CAPITULO I. GRÁFICAS EN OCTAVE

2.1 Gráficas en el plano bidimensional

A través de los siguientes comandos, Octave puede crear objetos gráficos en $\mathbb{R}^2$ y $\mathbb{R}^3$ e incluso artísticos utilizando expresiones matemáticas.

Comandos	Definición
plot(x_1, x_2)	La función plot dibuja un conjunto de puntos (x_1, x_2) en un sistema bidimensional.
bar(x_1, x_2)	La función bar dibuja una secuencia de barras verticales, donde x_2 representa las frecuencias y x_1 define los espacios en el eje horizontal.

Objetivo:

Utilizar funciones de octave para elaborar gráficos en el plano y el espacio.

CONTENIDO:

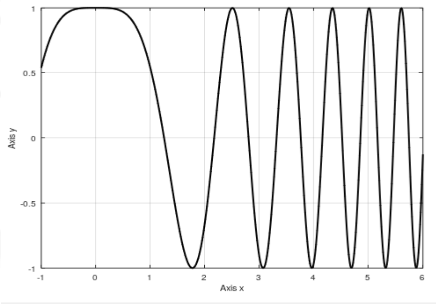
- ✓ Gráfica de múltiples curvas.
- ✓ Gráficas especiales en el plano bidimensional.
- ✓ Gráfica de superficies.
- ✓ Gráfica en coordenadas polares.
- ✓ Gráfica de campos vectoriales 2D y 3D con flechas.

Capítulo 1

barh(x_1, x_2)	La función barh dibuja una secuencia de barras horizontales, donde x_2 son las frecuencias y x_1 son los espacios en el eje horizontal
stem(x_1, x_2)	La función stem dibuja una secuencia de bastones en forma vertical, donde x_2 son las frecuencias y x_1 son los espacios en el eje horizontal.
stairs(x_1, x_2)	La función stairs dibuja una curva de forma escalonada.
polar(x_1, x_2)	La función polar dibuja curvas en el sistema coordenado polar.
pie(x_1)	La función pie grafica sectores en correspondencia al vector x_1.
rose(x_1)	La función rose dibuja histogramas en correspondencia al vector x_2.

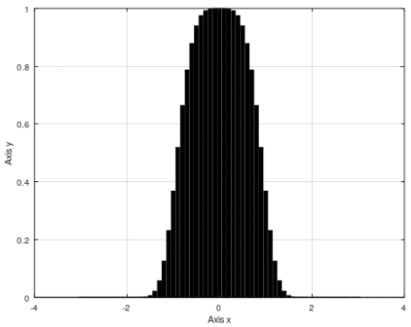
Ejemplo 1.1

```
% line:
x1=-1:0.01:6;
x2=cos(x1.^2);
plot(x1,x2,'k',
'linewidth',1)
grid
xlabel('Axis x')
ylabel('Axis y')
```



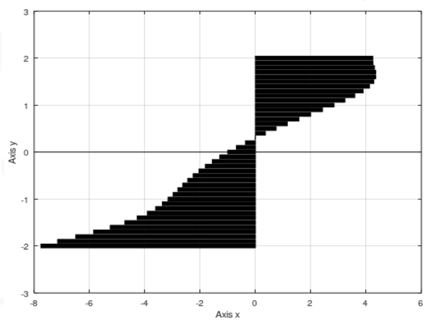
Ejemplo 1.2

```
% bar:
x1=-3:0.1:3;
x2=exp(-x1.^4);
bar(x1,x2,'k')
grid
xlabel('Axis x')
ylabel('Axis y')
```



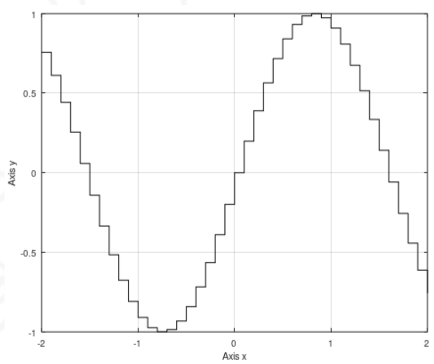
Ejemplo 1.3

```
% barh:
x1=-2:0.1:2;
x2=sin(x1.^2)
+3*x1-1;
barh(x1,x2,'k');
grid
xlabel('Axis x')
ylabel('Axis y')
```



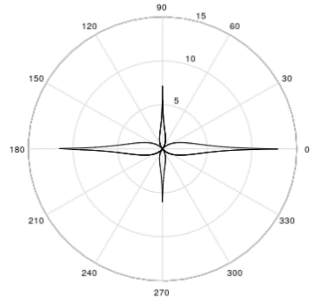
Ejemplo 1.4

```
% stairs:
x1=0:0.1:15;
x2=sin(2*x1);
stairs(x1,x2,'k')
grid
xlabel('Axis x')
ylabel('Axis y')
```



