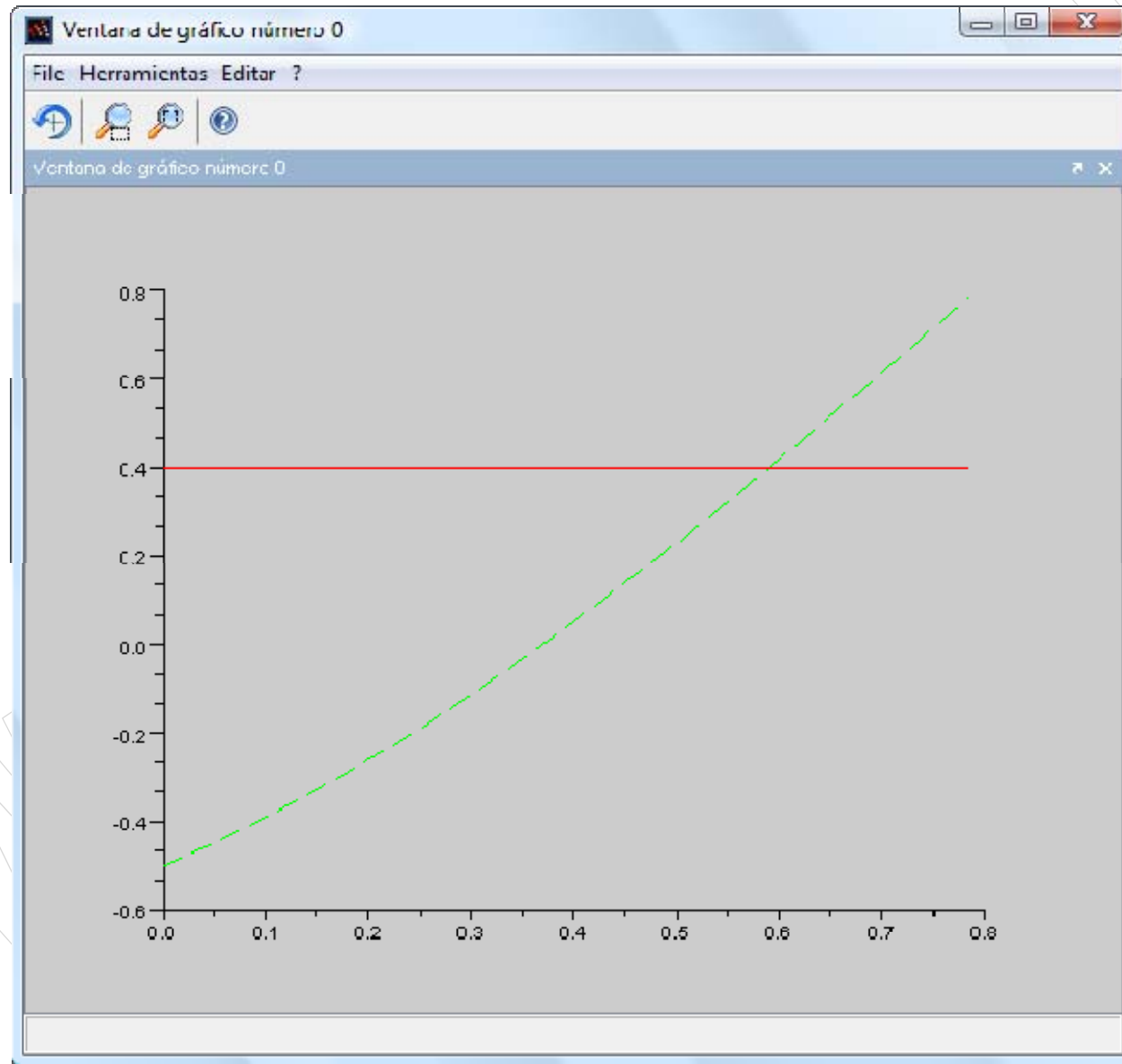
EJEMPLO

Calcular gráficamente las soluciones de la ecuación

$$\frac{2x - \cos(2x)}{2} = 0.4$$

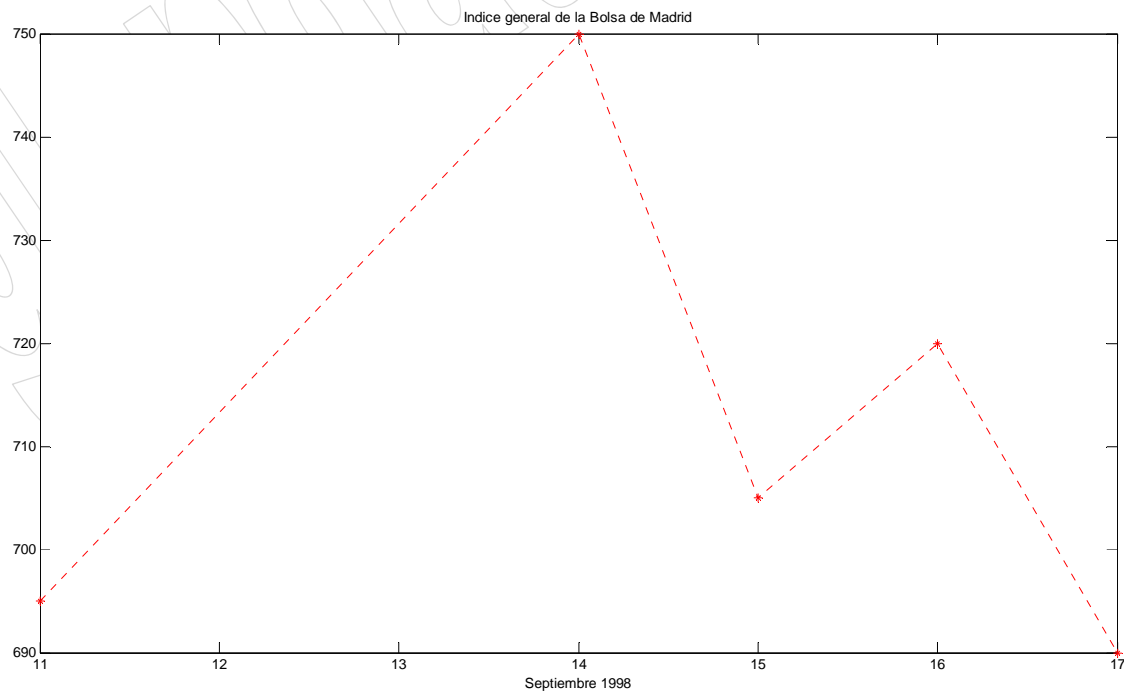
```
» xi=0;h=%pi/360;xf=%pi/4;
» teta=xi:h:xf;
» f1=(2*teta-cos(2*teta))/2;
» f2=0.4*ones(sin(f1));
» figure
» plot(teta,f1,'g--',teta,f2,'r')
» xlabel('Ángulo (radianes)')
» xclick
» xstring(0.2,0.43,'2x-cos(2x))/2')
» // Cambie los valores de 0.2 y 0.43
» // por las dos ultimas cifras que arroja xclick
» title('Raiz aproximada')
```

