

Programación Numérica para Geofísica

PNG

Andrés Sepúlveda

Departamento de Geofísica
Universidad de Concepción

25/05/2020

Anuncios

- Dudas, consultas, quejas, alabanzas, ...
- Hoy: Gráficos Básicos

- En Octave/Matlab los gráficos son presentados en un objeto que se llama *figure*, el cual puede tener varios elementos.
- Por defecto se grafica en *figure(1)*.
- Si queremos tener dos ventanas de gráficos en paralelo, podemos hacer:

```
figure(1)
    plot(x,y,'-*')
figure(2)
    plot(x,z,'-o')
```

- También podemos poner varios gráficos en una misma ventana, usando la función *subplot*

```
subplot(2,1,1);
    plot(x,y,'-*')
subplot(2,1,2);
    plot(x,z,'-o')
```

- En *subplot(A,B,n)* se crea una ventana con **A** filas, **B** columnas, y se grafica la **n**-ésima subventana de un total de **A*B**.

Números Aleatorios

Para practicar, graficaremos matrices o vectores de números aleatorios. Hay varias formas de generar números aleatorios, los más usados son:

- Matriz Aleatoria / Vector Aleatorio (Normal o Gaussiano)

```
r = randn(5);  
rv = randn(5,1);
```

- Matriz Aleatoria / Vector Aleatorio (Uniforme)

```
ru = rand(10,1)
```

- Comparamos ambos

```
figure(3)
subplot(2,2,1); r1 = rand(10,1);
hist(r1); title('U: 10 elementos')

subplot(2,2,2); r2 = rand(10000,1);
hist(r2); title('U: 10000 elementos')

subplot(2,2,3); r3 = randn(10,1);
hist(r3); title('G: 10 elementos')

subplot(2,2,4); r4 = randn(10000,1);
hist(r4); title('G: 10000 elementos')
```

- Ojo con hist()

```
figure(3)
subplot(2,1,1); r1 = randn(10000,1);
hist(r1);      title(' 10 divisiones')

subplot(2,1,2); r2 = randn(10000,1);
hist(r2,100); title(' 100 divisiones')
```

Polyfit - Recta y más

Es MUY común querer ajustar una recta entre dos variables.

- La solución ya programada:

```
p=polyfit(x,y,n);
```

- Los valores ajustados

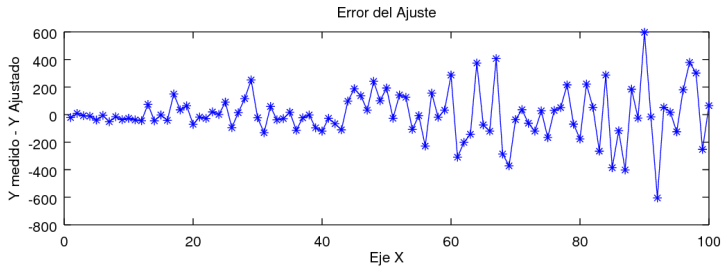
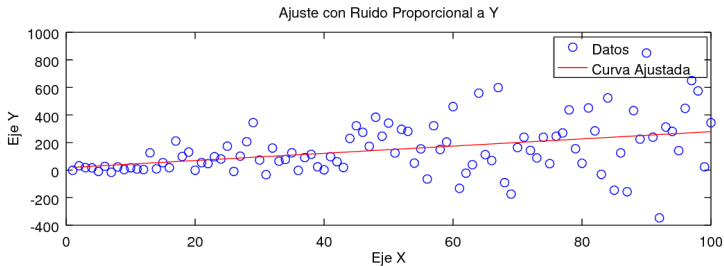
```
y2 = polyval(p,x)
```

- El error del ajuste

```
error = y2 - y
```

¿Qué falta?

