

Gráficas

Universidad de Concepción, Chile
Departamento de Geofísica
Programación Científica con Software libre

Primavera, 2011



Contenidos

- 1 Figuras
- 2 Líneas
- 3 Superficies
- 4 Contornos
- 5 Plot 3D
- 6 Avanzada

Figuras

Para guardar una copia del gráfico generado utilizaremos el comando `print`. Por lo general lo ejecutaremos de la forma:

- `print('FILENAME', OPTIONS)`
- `print(h, 'FILENAME', OPTIONS)`

La orientación puede ser modificada utilizando las opciones

- `'-portrait'`
- `'-landscape'`

figuras

La extensión con la cual queremos que nuestra imagen sea guardada se puede manejar con la opción '-dDEVICE', en donde DEVICE puede ser

- ps
- eps
- pdf
- png

Figuras

Comando útiles

- **clf** limpia la figura desplegada
- **hold on** fija la figura desplegada para realizar otra gráfica sobre ésta
- **hold off** libera la figura
- **subplot(n,m,p)** divide la ventana en una cuadrícula de n filas por m columnas, p entrega la posición del gráfico en la cuadrícula
- **box off** para gráficos 3D elimina las aristas del cubo dentro del cual se encuentra la gráfica

Personalización de gráficos

- **axis**

```
axis([min_x max_x min_y max_y])
```

- **labels**

```
xlabel('nombre eje x'), ylabel('nombre eje y'),
```

- **title**

```
title('titulo del grafico')
```

- **legend**

```
legent('var1', 'var2', ..)
```

- **grid**

Graficos de línea

Suponga que desea desplegar los valores de una función $y = f(x)$ para x en un intervalo dado, esto se puede conseguir haciendo:

- `plot(y)`
- `plot(x,y)`
- `plot(x,y,property,value,...)`

Graficos de línea

Ejemplo 1

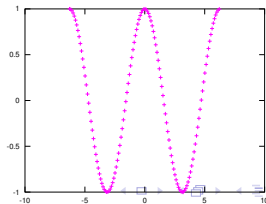
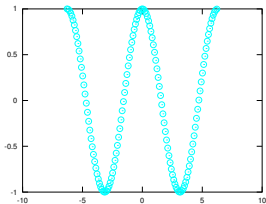
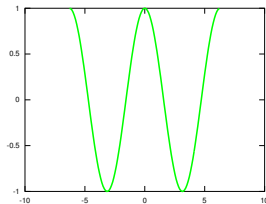
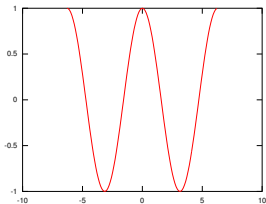
Ejemplo 1

Sea $y = \cos(x)$ con $x \in [-2\pi, 2\pi]$

- `plot(x,y,'r')`
- `plot(x,y,'g','linewidth',3)`
- `plot(x,y,'co','markersize',9)`
- `plot(x,y,'4+')`

Graficos de línea

Ejemplo 1



Gráficos de línea

Otras de las funciones gráficas de gran relevancia son:

- **hist** histogramas
- **semilogy** escala logarítmica en el eje y
- **errorbar** despliega datos con sus barras de error
- **stem** visualización de datos discretos

Gráficos de línea

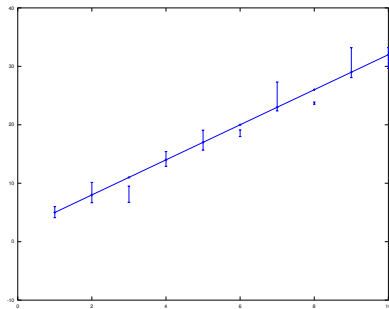
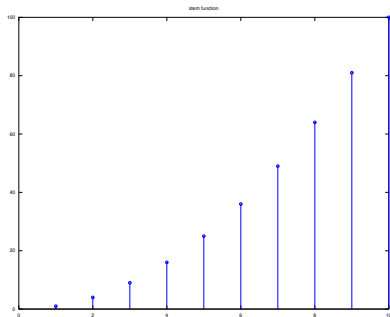
Ejemplo 2

Ejemplo 2

```
%ejm funcion stem  
x=1:1:10;  
y=x.^2;  
figure(1)  
stem(x,y,'b')  
title('stem_function')  
%ejm funcion  
x1=[1:1:10];  
y1=3*x1 +2;  
yp=2*randn(length(x1));  
ym=-2*randn(length(x1));  
figure(2)  
errorbar(x1,y1,ym,yp);  
title('errorbar_function')
```

Gráficos de línea

Ejemplo 2



Graficos de superficie

Si se tiene una función $f(x, y)$ donde f define una superficie, es posible graficarla utilizando los comandos:

- **surf** dados los vectores x , y grafica la superficie $z(x,y)$
`surf(x, y, z)`
- **surfc** dados los vectores x , y grafica la superficie $z(x,y)$ y los contornos de la proyección de ésta sobre el plano xy
`surfc(x, y, z)`
- **mesh** dados las matrices x , y grafica la superficie $z(x,y)$
`mesh(x, y, z)`

Gráficos de superficie

Ejemplo 3

Dada la función

$$f(x, y) = xe^{-(x^2+y^2)}$$

con $x, y \in [-2, 2]$. Utilice los comando descritos anteriormente para generar gráficos de la función f , incorpore nombre de los ejes coordenados y titulo en las figuras obtenidas.

Gráficos de superficie

Ejemplo 3

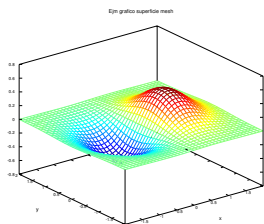
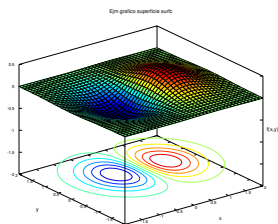
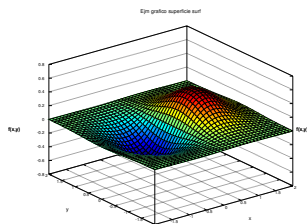
Ejemplo 3

```
# ejm 3
x=[-2:0.1:2];
y=[-2:0.1:2];

[X,Y]=meshgrid(x,y);
z=X.*exp(-X.^2-Y.^2);
figure(1)
surf(x,y,z)
title('Ejm_grafico_superficie_surf','fontsize',15)
xlabel('x','fontsize',15); ylabel('y','fontsize',15);
zlabel('f(x,y)','fontsize',15)
figure(2)
surfc(x,y,z)
title('Ejm_grafico_superficie_surfc','fontsize',15)
xlabel('x','fontsize',15); ylabel('y','fontsize',15);
zlabel('f(x,y)','fontsize',15)
figure(3)
mesh(X,Y,z)
title('Ejm_grafico_superficie_mesh','fontsize',15)
xlabel('x','fontsize',15); ylabel('y','fontsize',15);
zlabel('f(x,y)','fontsize',15)
```

Gráficos de superficie

Ejemplo 3



Gráficos de contorno

Nos permiten conocer la proyección de una superficie sobre el plano indicándonos curvas de nivel. Estos gráficos pueden ser obtenidos utilizando el comando:

- **contour** dados las matrices x , y grafica la proyección de la superficie $z(x,y)$ sobre el plano

```
contour(x,y,z)
```

En este tipo de figuras es útil agregar un indicador de los valores de las diferentes curvas, lo anterior es conseguido utilizando:

- **colorbar** despliega una barra de color al lado derecho de la figura

```
colorbar('EastOutside')
```

Gráficos de contorno

Ejemplo 4

Dada la función

$$g(x, y) = \sin(x^2 + y^2) \cos(x) \sin(y)$$

para $x, y \in [-\pi, \pi]$. Obtenga un gráfico de superficie y un gráfico de contornos para la superficie definida por g .

Gráficos de contorno

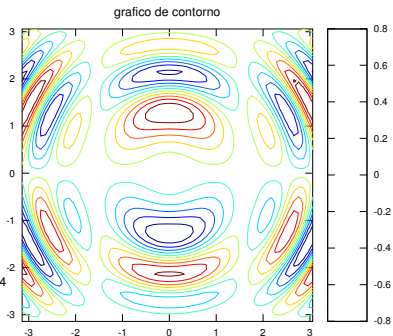
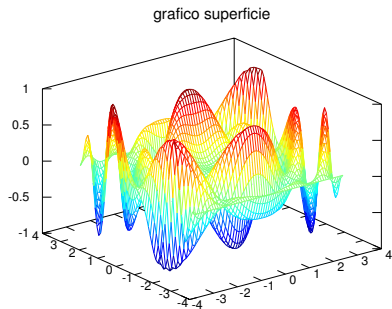
Ejemplo 4

Ejemplo 4

```
x = [-pi:0.1:pi];  
y = [-pi:0.1:pi];  
[X,Y] = meshgrid(x,y);  
z = sin(X.^2+Y.^2).*cos(X).*sin(Y);  
figure(1)  
mesh(X,Y,z)  
title('grafico_superficie','fontsize',15)  
print -dpdf cp1.pdf  
figure(2)  
contour(X,Y,z)  
axis('tight')  
colorbar('EastOutside')  
title('grafico_de_contorno','fontsize',15)  
print -dpdf cp2.pdf
```

Gráficos de contorno

Ejemplo 4



Gráficos de contorno

Cuando necesitamos graficar campos vectoriales (e.g. campos de velocidad) podemos hacer uso del comando

- **quiver** grafica las componentes vectoriales U , V en una grilla definida por x , y

```
quiver(x,y,U,V)
```

Gráficos de contorno

Ejemplo 5

Dado el potencial

$$\phi(x, y) = \frac{Q}{2\pi} \log(|r - r_0|)$$

utilizando *quiver* grafique el campo de velocidad definido por $\nabla\phi$.