EJEMPLO



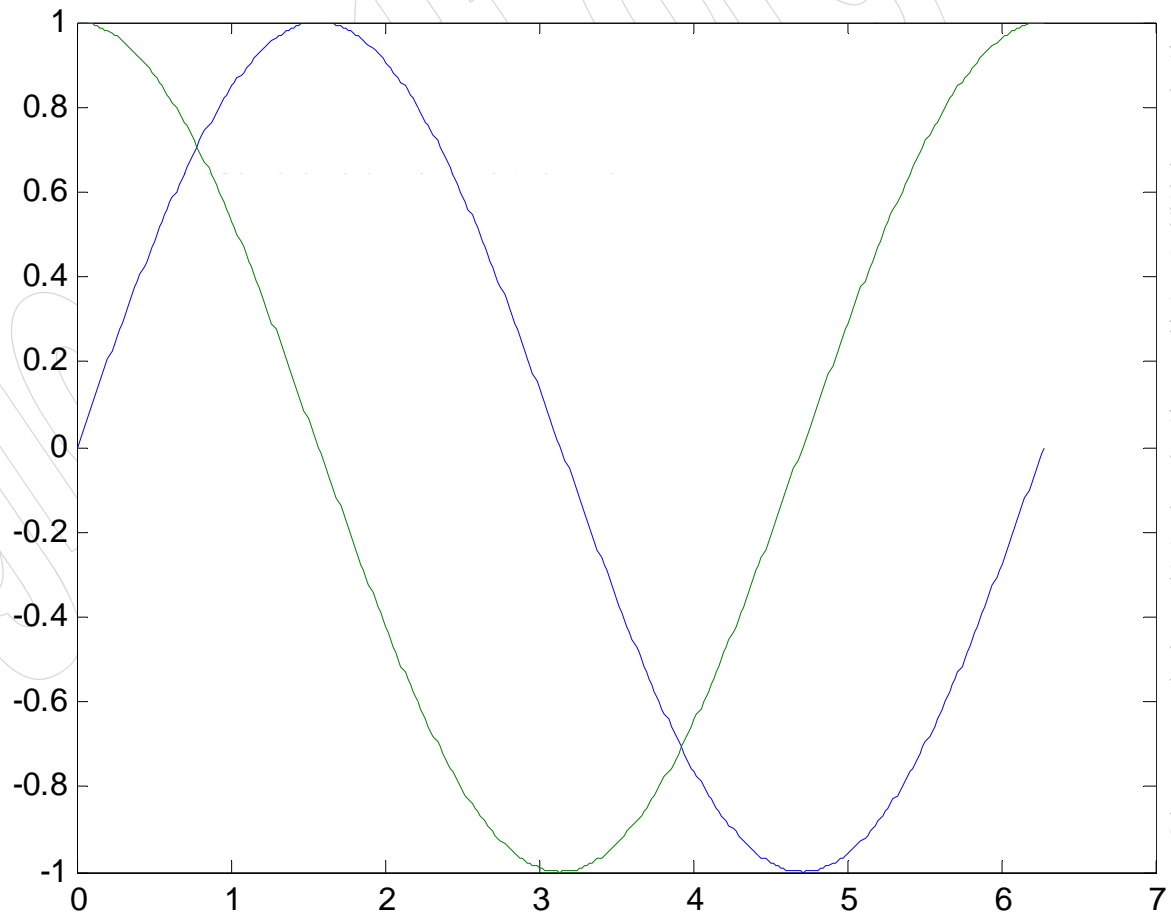
EJEMPLOS

```
x = [11 14 15 16 17];  
y = [695 750 705 720 690];  
plot(x,y,'r*:') ,  
title('Indice general de la Bolsa de Madrid'),  
xlabel('Septiembre 1998')
```



EJEMPLO

```
» x=0:%pi/180:2*%pi;  
» y=sin(x);  
» z=cos(x);  
» plot(x,y,x,z)  
» A=[y' z']  
» plot(x,A)
```



GRÁFICAS EN 2D Y 3D

- Función `eval` se utiliza con funciones definidas con un carácter. `y = eval('carácter')`

```
» f= 'sin(x)-2*cos(x)';  
» x=0:%pi/90:2*%pi;  
» y=eval(f);  
» plot2d(x,y,nax=[0,6,0,2.4],rect=[0,0,6,2.4])  
» [b,xb,yb]= xclick();  
» xstring(xb,yb,' sen(x)-2cos(x) ')
```

COMANDOS UTILES

- Llamar una nueva figura **figure** o para referirse a una figura ya hecha **figure(n)**
- Borrar la figura actual **clf**
- **close all** borra todas las figuras.
close (figure(n)) borra la figura n

