

Gráficas en MATLAB

Pontificia Universidad Javeriana
Curso de Análisis Numérico

Victor Manuel Peñaranda Vélez

vpenarandav@unal.edu.co

hidroingvmpv@gmail.com

Agosto 08 y 15 de 2008

Introducción

MATLAB es provisto de un conjunto de subrutinas para el desarrollo de gráficas. Estas subrutinas emplean técnicas convencionales para la presentación gráfica de datos, tales como: gráfica de funciones en coordenadas rectangulares y polares, histogramas, contornos, superficies, y animaciones.

Introducción

Como lo hemos visto con anterioridad, MATLAB, está enfocado principalmente al trabajo con matrices, por lo tanto no es de extrañar que los argumentos principales de las funciones de graficación, sean de este tipo.

- ❖ Las matrices argumentos, contienen las posiciones, de puntos por donde pasara la línea o la superficie.
- ❖ Estas matrices pueden ser generadas, por medio del operador “ : ” , o con el comando **“linspace”**

Pasos para construir una gráfica

En el proceso de construcción de una gráfica se sugiere el siguiente procedimiento:

1. Preparación de los datos.

```
>> x = 0:0.1:10;  
>> y = sin(x) + cos(x.^2);
```

2. Seleccionar la ventana de gráficas y la posición de la gráfica dentro de la misma.

```
>> h1 = figure(10);
```


Pasos para construir una gráfica

3. Llamado de la función de graficación.

```
>> h2 = plot(x,y,x,y);
```

4. Establecer las características de la línea y marcadores.

```
>> set(h2,'LineWidth',2,{'LineStyle'},{'--';'.'})
```

```
>> set(h2,{'Color'},{'r';'b'})
```

```
>> set(h2(2),'MarkerSize',16)
```

Pasos para construir una gráfica

5. Establecer las propiedades de los ejes.

```
>> axis([0 10 -2.5 2.5])  
>> grid on
```

6. Construir las etiquetas y anotaciones necesarias en la gráfica.

```
>> xlabel('Tiempo')  
>> ylabel('Amplitud')  
>> legend(h2, 'Línea continua','Puntos')  
>> title('Función Trigonométrica')  
>> [yi,ix] = min(y);  
>> text(x(ix),yi,'First Min \rightarrow','HorizontalAlignment','right')
```


Pasos para construir una gráfica

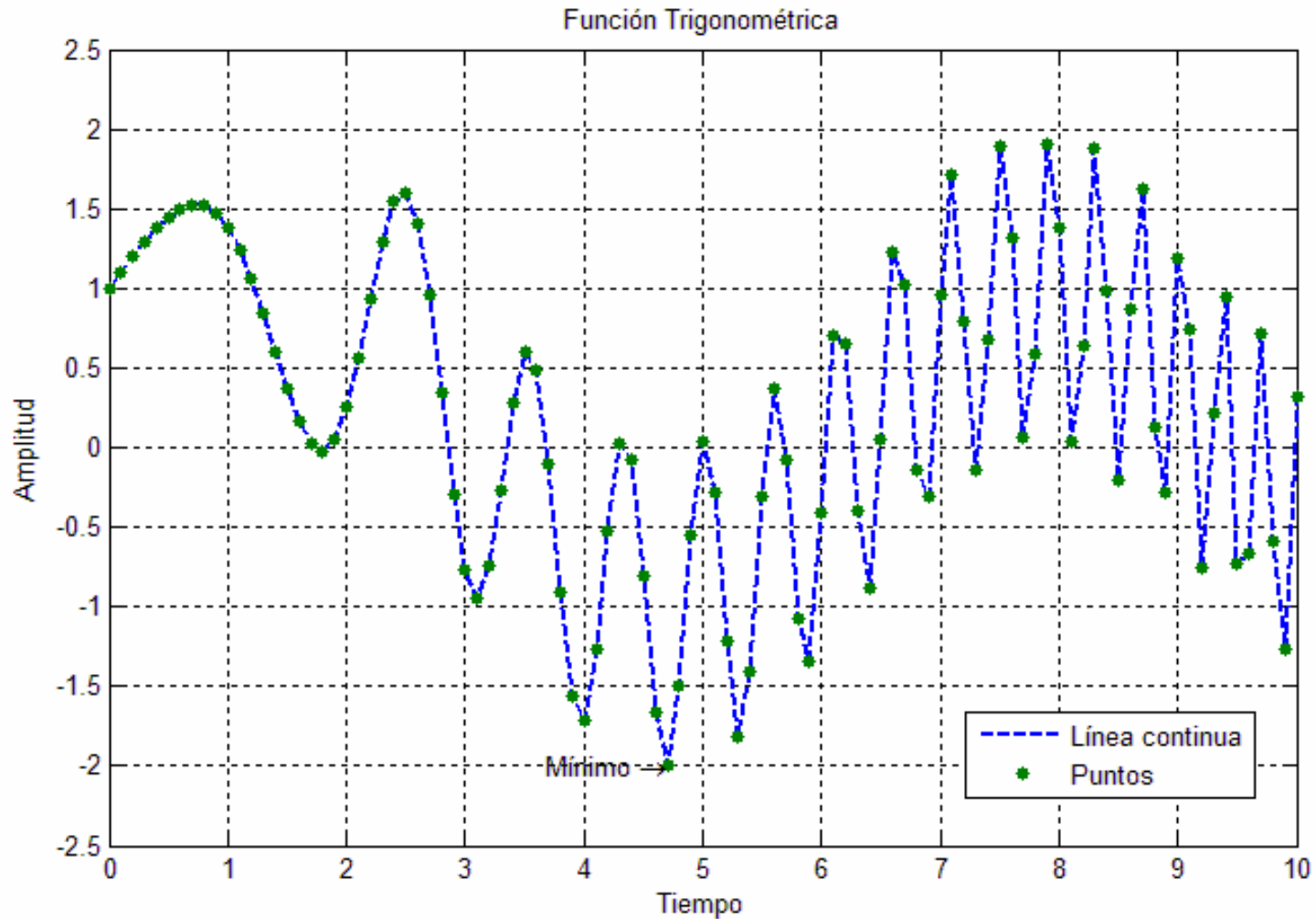
7. Establecer las características de la ventana.

```
>> set(h1,'Color',[0.3,0.2,0.1], 'Position',[50 50 1000 700], ...  
        'Name','Función Trigonométrica')
```