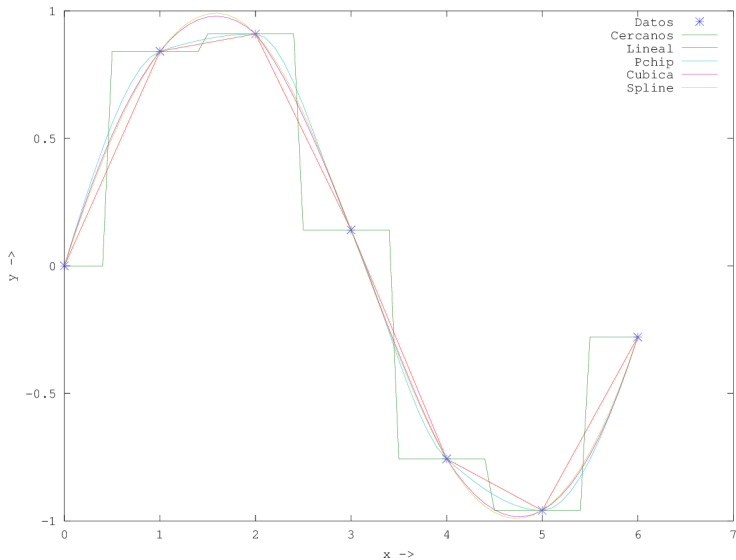
Polyfit



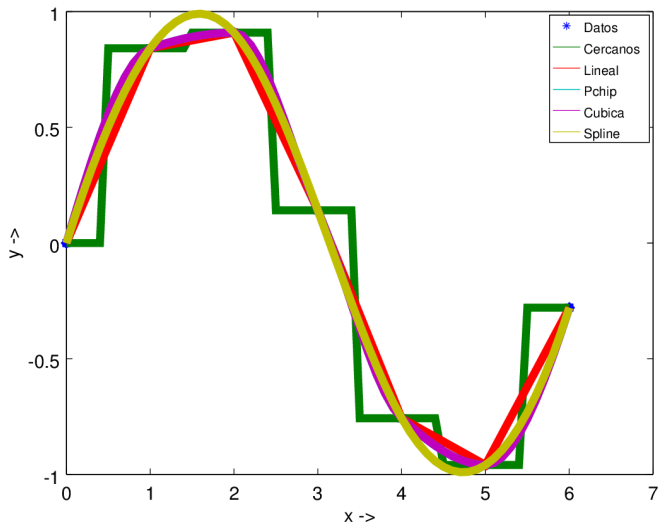
Interpolación

Hay múltiples formas de interpolar datos:

```
x = 0:2*pi;
y = sin(x);
xi = 0:0.1:2*pi;
y_n = interp1(x, y, xi, "nearest");
y_l = interp1(x, y, xi, "linear");
y_p = interp1(x, y, xi, "pchip");
y_c = interp1(x, y, xi, "cubic");
y_s = interp1(x, y, xi, "spline");
plot(x, y, "*", xi, y_n, "-", xi, y_l, "-", ...
     xi, y_p, "-", xi, y_c, "-", xi, y_s, "-");
xlabel("x ->");
ylabel("y ->");
legend("Dat", "Cercanos", "Lineal", "Pchip", "Cubica", "Spline");
print("-dpng", "Distintas_interpolaciones.png");
```



¿Se ve bien?



Para PPT

Otras soluciones -Versión PPT

```
plot(x, y, "*", 'LineWidth',6, xi, y_n, "-", ...  
     'LineWidth',6, xi, y_l, "-", 'LineWidth',6, ...  
     xi, y_p, "-", 'LineWidth',6, xi, y_c, "-", ...  
     'LineWidth',6,xi, y_s, "-", 'LineWidth',6);  
xlabel("x ->", 'FontSize',14);  
ylabel("y ->", 'FontSize',14);  
legend("Datos", "Cercanos", "Lineal", "Pchip", "Cubica", "Spline");  
set (gca, "FontSize", 14)
```