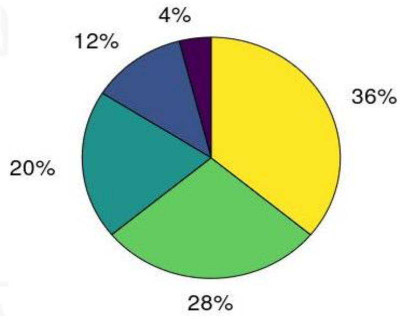
Ejemplo 1.5

```
% polar:
t=0:0.01:3*pi;
x1=log(sin(t).*tan(t));
polar(t,x1,'k')
```



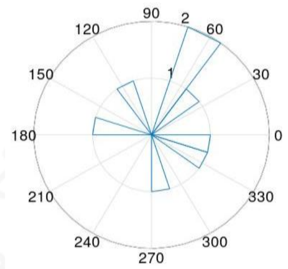
Ejemplo 1.6

```
% pie:
x1=1:2:9;
pie(x1);
```



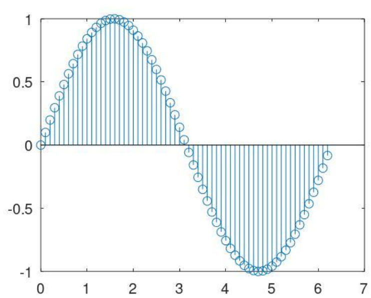
Ejemplo 1.7

```
% rose:
x1=[1 3 6 2 7 1 5 12];
rose(x1);
```



Ejemplo 1.8

```
% stem:
x1=0:0.1:2*pi;
x2=sin(x1);
stem(x1,x2);
```

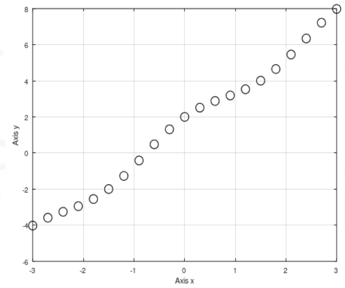


Octave gráfica expresiones matemáticas utilizando diferentes características, como se muestra a continuación.

Caracteres	Definición
+	Más
.	Punto
x	Equis
o	Circunferencia
*	Asterisco

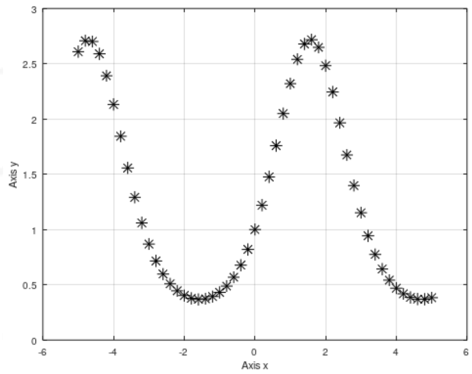
Ejemplo 1.9

```
x1=-3:0.3:3;
x2=cos(x1).^2+2*x1+1;
plot(x1,x2,'ok')
grid
xlabel('Axis x')
ylabel('Axis y')
```



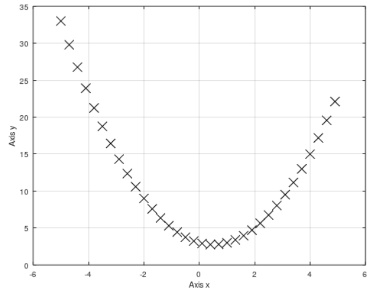
Ejemplo 1.10

x1=-5:0.2:5; x2=exp(sin(x1)); plot(x1,x2,'*k'); grid;
xlabel('Axis x'); ylabel('Axis y');



Ejemplo 1.11

```
x1=-5:0.3:5;
x2=x1.^2-x1+3;
plot(x1,x2,'xk')
grid
xlabel('Axis x')
ylabel('Axis y')
```



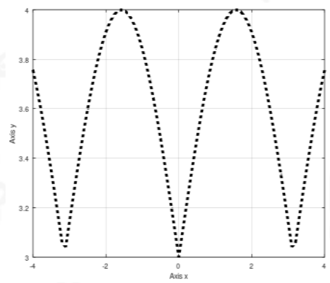
Octave posee muchas variedades en el diseño de gráficas como también una variedad de colores.

Líneas	Definición
--	Guiones
-	Continua
-.	Guiones y punto
:	Punteada

Color	Definición
y	Amarillo
r	Rojo
w	Blanco
b	Azul
g	Verde
k	Negro
c	Turquesa
m	Magenta

Ejemplo 1.12

```
x1=-4:0.1:4;
x2=abs(sin(x1))+3;
plot(x1,x2,'k:','linewid',2)
grid
xlabel('Axis x')
ylabel('Axis y')
```



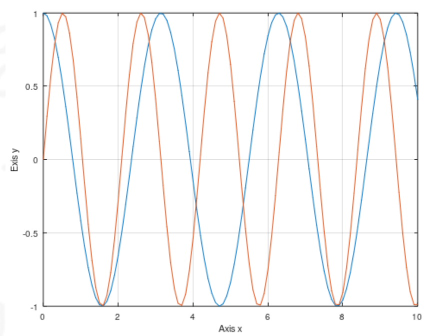
1.2 Gráfico de múltiples curvas

Con el comando **plot** de Octave se puede graficar una o varias curvas simultáneamente, estableciendo por defecto dos colores distintos como se muestra a continuación:

El comando **plot(x1, y1, x1, y2):** esta función dibuja dos curvas con diferente color.