8. Guardar la gráfica.

```
>> saveas(h1, 'Ejemplo-grafica.bmp')
```

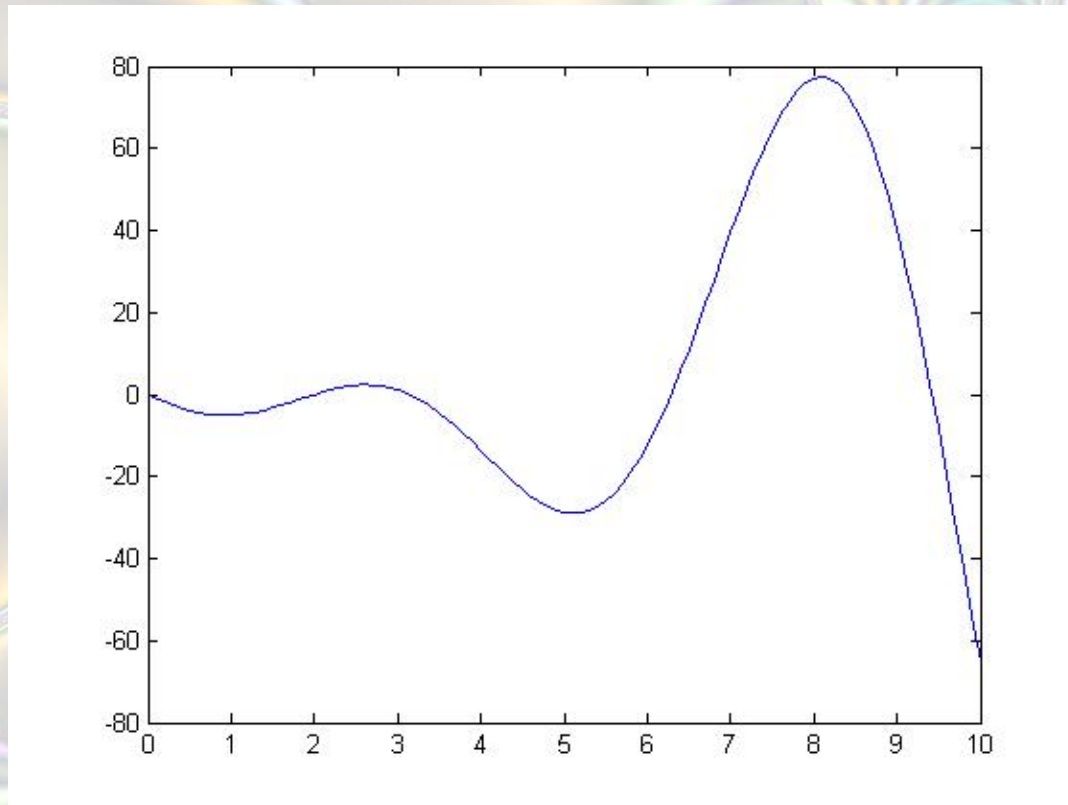
Pasos para construir una gráfica



Gráficas en 2D

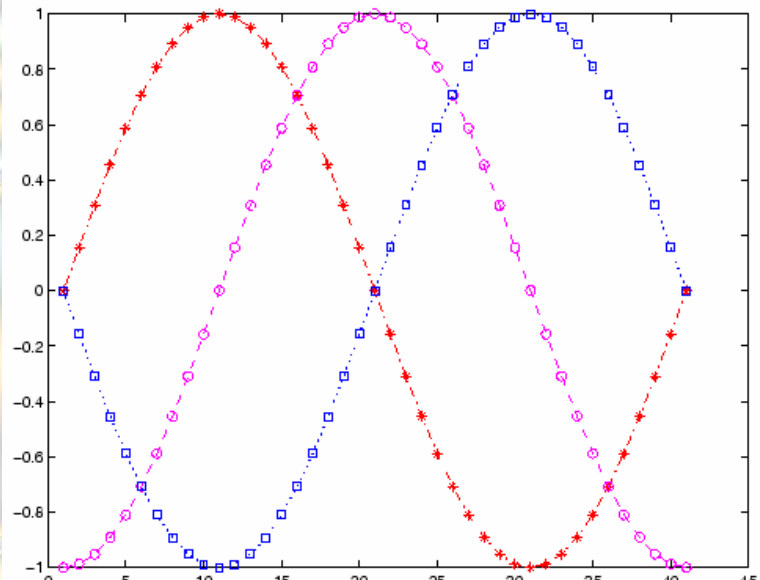
Comando “plot”:

Creación del vector objetivo $y = f(x)$
 $y = (x^2 + 3x - 10) \sin x$



Estilo de líneas y marcadores:

Specifier	Line Style
-	Solid line (default)
--	Dashed line
:	Dotted line
-.	Dash-dot line



Specifier	Marker Type
+	Plus sign
o	Circle
*	Asterisk
.	Point
x	Cross
'square' or s	Square
'diamond' or d	Diamond
^	Upward-pointing triangle
v	Downward-pointing triangle
>	Right-pointing triangle
<	Left-pointing triangle
'pentagram' or p	Five-pointed star (pentagram)
'hexagram' or h	Six-pointed star (hexagram)

Especificando colores:

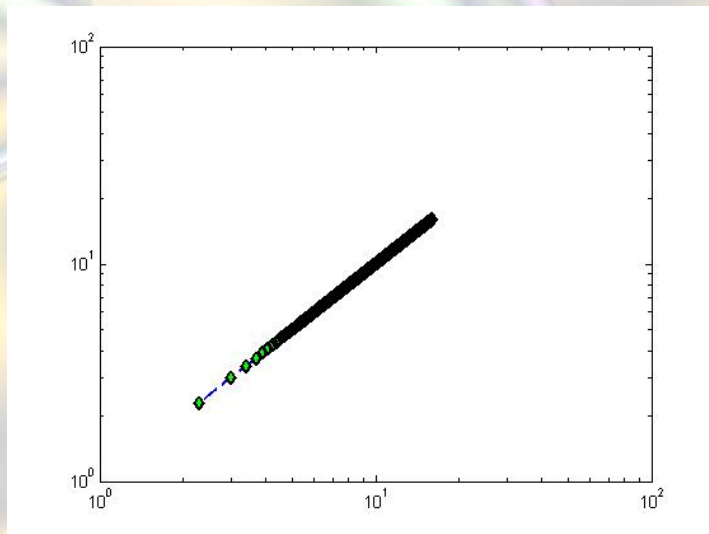
Specifier	Color
r	Red
g	Green
b	Blue
c	Cyan
m	Magenta
y	Yellow
k	Black
w	White

```
>> x = -pi:pi/10:pi;  
>> y = exp(tan(x));  
>> plot(x,y,'--bs','LineWidth',2,...  
        'MarkerEdgeColor','k',...  
        'MarkerFaceColor','g',...  
        'MarkerSize',8)
```

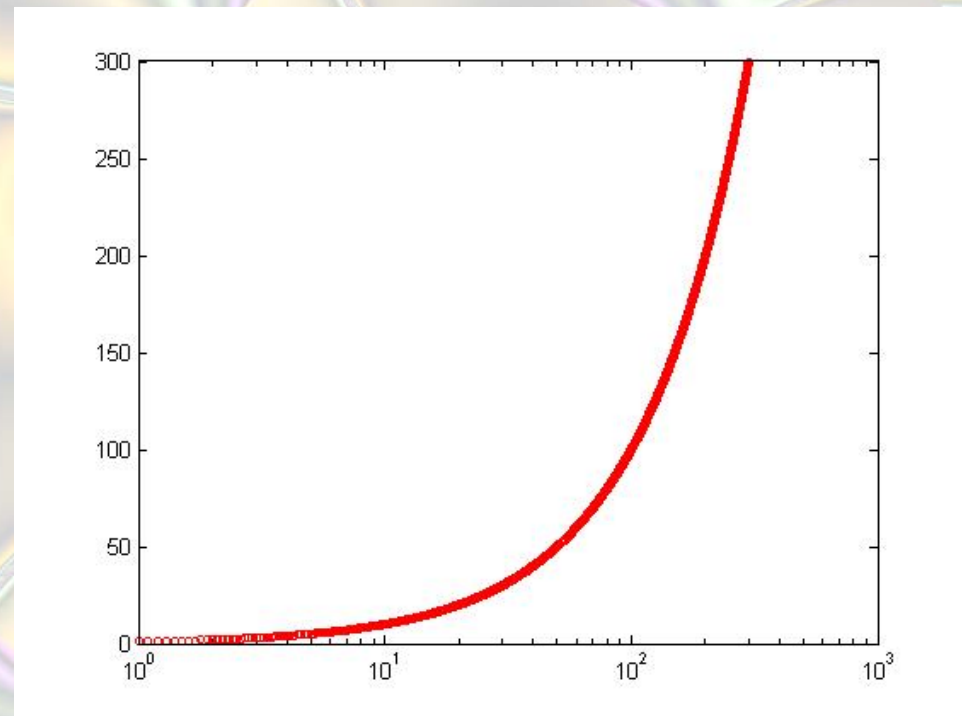
Gráficas en 2D

Comando “loglog”, “semilogx”, “semilogy”:

```
>> x = log(0.0001:10:10000000);  
>> y = log(0.0001:10:10000000);  
>> loglog(x,y,'--bd','LineWidth',2,...  
    'MarkerEdgeColor','k',...  
    'MarkerFaceColor','g',...  
    'MarkerSize',7)
```



```
>> x = 1:0.1:300;  
>> semilogx(x,x,'or','MarkerSize',3)  
>> grid on
```