Barra de errores

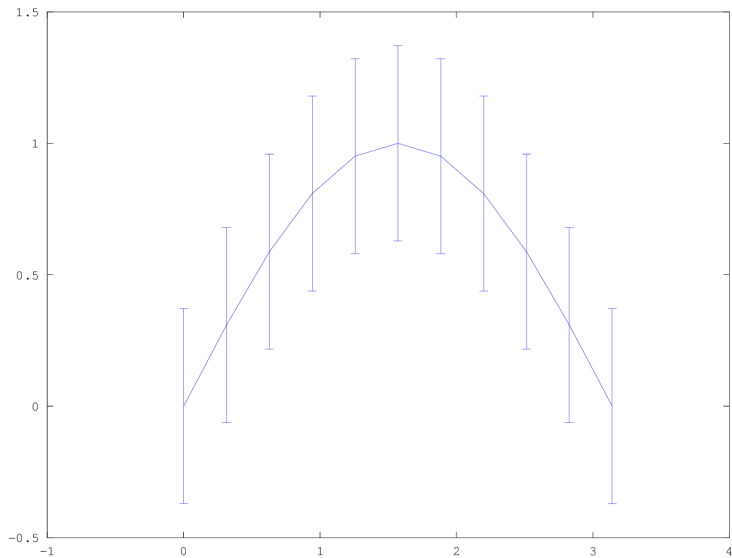
errorbar

En lo posible, hay que representar la incerteza asociada a las mediciones y/o resultados.

```
X = 0:pi/10:pi;  
Y = sin(X);  
E = std(Y)*ones(size(X));  
errorbar(X,Y,E)
```

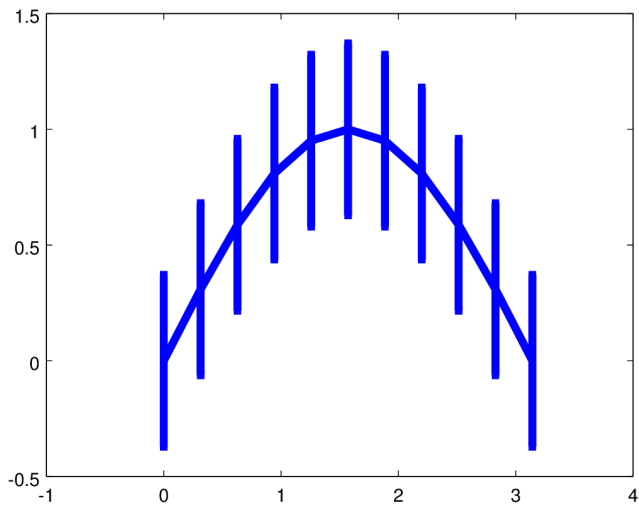
Gráficos

errorbar



Gráficos

errorbar - PPT



Manejo de ejes y símbolos

Es posible incluir letras griegas o símbolos

```
x = -pi:.1:pi;  
y = sin(x);  
p = plot(x,y);  
set(gca,'XTick',-pi:pi/2:pi);
```

% Matlab

```
% set(gca,'XTickLabel', '-pi','-pi/2','0','pi/2','pi');
```

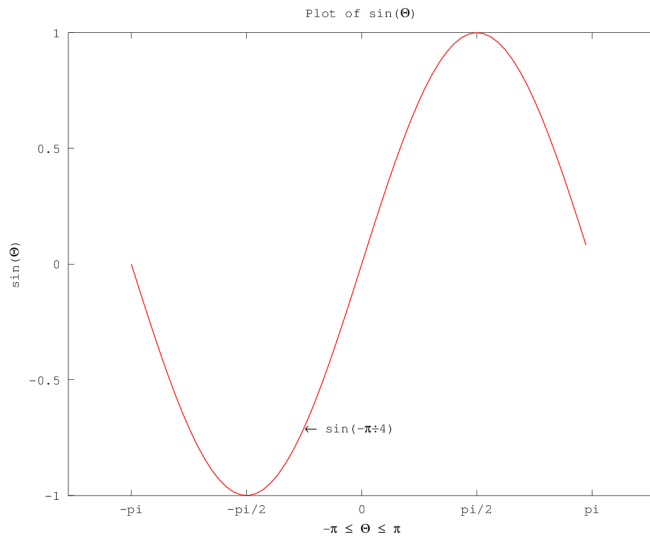
% Octave

```
set(gca,'XTickLabel',{'-pi','-pi/2','0','pi/2','pi'});
```

```
xlabel('-\pi \leq \Theta \leq \pi');  
ylabel('sin(\Theta)'); % Ojo con simbolos griegos  
title('Plot of sin(\Theta)');  
text(-pi/4,sin(-pi/4),'\leftarrow sin(-\pi\div4)',...  
'HorizontalAlignment','left');  
set(p,'Color','red','LineWidth',2); % cambio color  
% y ancho de línea
```