COMANDOS UTILES

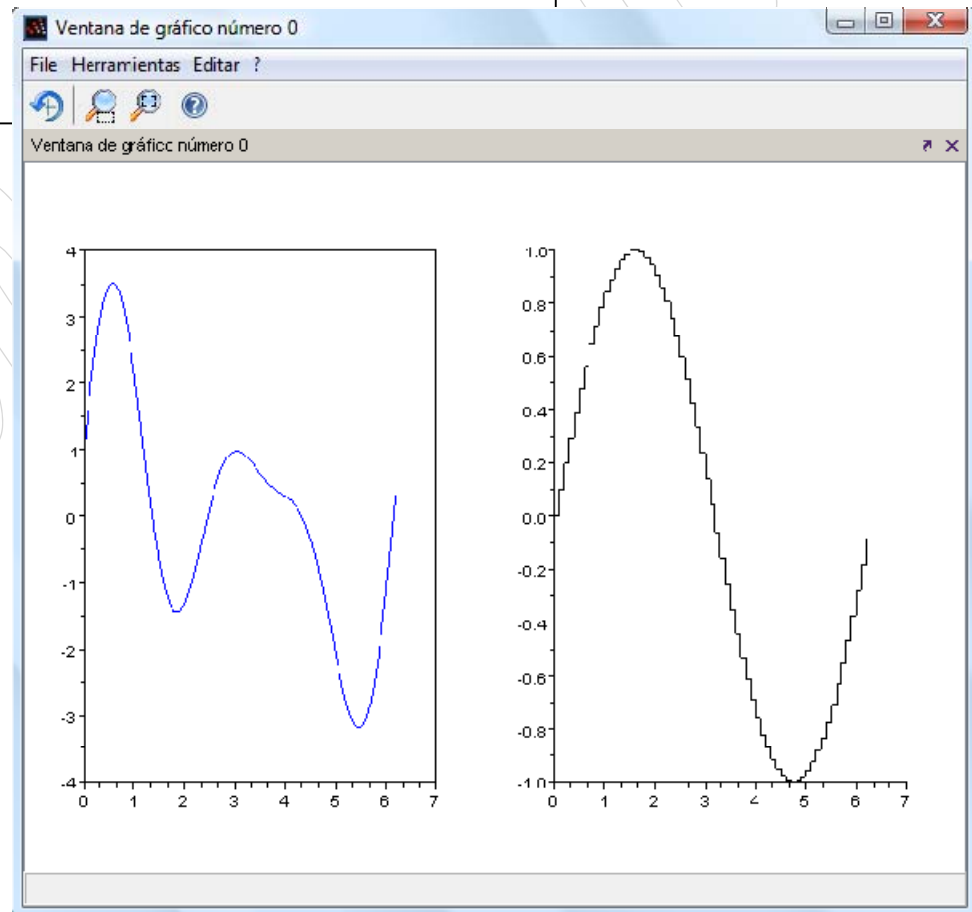
- Una ventana gráfica se puede dividir en m particiones horizontales y n verticales, con objeto de representar múltiples gráficos en ella.
- Cada una de estas subventanas tiene sus propios ejes, aunque otras propiedades son comunes a toda la figura.
- La forma general de este comando es:
subplot(m,n,i)

COMANDOS UTILES

- **subplot(m,n,i)** donde **m** y **n** son el número de subdivisiones en filas y columnas, e **i** es la subdivisión que se convierte en activa. Las subdivisiones se numeran consecutivamente empezando por las de la primera fila, siguiendo por las de la segunda, etc.