Gráficas en 2D

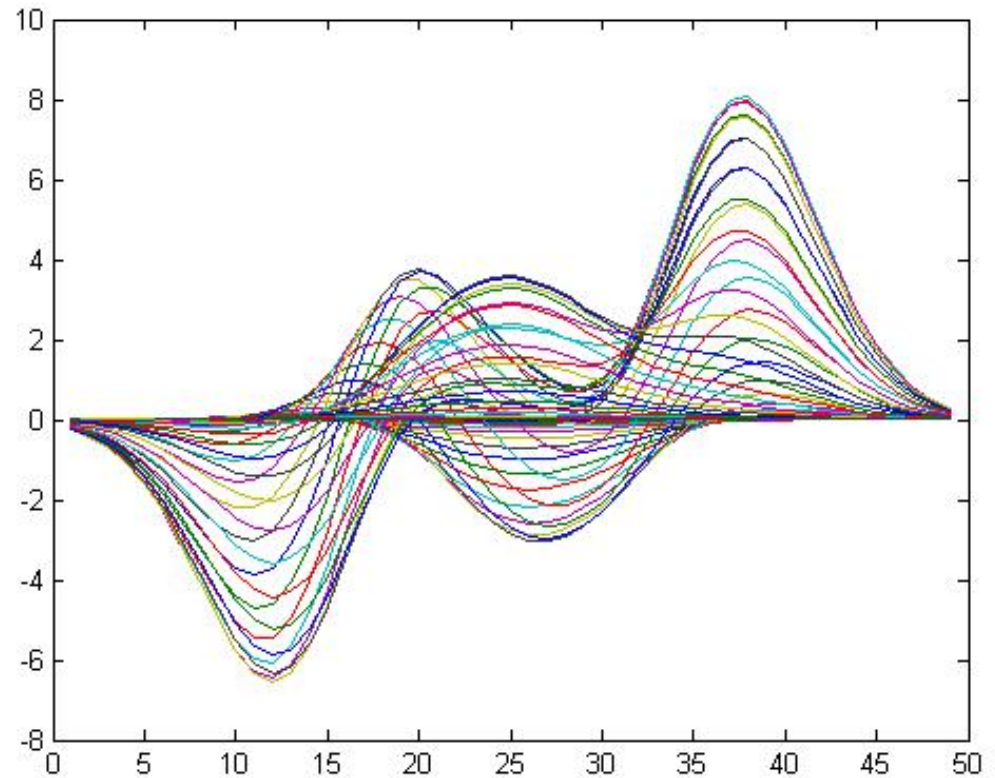
Graficación de múltiples datos:

```
>> y = peaks;
```

```
>> plot(y)
```

```
>> y = 1:length(peaks);
```

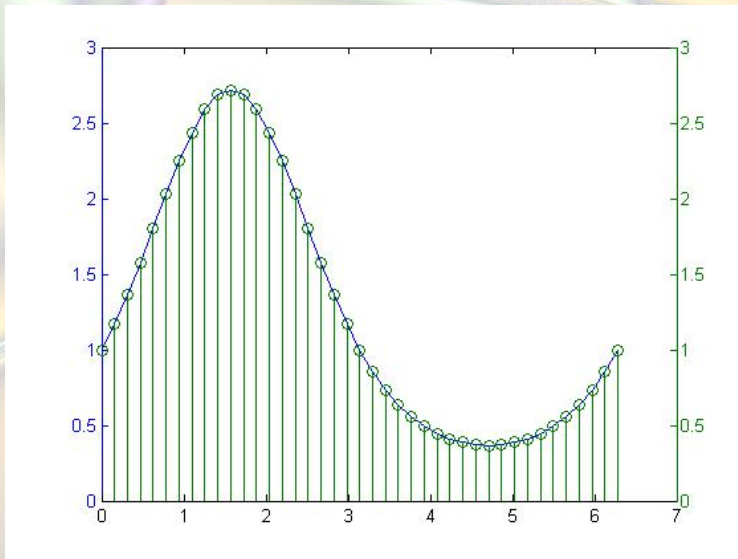
```
>> plot(peaks,y)
```



Gráficas en 2D

Graficación con dos ejes y:

```
>> t = 0:pi/20:2*pi;  
>> y = exp(sin(t));  
>> plotyy(t,y,t,y,'plot','stem')
```



```
>> t = 0:pi/20:2*pi;  
>> t = 0:900; A = 1000;  
>> a = 0.005; b = 0.005;  
>> z1 = A*exp(-a*t); z2 = sin(b*t);  
>> [ha,h1,h2] = plotyy(t,z1,t,z2,'semilogy','plot')  
>> axes(ha(1))  
>> ylabel('Semilogarítmico')  
>> axes(ha(2))  
>> ylabel('Lineal')  
>> set(h2,'LineStyle','--')
```


Características de los ejes

```
>> x = -pi:.1:pi; y = sin(x);  
>> plot(x,y)  
>> set(gca,'XTick',-pi:pi/2:pi)  
>> set(gca,'XTickLabel',{'-pi','-pi/2','0','pi/2','pi'})  
>> xlabel('-\pi \leq \Theta \leq \pi')  
>> ylabel('sin(\Theta)')  
>> title('Plot of sin(\Theta)')  
>> text(-pi/4,sin(-pi/4),'\leftarrow sin(-\pi\div4)',...  
        'HorizontalAlignment','left')  
>> set(findobj(gca,'Type','line','Color',[0 0 1]),...  
        'Color',[0.5,0,0.5],'LineWidth',2)
```

Interprete TEX

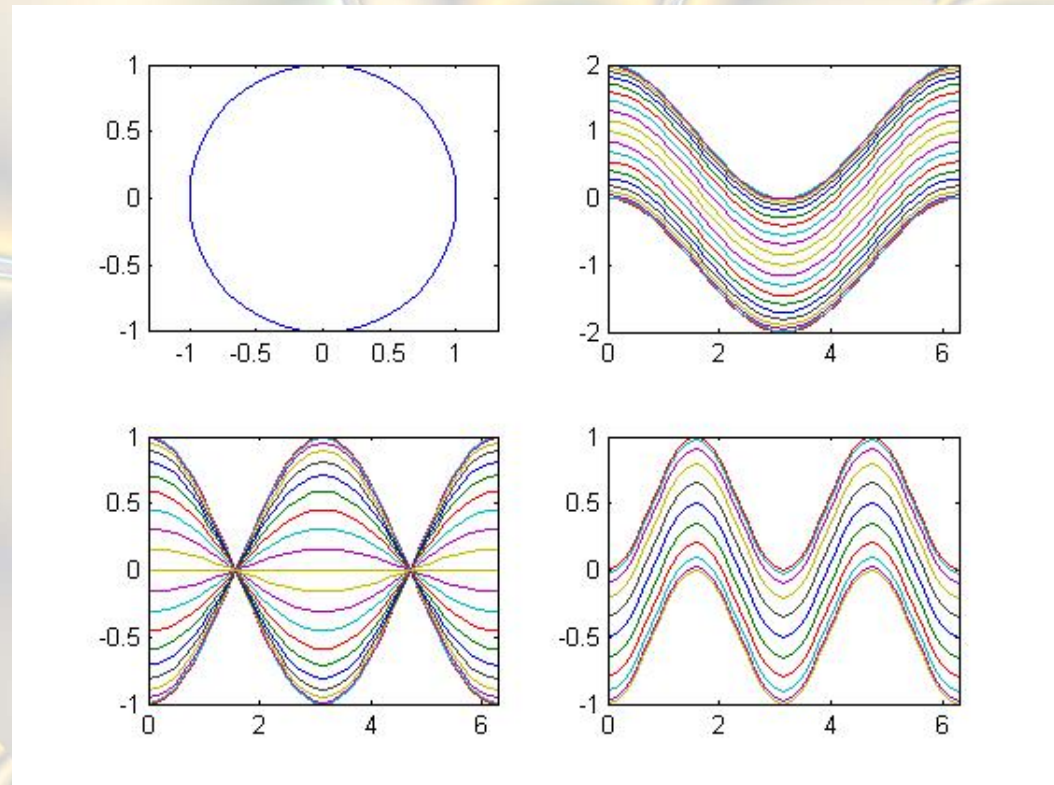
Character Sequence	Symbol	Character Sequence	Symbol	Character Sequence	Symbol
<code>\alpha</code>	α	<code>\upsilon</code>	υ	<code>\sim</code>	\sim
<code>\beta</code>	β	<code>\phi</code>	Φ	<code>\leq</code>	\leq
<code>\gamma</code>	γ	<code>\chi</code>	χ	<code>\infty</code>	∞
<code>\delta</code>	δ	<code>\psi</code>	ψ	<code>\clubsuit</code>	\clubsuit
<code>\epsilon</code>	ϵ	<code>\omega</code>	ω	<code>\diamondsuit</code>	\diamondsuit
<code>\zeta</code>	ζ	<code>\Gamma</code>	Γ	<code>\heartsuit</code>	\heartsuit
<code>\eta</code>	η	<code>\Delta</code>	Δ	<code>\spadesuit</code>	\spadesuit
<code>\theta</code>	Θ	<code>\Theta</code>	Θ	<code>\leftrightarrow</code>	\leftrightarrow
<code>\vartheta</code>	ϑ	<code>\Lambda</code>	Λ	<code>\leftarrow</code>	\leftarrow
<code>\iota</code>	ι	<code>\Xi</code>	Ξ	<code>\uparrow</code>	\uparrow
<code>\kappa</code>	κ	<code>\Pi</code>	Π	<code>\rightarrow</code>	\rightarrow
<code>\lambda</code>	λ	<code>\Sigma</code>	Σ	<code>\downarrow</code>	\downarrow
<code>\mu</code>	μ	<code>\Upsilon</code>	Υ	<code>\circ</code>	\circ
<code>\nu</code>	ν	<code>\Phi</code>	Φ	<code>\pm</code>	\pm