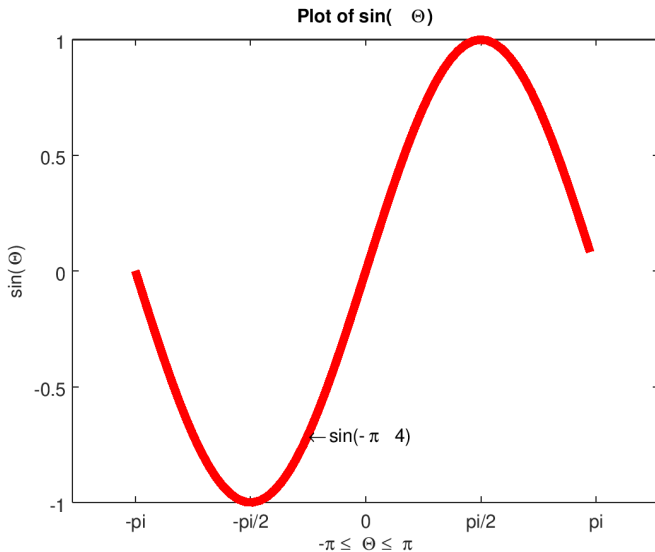

Gráficos

Ejes y Símbolos



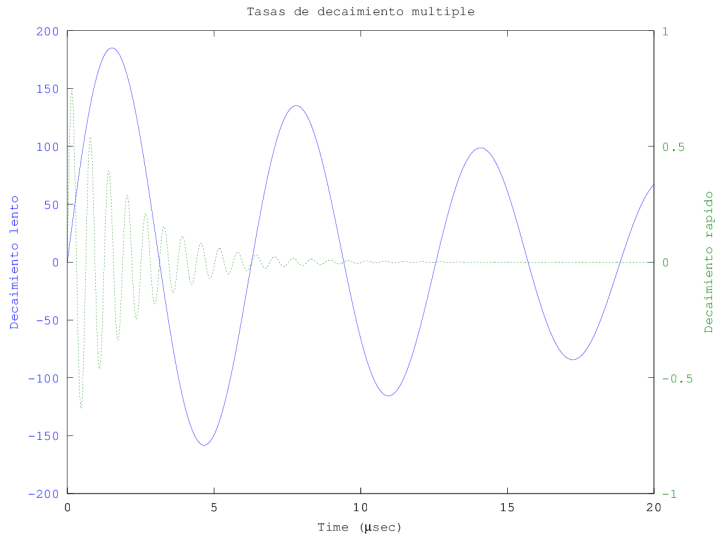
Gráficos

Ejes y Símbolos



Dos ejes Y

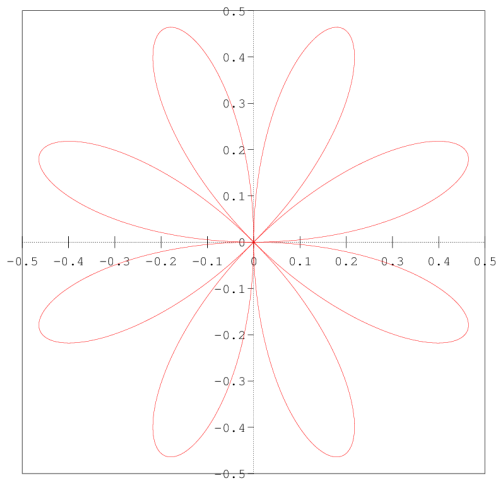
```
x = 0:0.01:20;  
y1 = 200*exp(-0.05*x).*sin(x);  
y2 = 0.8*exp(-0.5*x).*sin(10*x);  
[AX,H1,H2] = plotyy(x,y1,x,y2,'plot');  
set(get(AX(1),'Ylabel'),'String','Decaimiento lento');  
set(get(AX(2),'Ylabel'),'String','Decaimiento rapido');  
xlabel('Time (\musec)') ;  
title('Tasas de decaimiento multiple');  
set(H1,'LineStyle','-');  
set(H2,'LineStyle',':');
```