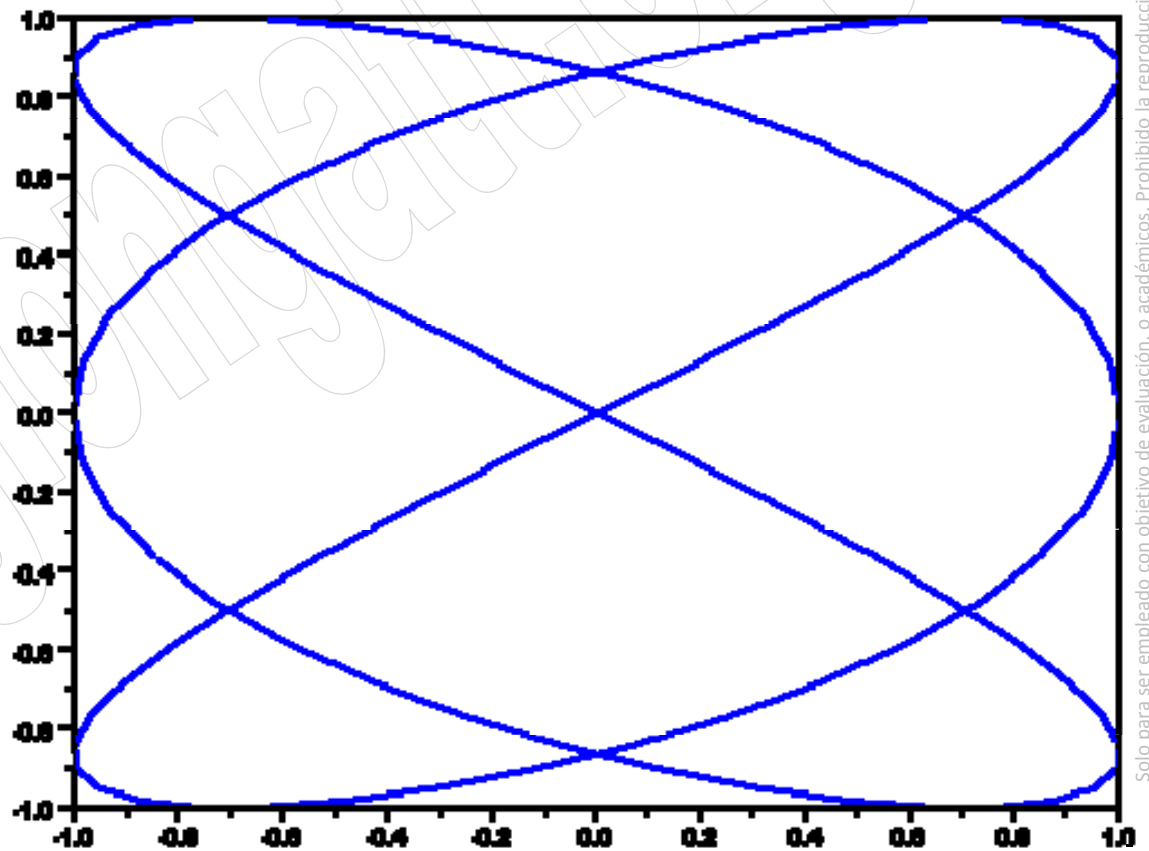
EJEMPLO

```
» x=[0:0.1:2*%pi];  
» subplot(121)  
» plot(x,sin(x)+2*sin(2*x+0.5)+sin(3*x))  
» subplot(122)  
» plot2d2(x,sin(x))
```



CURVAS DE LISAJOUX

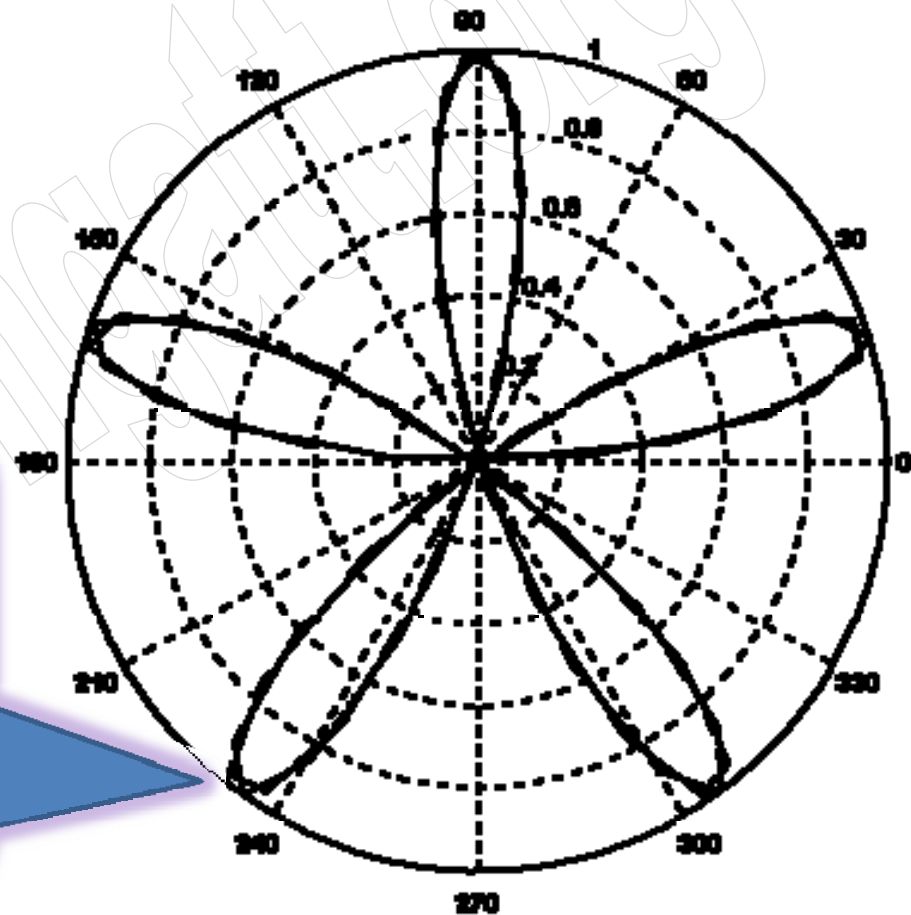
```
t = 0:2*pi/100:2*pi;  
x =sin(2*t);  
y =sin(3*t);  
plot(y,x)
```