Interprete TEX

Character Sequence	Symbol	Character Sequence	Symbol	Character Sequence	Symbol
<code>\xi</code>	ξ	<code>\Psi</code>	Ψ	<code>\geq</code>	\geq
<code>\pi</code>	π	<code>\Omega</code>	Ω	<code>\propto</code>	\propto
<code>\rho</code>	ρ	<code>\forall</code>	\forall	<code>\partial</code>	∂
<code>\sigma</code>	σ	<code>\exists</code>	\exists	<code>\bullet</code>	\bullet
<code>\varsigma</code>	ς	<code>\ni</code>	\ni	<code>\div</code>	\div
<code>\tau</code>	τ	<code>\cong</code>	\cong	<code>\neq</code>	\neq
<code>\equiv</code>	\equiv	<code>\approx</code>	\approx	<code>\aleph</code>	\aleph
<code>\Im</code>	\Im	<code>\Re</code>	\Re	<code>\wp</code>	\wp
<code>\otimes</code>	\otimes	<code>\oplus</code>	\oplus	<code>\oslash</code>	\oslash
<code>\cap</code>	\cap	<code>\cup</code>	\cup	<code>\supseteq</code>	\supseteq
<code>\supset</code>	\supset	<code>\subseteq</code>	\subseteq	<code>\subset</code>	\subset
<code>\int</code>	\int	<code>\in</code>	\in	<code>\circ</code>	\circ
<code>\rfloor</code>	\rfloor	<code>\lceil</code>	\lceil	<code>\nabla</code>	∇

Graficación múltiple

```
>> t = 0:pi/20:2*pi;  
>> [x,y] = meshgrid(t);  
>> subplot(2,2,1)  
>> plot(sin(t),cos(t)), axis equal  
>> subplot(2,2,2)  
>> z = sin(x)+cos(y);  
>> plot(t,z), axis([0 2*pi -2 2])  
>> subplot(2,2,3)  
>> z = sin(x).*cos(y);  
>> plot(t,z), axis([0 2*pi -1 1])  
>> subplot(2,2,4)  
>> z = (sin(x).^2)-(cos(y).^2);  
>> plot(t,z), axis([0 2*pi -1 1])
```



Gráfica de barras agrupadas

```
>> Y = [5 2 1  
      8 7 3  
      9 8 6  
      5 5 5  
      4 3 2];
```

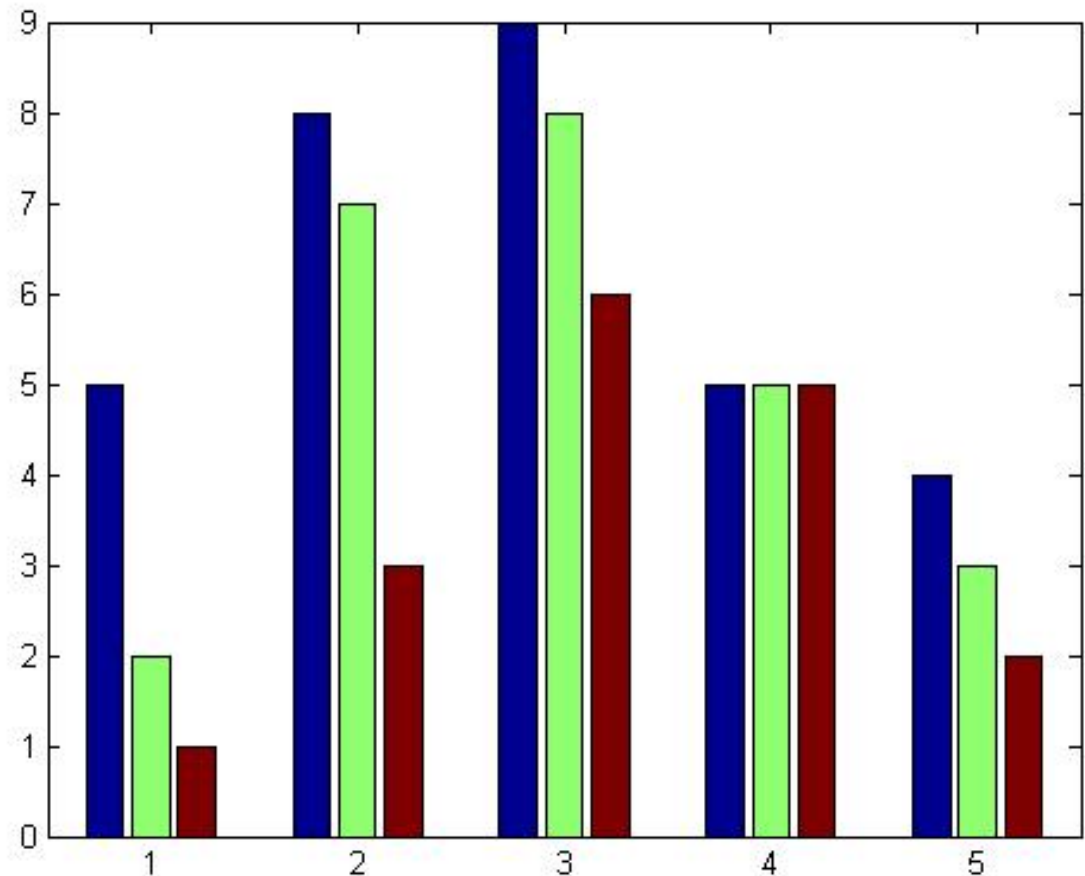
```
>> bar(Y)
```

¿Qué sucede con
“bar3”?

Revisar:

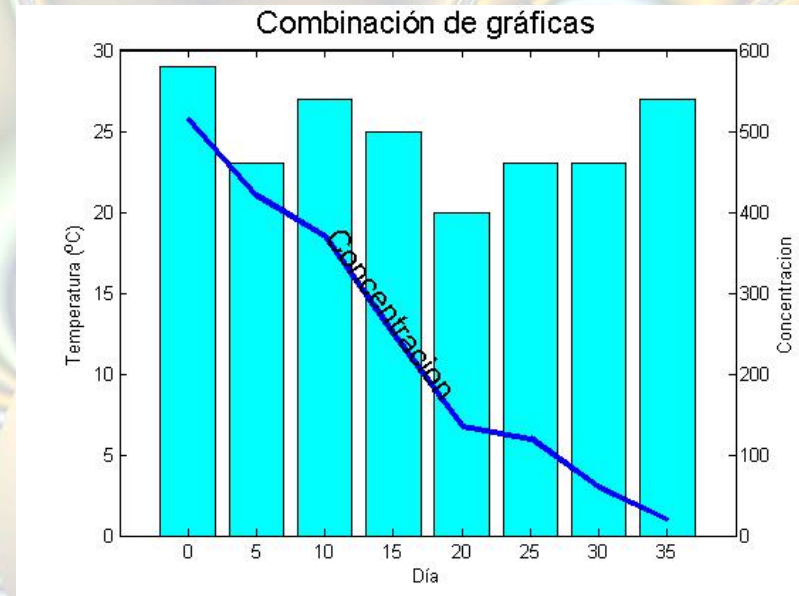
```
>> bar(Y,'stack')
```

```
>> barh(Y,'stack')
```