Gráficos de contorno

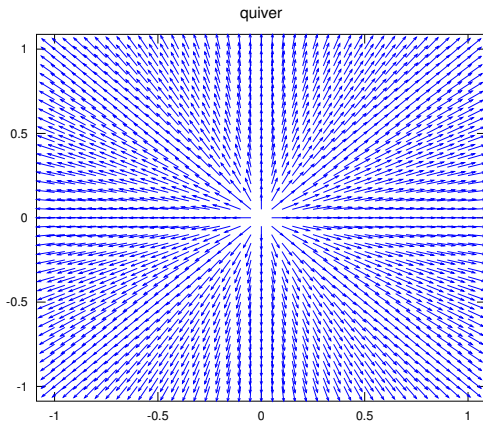
Ejemplo 5

Ejemplo 5

```
x = [-1:0.05:1];  
y = [-1:0.05:1];  
x0 = 0;  
y0 = 0;  
Q = 1;  
[X,Y] = meshgrid(x,y);  
r = sqrt((X-x0).^2+(Y-y0).^2);  
phi = Q/2/pi*log(r);  
[dx,dy] = gradient(r,0.1,0.1);  
quiver(X,Y,dx,dy)  
axis('tight')  
title('quiver','fontsize',15)  
print -dpdf qexm.pdf
```

Gráficos de contorno

Ejemplo 5



Gráficos 3D

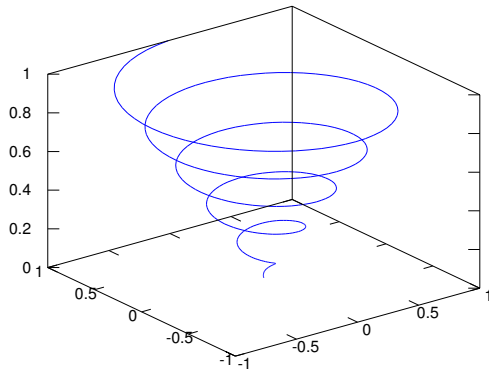
plot3 despliega datos en 3D sin necesidad que éstos formen una superficie.

Ejemplo 6

```
# ejm helice  
t = 0:0.1:10*pi;  
r = linspace(0,1,numel(t));  
z = linspace(0,1,numel(t));  
plot3(r.*sin(t),r.*cos(t),z)  
box off  
print -dpdf heli.pdf
```

Gráficos 3D

Ejemplo 6



Configuración Avanzada

Las figuras en octave son construcciones de los objetos gráficos

root figure	figura padre de todos los objetos
figure	ventana de la figura
axes	ejes coordenados
line	línea en 2D o 3D
text	anotaciones, leyendas
image	imagen en mapa de caracteres
patch	curva polinomial cerrada limitada a 2D
surface	superficie 3D

cada uno de estos objetos tiene propiedades que pueden ser modificadas para mejorar la apariencia del gráfico

Configuración Avanzada

Si queremos cambiar algún atributo del objeto gráfico utilizamos **set**

- `set(H,P,V,...)`
El objeto H cambia la propiedad P al valor V
- `set(gca,'xlim',[-1,1])`
Redefine los límites del eje x

Configuración Avanzada

`gcf` Retorna la figura referenciada, si ésta no existe la crea. El comando será útil cuando necesitemos modificar alguna propiedad de la figura.

- `set(gcf,'visible','off')`

Modificamos el objeto figura para que ésta no sea desplegada

- `set(gcf,'paperorientation','landscape')`
`papersize = [30, 21]/2.54;`
`set (gcf,'papersize',papersize)`
`set (gcf,'paperposition',[0.25 0.25, papersize-0.5])`

La figura será impresa dejando un margen de $0.25in$ en la hoja, útil cuando se quiera guardar la imagen como pdf o eps

Configuración Avanzada

`gca` Retorna los ejes coordenados referenciados. Este comando será útil cuando necesitemos modificar alguna propiedad de los ejes

- `set(gca, 'fontsize', 14)`
Modifica el tamaño de la fuente de los marcadores
- `set(gca, 'yticklabel', ylab)`
Modifica los marcadores del eje y, ylab puede ser una columna de strings