DIAGRAMAS DE RADIACIÓN DE ANTENAS

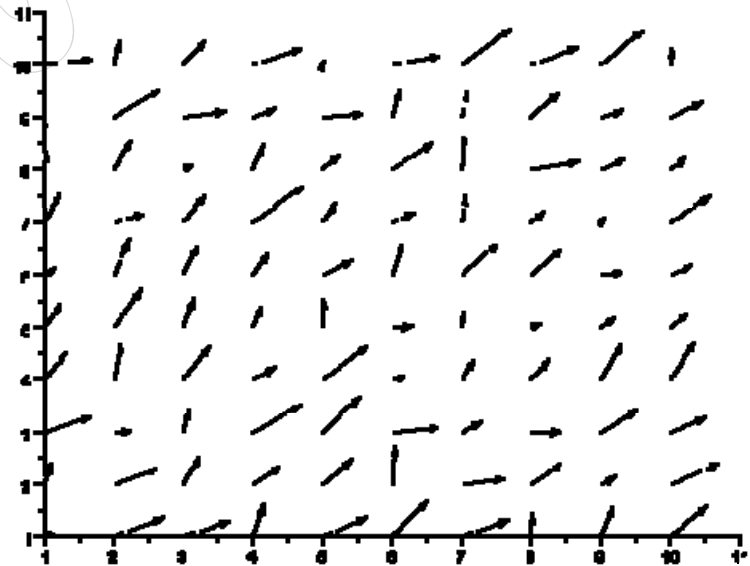
```
z = 0:0.1:2*%pi;  
r = sin(5*z);  
polarplot(z,r)
```

NOTEN QUE EN EL TRAZO DE LA CURVA PRESENTA IRREGULARIDADES, ELLO PUEDE OCURRIR SI EL NÚMERO DE PUNTOS A GRAFICAR SON RELATIVAMENTE POCOS, PARA ESTE EJEMPLO ELLO SE PUEDE ARREGLAR DISMINUYENDO EL PASO DE 0.1 A 0.01



OTRAS FUNCIONES EN 2D

- **bar()** crea diagramas de barras.
- **barh()** diagramas de barras horizontales.
- **pie()** gráficos con forma de “torta”.
- **champ()** graficas de campos vectoriales
- **fchamp()** campo vectorial (2D) correspondiente una EDO de segundo orden.



GRÁFICOS: 2D Y 3D

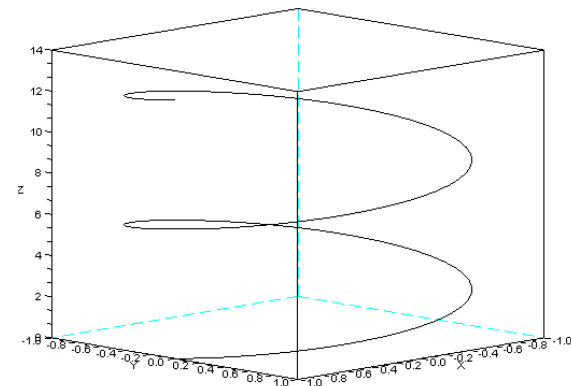
Funciones gráficas 3D elementales:

La función `plot3d` es análoga a su homóloga bidimensional `plot`. Su forma más sencilla es la siguiente:

» `plot3d(x,y,z)`

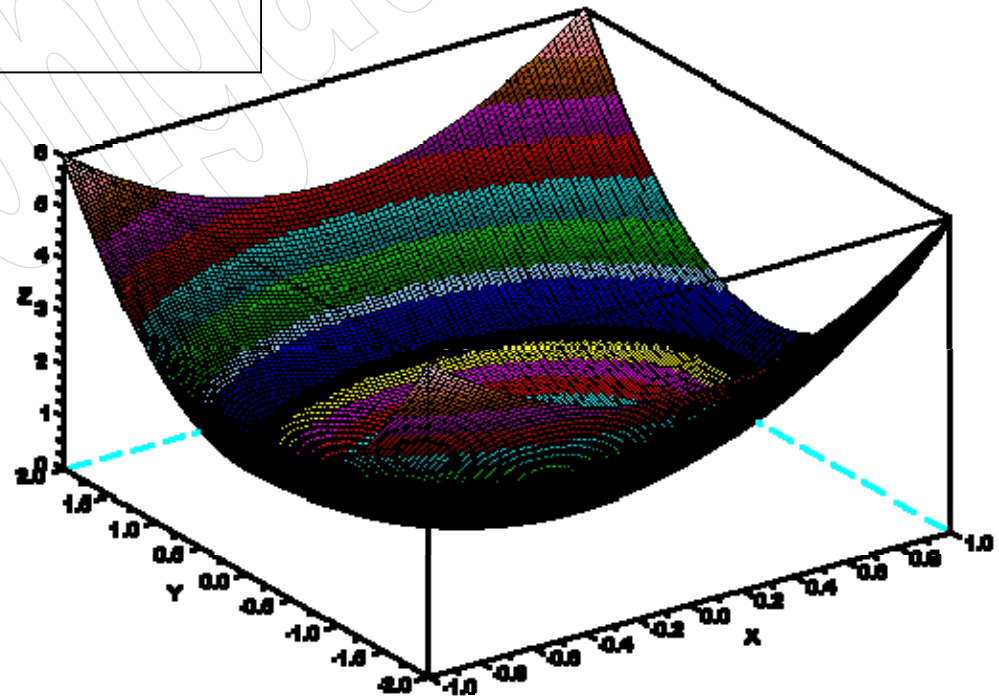
Aunque también existen otros comando de graficación en 3d como `param3d`

```
» t=linspace(0,4*pi,100);  
» param3d(cos(t),sin(t),t)
```



SURF

```
function z=f(x,y)
z=2*x^2+y^2;
endfunction
x=linspace(-1,1,100);
y=linspace(-2,2,200);
z=(feval(x,y,f))';
clf
surf(x,y,z)
```



GRÁFICOS: 2D Y 3D

Representación gráfica de superficies. `mesh(x,y,Z)`,

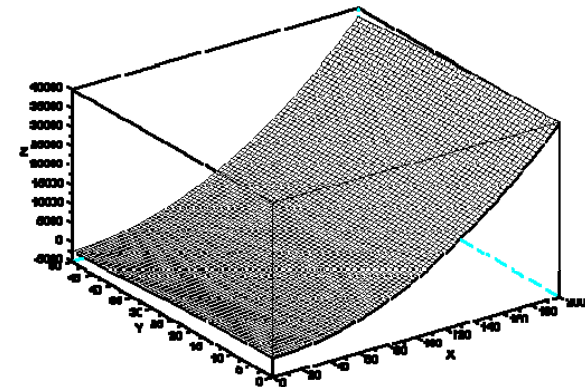
Creación de una malla `[X, Y]=meshgrid(x,y)`

Gráfica de la malla construida sobre la superficie z

`mesh(X,Y,Z)`

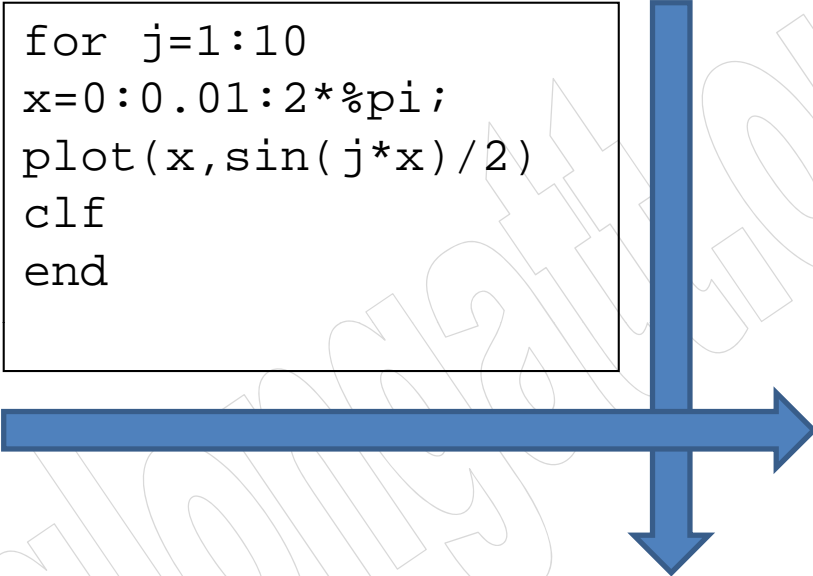
Otros graficados `surf`, `plot3d1`

```
» x=[0:2:200];y=[0:50];  
» // Obtenemos la malla del dominio  
» [X Y]=meshgrid(x,y);  
» length(x),length(y)  
» size(X), size(Y)  
» Z=X.^2-Y.^2;  
» figure(1);mesh(X,Y,Z)  
» figure(2);surf(X,Y,Z)  
» figure(3);plot3d1(X,Y,Z)
```