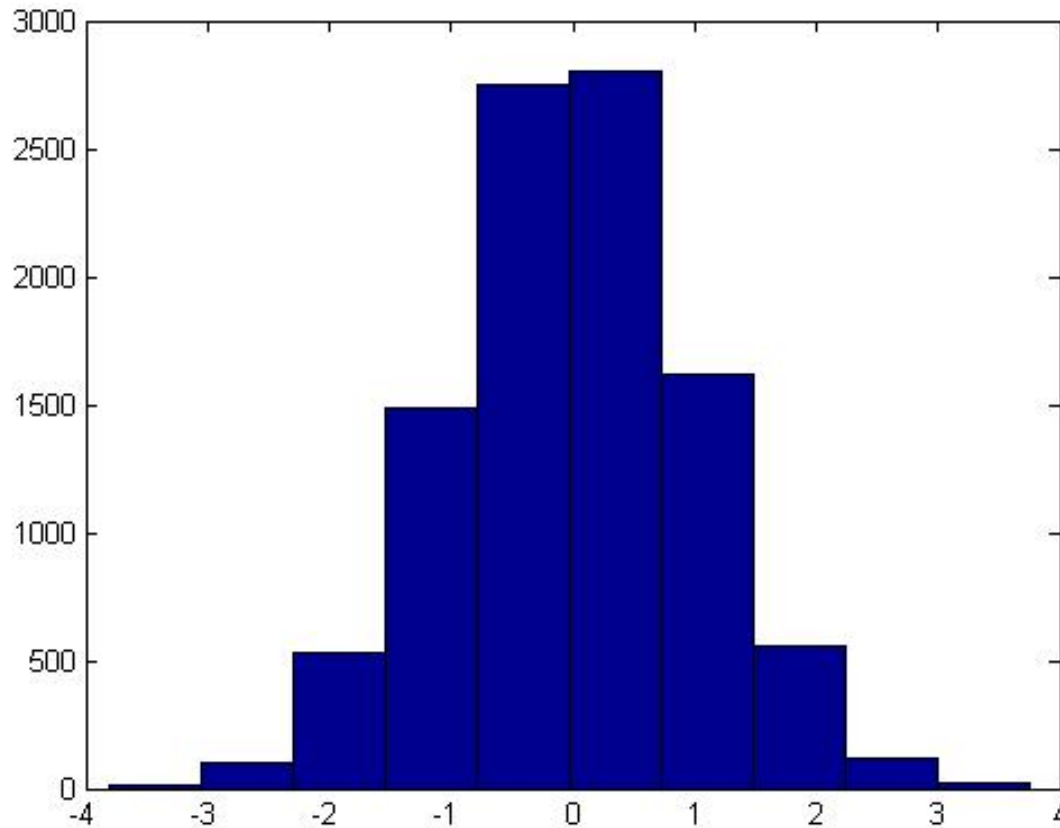
Combinación de gráficas “bar–plot”

```
>> con = [515 420 370 250 135 120 60 20];  
>> tem = [29 23 27 25 20 23 23 27];  
>> dia = 0:5:35;  
>> bar(dia,tem)  
>> xlabel('Día')  
>> ylabel('Temperatura (^{o}C)')  
>> h1 = gca;  
>> h2 = axes('Position',get(h1,'Position'));  
>> plot(dia,con,'LineWidth',3)  
>> set(h2,'YAxisLocation','right', ...  
      'Color','none','XTickLabel',[])  
>> set(h2,'XLim',get(h1,'XLim'),'Layer','top')  
>> text(11,380,'Concentracion',...  
      'Rotation',-55,'FontSize',16)  
>> ylabel('Concentracion'), colormap cool  
>> title('Combinación de gráficas', 'FontSize',16)
```



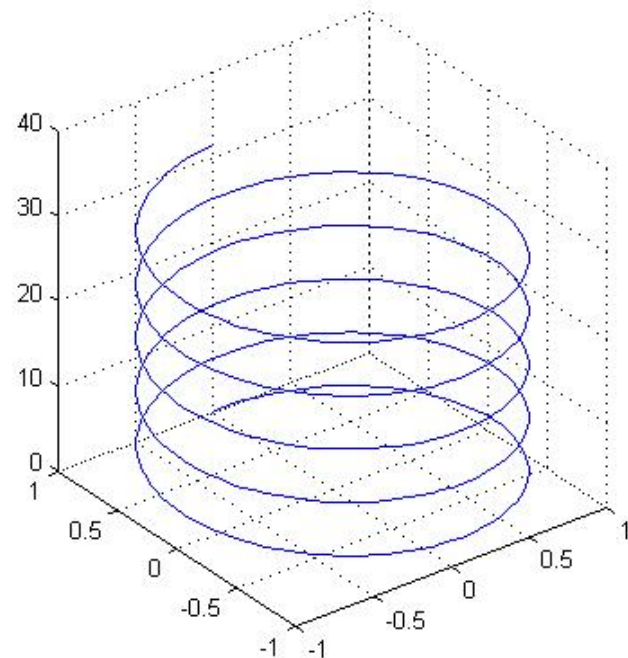
Histogramas

```
>> y = randn(10000,1);  
>> hist(y)
```



Gráficas en 3D

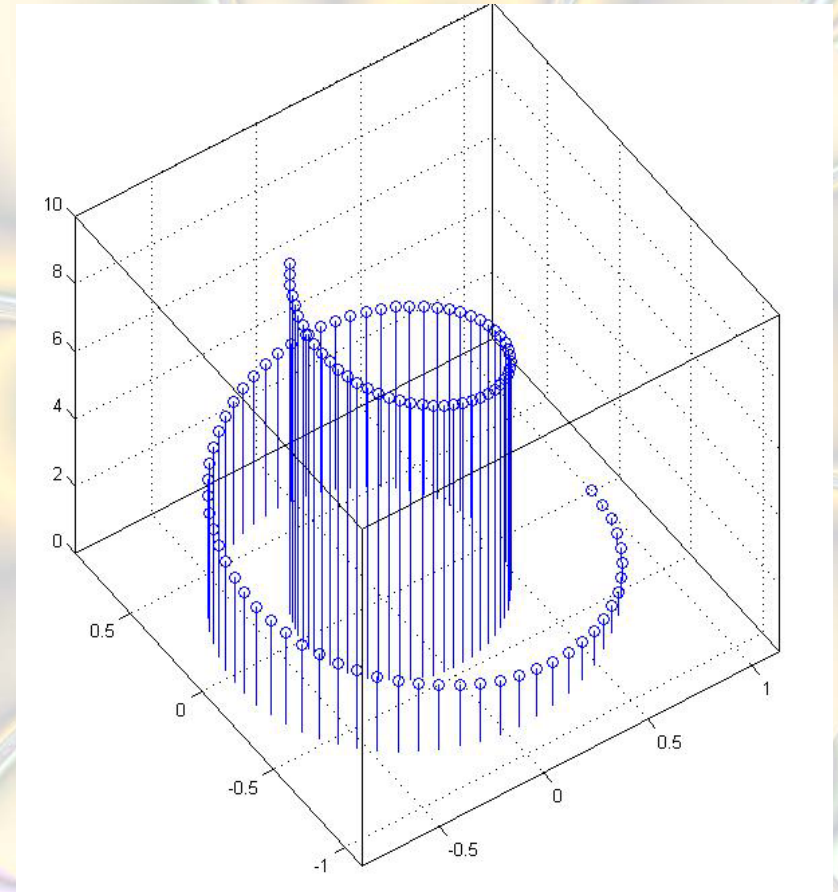
```
>> t = 0:pi/50:10*pi;  
>> plot3(sin(t),cos(t),t)  
>> axis square; grid on  
  
>> [X,Y] = meshgrid([-2:0.1:2]);  
>> Z = X.*exp(-X.^2-Y.^2);  
>> plot3(X,Y,Z)  
>> grid on
```



Gráficas en 3D

Transformada de Laplace:

```
>> t = 0:0.1:10;  
>> s = 0.1+i;  
>> y = exp(-s*t);  
>> stem3(real(y),imag(y),t)  
>> hold on  
>> plot3(real(y),imag(y),t,'k')  
>> hold off  
>> xlabel('Real')  
>> ylabel('Imaginario')  
>> zlabel('Magnitud')
```



Gráficos en coordenadas polares

```
>> t = 0:0.05:pi+0.01;  
>> y = sin(4*t).*exp(-0.3*t);  
>> polar(t,y)  
>> title('Gráfica Polar')  
>> grid
```