Polar

```
t = 0:.01:2*pi;  
polar(t,sin(2*t).*cos(2*t),'-r')
```

Gráfico Polar

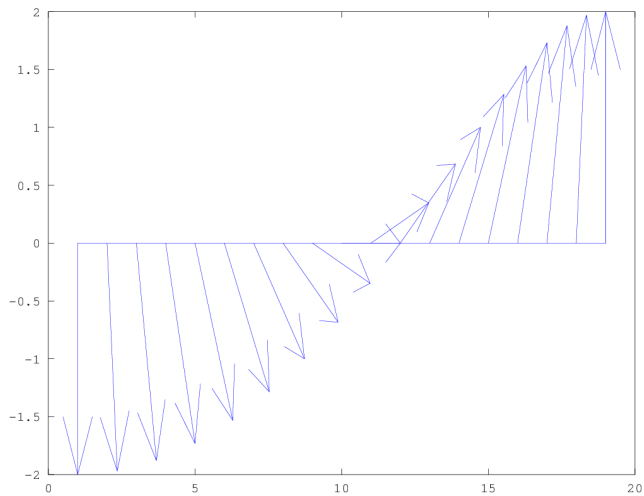


Muy útil para representar series de tiempo de vectores.

```
theta = (-90:10:90)*pi/180;  
r = 2*ones(size(theta));  
[u,v] = pol2cart(theta,r);  
feather(u,v);
```

Gráficos

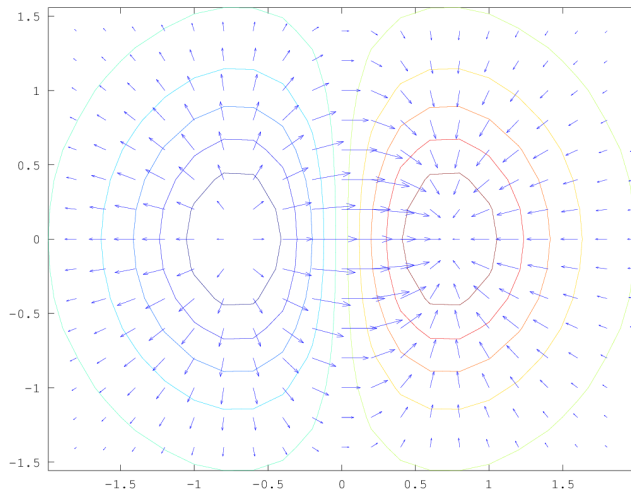
feather



Vectores 2D

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

Gráfico quiver



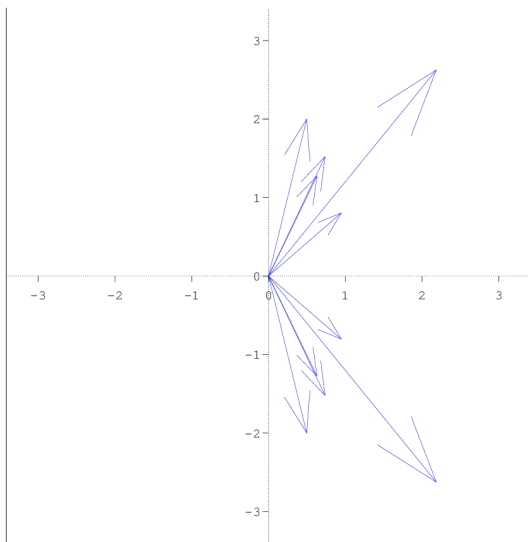
Función box

Agrega o elimina el marco

```
[X,Y] = meshgrid(-2:.2:2);  
Z = X.*exp(-X.^ 2 - Y.^ 2);  
[DX,DY] = gradient(Z,.2,.2);  
contour(X,Y,Z)  
hold on  
quiver(X,Y,DX,DY)  
colormap;  
box off
```

Gráficos

box



Gráficos

Más comunes

box	errorbar	loglog
plot	plot3	plotyy
polar	semilogx	semilogy
compass	feather	quiver
quiver3	hist	

Recuerde que es obligatorio poner título al gráfico, describir cada eje (con sus unidades, si corresponde), y las variables graficadas (usando **legend**).

Use símbolos para identificar cada punto que es graficado, e.g.
`plot(x,y,'-*')`