VIDEOS EN SCILAB

```
for j=1:10
x=0:0.01:2*%pi;
plot(x,sin(j*x)/2)
clf
end
```





PROGRAMACIÓN EN SCILAB

INTRODUCCIÓN



SCE - FILES

¿Qué son?

¿Para qué sirven?

Tipos de archivos sci

Características de funciones

Ejemplo de función

Pasos que sigue SCILAB

¿QUÉ SON LOS SCE-FILES?

- SCILAB permite crear funciones nuevas en forma de archivos con extensión ***.sce** y almacenarlos
- Un archivo ***.sce** es una secuencia de órdenes de SCILAB que puede contener, incluso, referencias a otros archivo ***.sce**
- Los archivo ***.sce** son textos ASCII creados con cualquier editor o procesador de texto

¿QUÉ SON LOS SCE-FILES?



¿PARA QUÉ SIRVEN?

Automatizar secuencias de órdenes que se utilizan de forma repetitiva.

Proporcionar extensibilidad a SCILAB con la posibilidad de añadir nuevas funciones cuya utilización no difiere de las que incluye originalmente.

TIPOS DE SCE - FILES

- Archivos predefinidos:
 - ✓ Seno, Coseno, Tangente.
- Archivos propios:
 - ✓ Son un compendio de funciones predefinidas ya sea matrices, vectores, senos, cosenos, etc. que generan un programa nuevo y específico

CARACTERÍSTICAS DE FUNCIONES

El nombre de la función y del archivo debe ser el mismo

Esta se ejecuta desde el entorno de SCILAB por primera vez.

Son capaces de generar programas emergentes y trabajar en un entorno fuera del SCILAB, pero los cálculos siguen siendo ejecutados dentro del SCILAB.

ADQUISICIÓN DE DATOS

```
» // Ubica, extrae y/o abre el documento de Excel
» [fd, SST, Sheetnames, Sheetpos] =
xls_open('C:\...\Nombre_del_Archivo.xls');
» // Lee la primera hoja de cálculo
» [Value, TextInd] = xls_read(fd, Sheetpos(1));
» // Cierra el documento
» fclose(fd)
» Value
```

EXISTEN OTRAS FORMAS DE IMPORTAR DATOS DE EXCEL, MEDIANTE OTROS COMANDOS COMO POR EJEMPLO `readxls`.

CLARO QUE TAMBIÉN ES POSIBLE LEER DATOS DE OTROS TIPOS DE ARCHIVOS, COMO POR EJEMPLO ARCHIVOS DE SONIDO `.WAV` MEDIANTE EL COMANDO `wavread`

COMANDOS `if`, `elseif` y `else`

```
» i=2
» for j = 1:3,
» if i == j then
» a(i,j) = 2;
» elseif abs(i-j) == 1
» then a(i,j) = -1;
» else a(i,j) = 0;
» end,
» end
```

COMANDO while

```
» k = round(rand(1)*10);
» i_max = 5;
» it_num = 1;
» while it_num < k
»   a(it_num,1)= 3*rand(1) + 5*sin(rand(1)*%pi);
»   it_num = it_num +1;
»   if it_num == i_max then
»     break
»   end
» end
» if it_num == k then
»   disp('El programa converge (k = it_num)')
» else
»   disp('Número de Iteraciones Excedidos (it_num = 5)')
» end
» disp('La matriz resultante es:  ')
» disp(' ')
» disp(a)
```

Preguntas