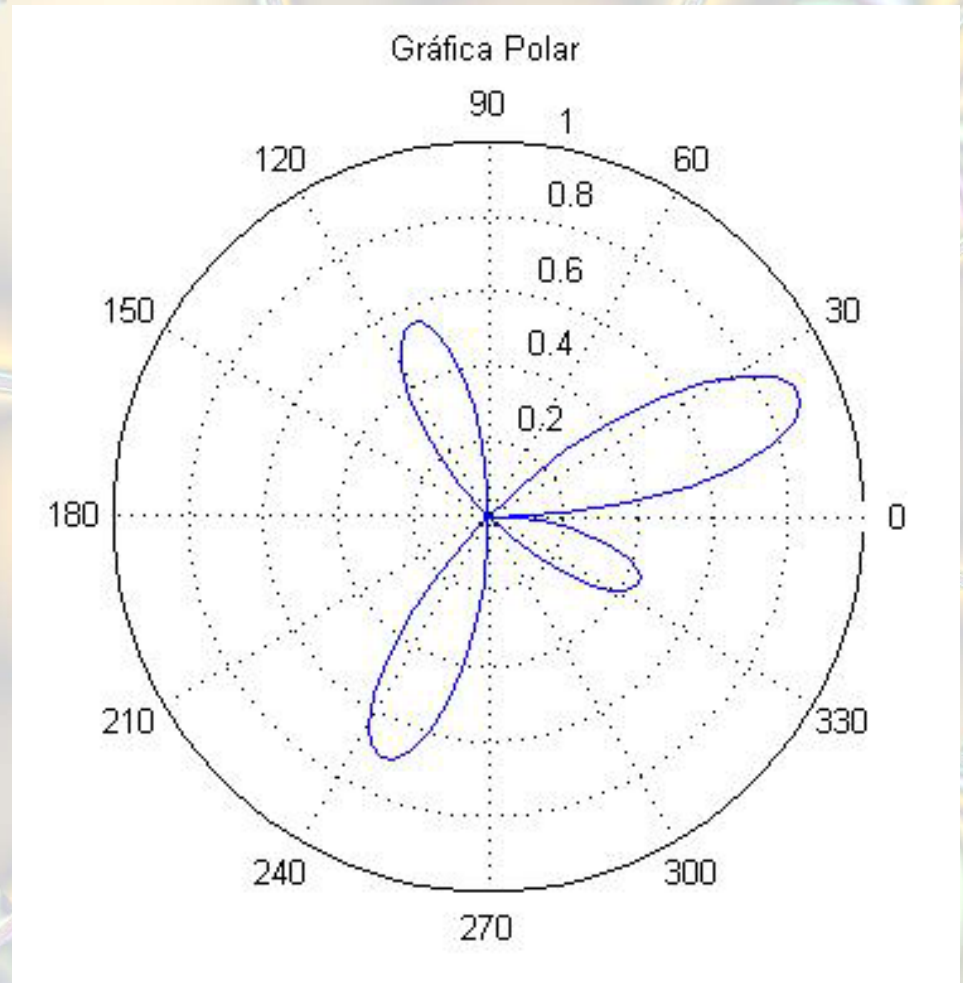


Gráfico de Compás

```
>> wdir = [45 90 90 45 360 335 360 ...  
           270 335 270 335 335];  
>> knots = [6 6 8 6 3 9 6 8 9 10 14 12];  
>> rdir = wdir * pi/180;  
>> [x,y] = pol2cart(rdir,knots);  
>> compass(x,y)  
>> desc = {'Velocidad y Magnitud del Viento', ...  
           'Aeropuerto "Logan"', ...  
           'De Nov. 3 a las 1800 a', ...  
           'Nov. 4 a las 0600'};  
>> text(-28,15,desc)
```

Velocidad y Magnitud del Viento
Aeropuerto "Logan"
De Nov. 3 a las 1800 a
Nov. 4 a las 0600

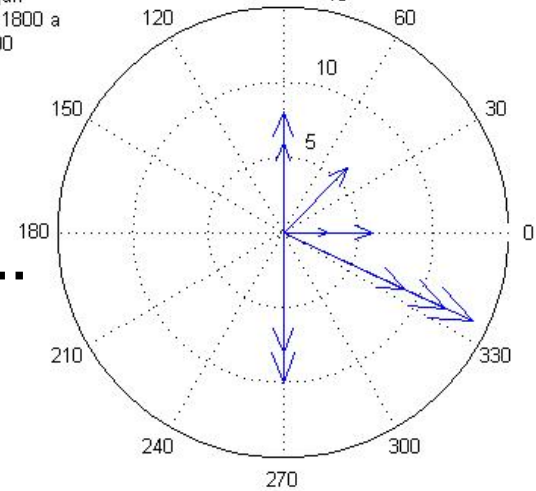
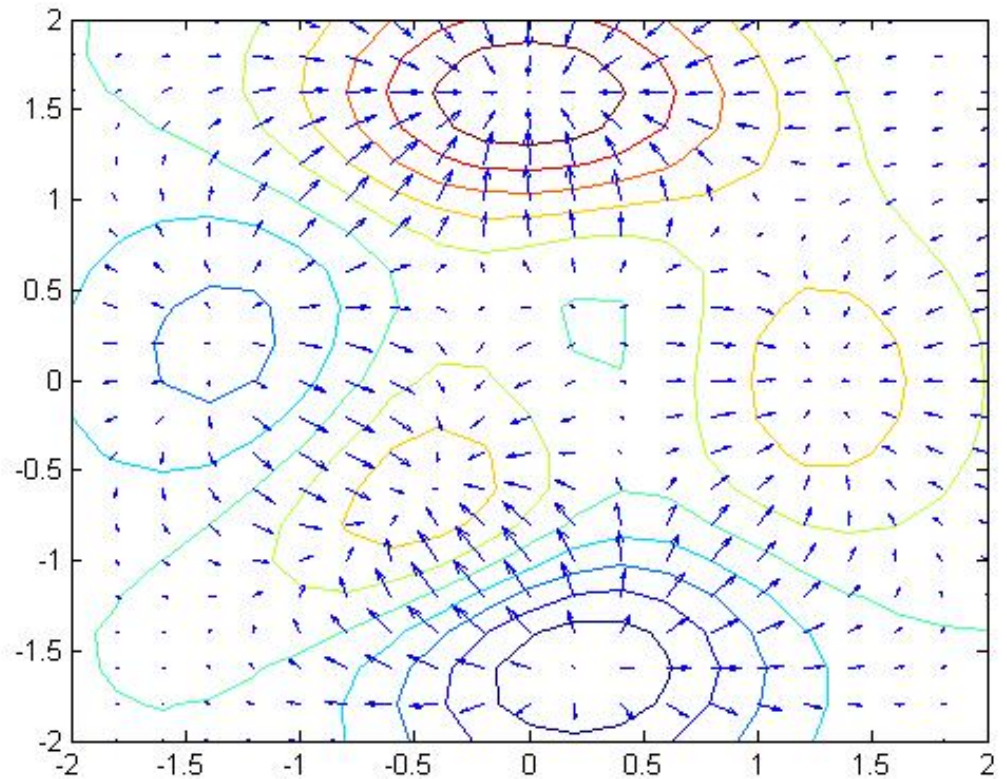


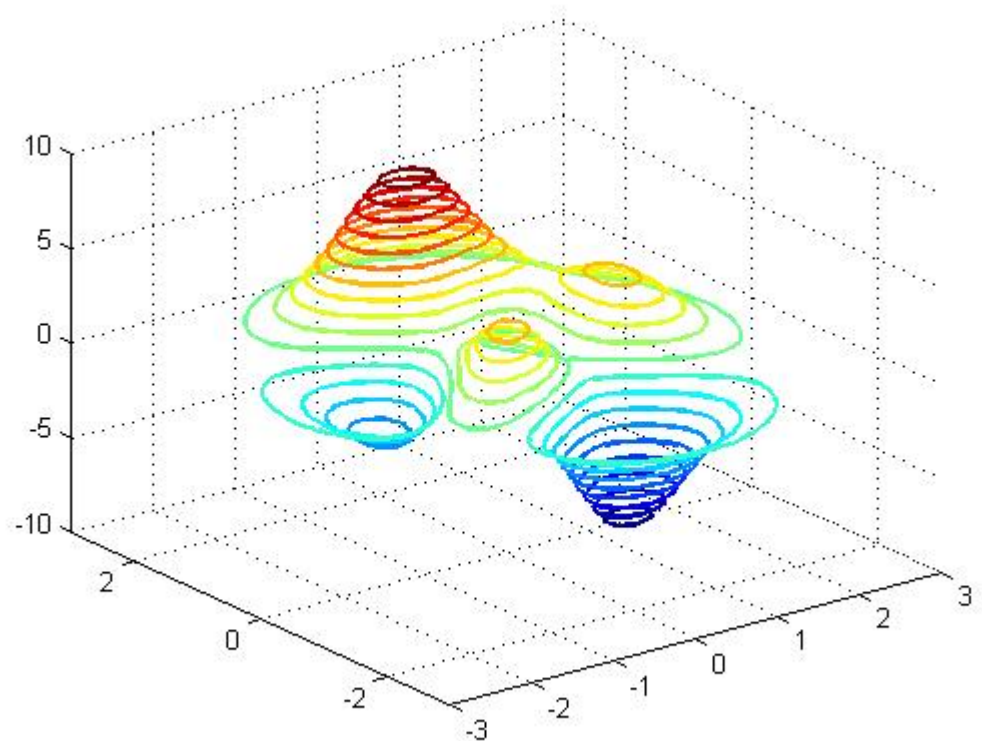
Gráfico de Contorno y Vectorial

```
>> n = -2.0:2.0;
>> [X,Y,Z] = peaks(n);
>> contour (X,Y,Z,10)
>> [U,V] = gradient(Z,.2);
>> hold on
>> quiver(X,Y,U,V)
>> hold off
```



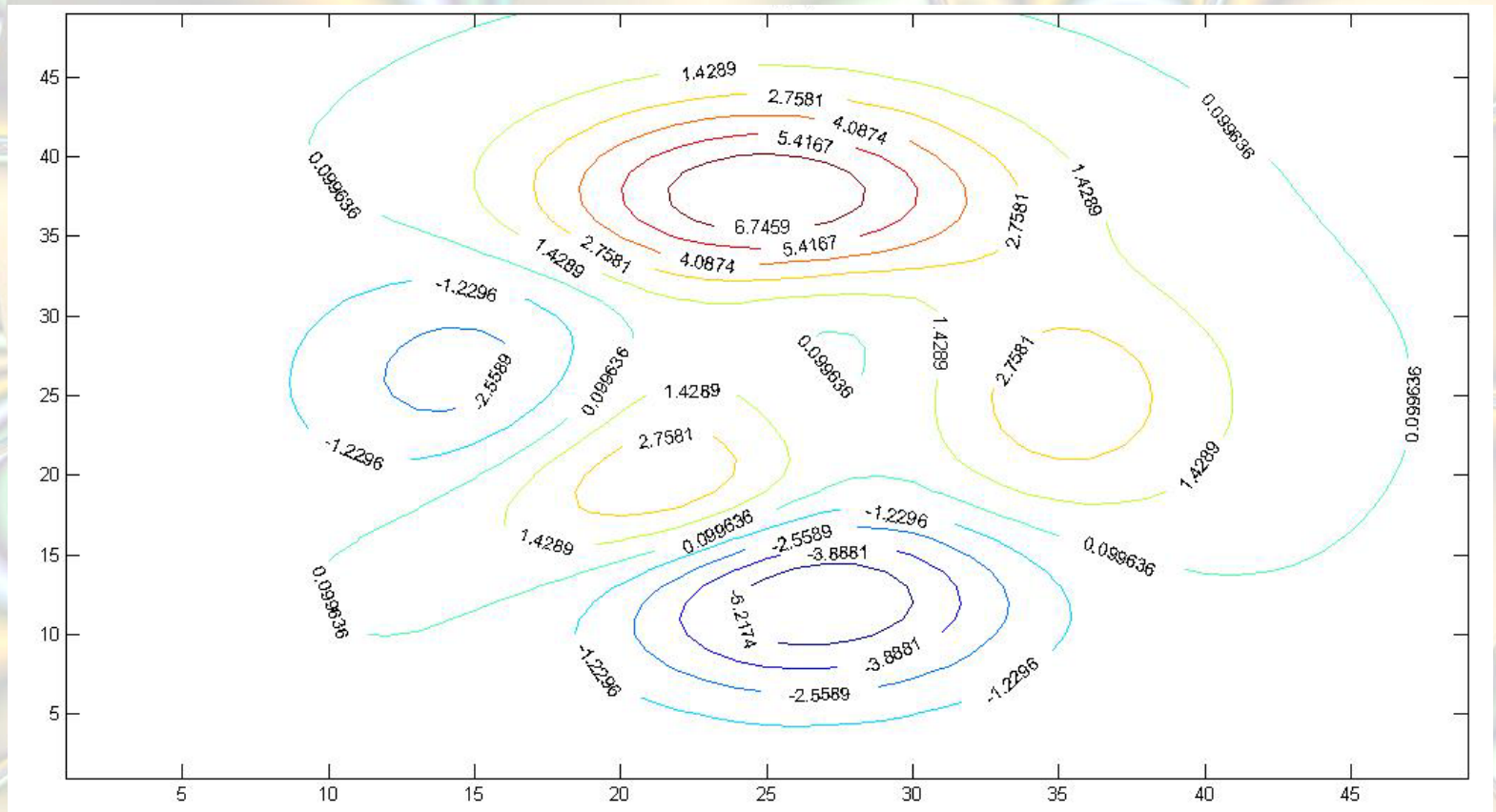
Contornos 3D

```
>> [X,Y,Z] = peaks;  
>> contour3(X,Y,Z,20)  
>> h = findobj('Type','patch');  
>> set(h,'LineWidth',1.5)
```



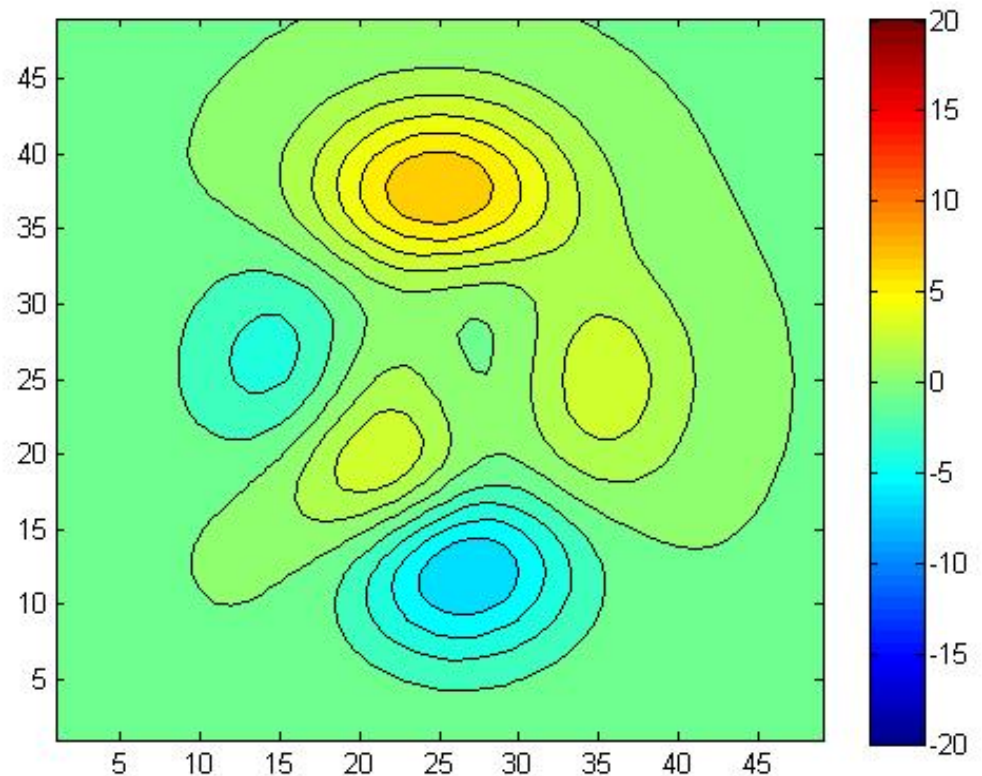
Mas Contornos ...

```
>> Z = peaks;  
>> [C,h] = contour(Z,10);  
>> clabel(C,h)
```



Mas Contornos ...

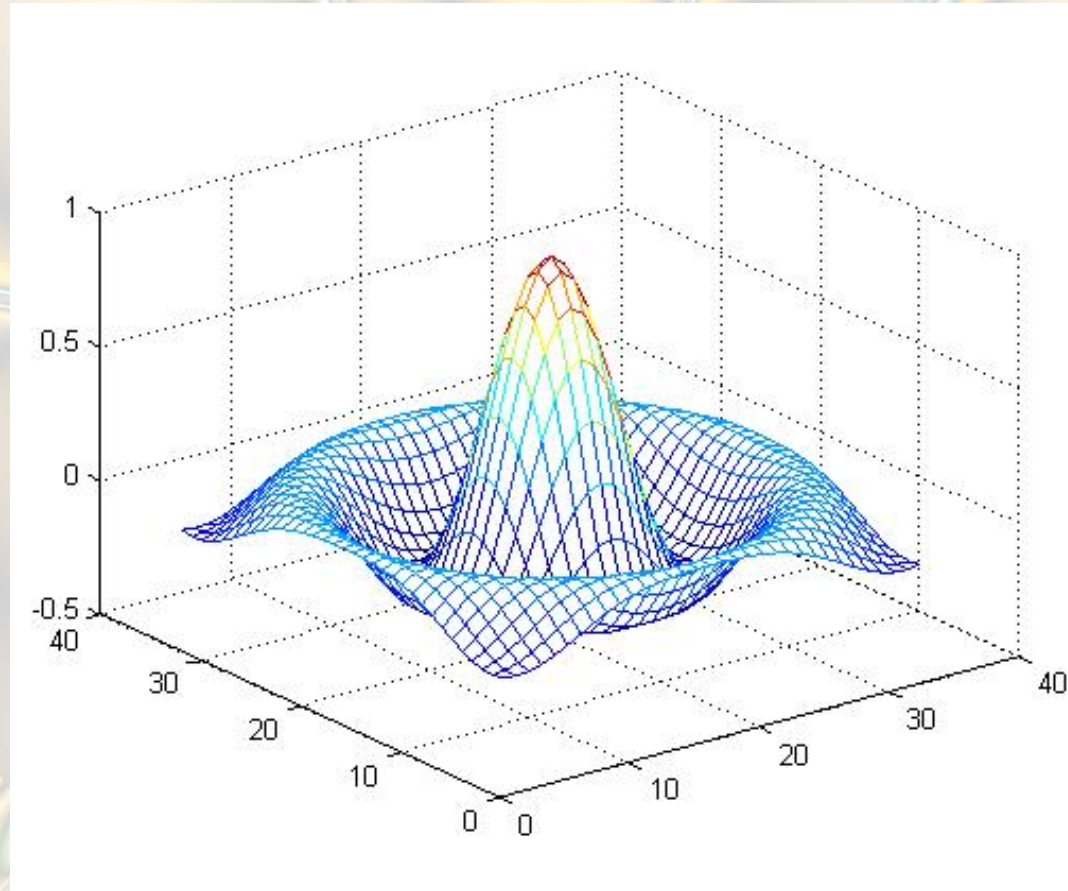
```
>> Z = peaks;  
>> [C,h] = contourf(Z,10);  
>> caxis([-20 20])  
>> colorbar
```



Superficies 3D

Comando “mesh”:

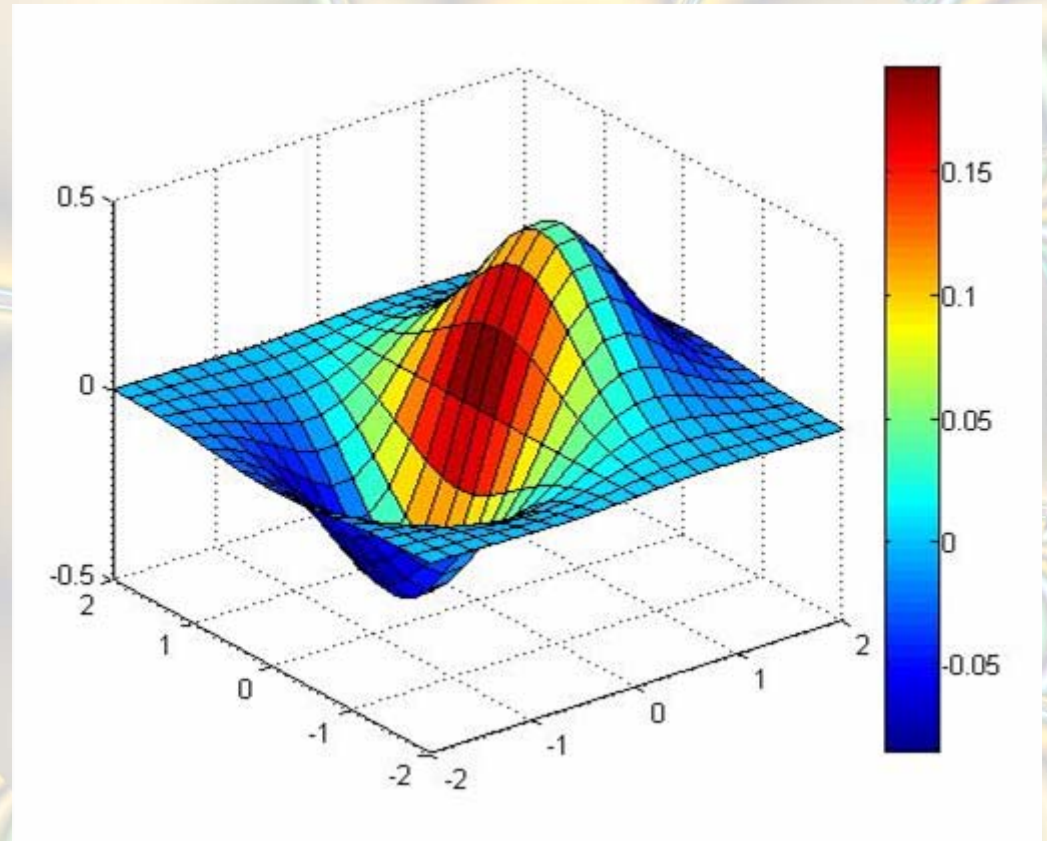
```
>> [X,Y] = meshgrid(-8:.5:8);  
>> R = sqrt(X.^2 + Y.^2) + eps;  
>> Z = sin(R)./R;  
>> mesh(Z)
```



Superficies 3D

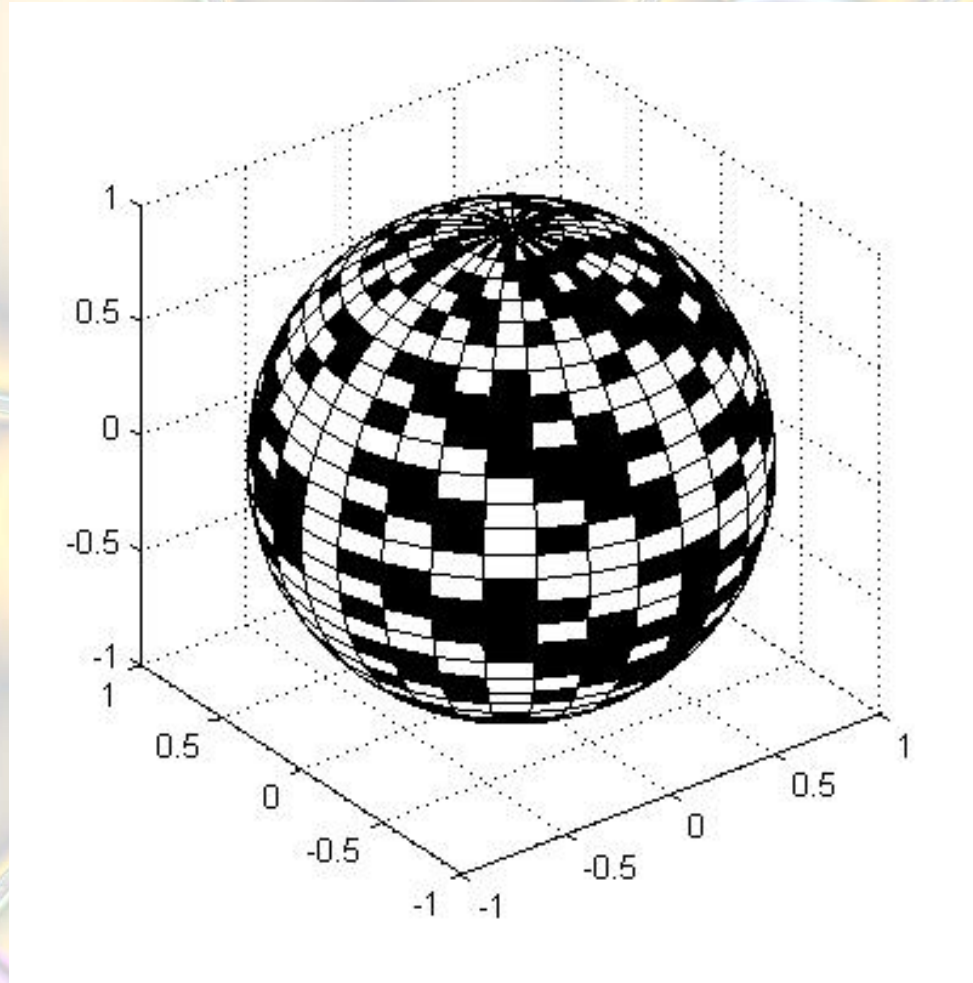
Comando “surf”:

```
>> [x,y] = meshgrid([-2:2:2]);  
>> Z = x.*exp(-x.^2-y.^2);  
>> surf(x,y,Z,gradient(Z))  
>> colorbar
```



Superficies Paramétricas

```
>> k = 5;  
>> n = 2^k-1;  
>> theta = pi*(-n:2:n)/n;  
>> phi = (pi/2)*(-n:2:n)'/n;  
>> X = cos(phi)*cos(theta);  
>> Y = cos(phi)*sin(theta);  
>> Z = sin(phi)*ones(size(theta));  
>> colormap([0 0 0;1 1 1])  
>> C = hadamard(2^k);  
>> surf(X,Y,Z,C)  
>> axis square
```



Conclusiones

La calidad en los gráficos depende del artista, hay infinidad de comandos para la modificación de estos:

1. Cambiar tipos de líneas.
2. Cambiar los colores de las líneas.
3. Cambiar el color de fondo.
4. Agregar grilla.
5. Agregar títulos y rótulos.
6. Agregar Leyenda.
7. Graficar una función definida previamente.
8. Diagramas de barras y histogramas



¡Gracias por su atención!