Ejemplo 1.13

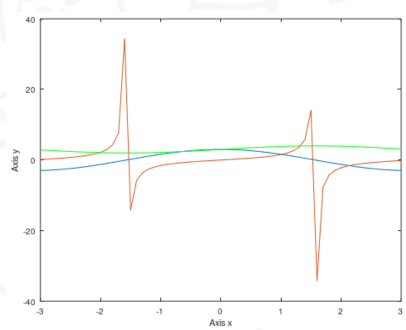
```
x1=0:0.1:10;
y1=cos(2*x1);
y2=sin(3*x1);
plot(x1,y1,x1,y2)
grid
xlabel('Axis x');
ylabel('Exis y')
```



Observación: Después de haber mostrado ejemplos con una variedad de gráficos, a continuación se mostrarán ejemplos donde se interceptan varias curvas en la misma figura, escribiendo la función **hold on**, en el penúltimo **plot** como se muestra a continuación.

Ejemplo 1.14

```
x1=-3:0.1:3;
x2=3*cos(x1);
x3=tan(x1);
plot(x1,x2,x1,x3)
hold on
w=sin(x1)+3;
plot(x1,w,'g')
xlabel('Axis x')
ylabel('Axis y')
```



1.3 Etiquetas en los ejes coordenados

Octave posee una variedad de comandos para colorear textos, etiquetar los ejes, colorear título, como se muestra a continuación:

xlabel: etiqueta al eje horizontal.

ylabel: etiqueta al eje vertical.

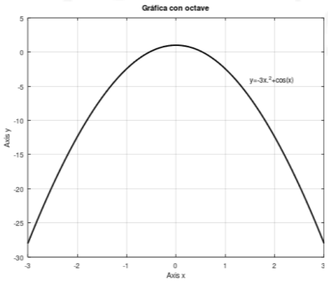
grid: Grafica rejillas (líneas horizontales y verticales) en la gráfica hecha con plot.

title: Coloca título a la gráfica.

text (x, y, 'cadena'): adiciona una cadena a la gráfica en el punto de coordenadas (x, y).

Ejemplo 1.15

```
x1=-3:0.05:3;
x2=-3*x1.^2+cos(x1);
plot(x1,x2,'k','linewid',1);
grid
text(1.5,-4,'x2=-
3x1.^2+cos(x1)');
title('Gráfica con octave');
xlabel('Axis x')
ylabel('Axis y')
```



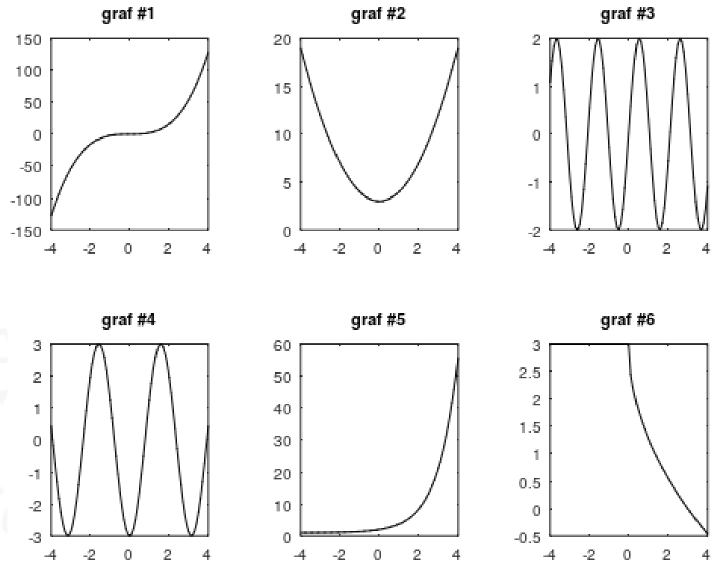
1.4 Gráficas en múltiples ejes

Octave permite la construcción de múltiples ejes en una sola pantalla que produce múltiples gráficos.

subplot (m_fil, n_col, k): Particiona la figura en m por n gráficas, siendo $k = m * n$, el número de gráficas.

Ejemplo 1.16

```
t=-4:0.1:4; x1=2*t.^3; x2=t.^2+3; x3=2*sin(3*t);

u1=-3*cos(2*t); u2=exp(t)+1; u3=3-sqrt(3*t);

subplot(2,3,1), plot(t,x1,'k'),title('graf #1');

subplot(2,3,2), plot(t,x2,'k'),title('graf #2');

subplot(2,3,3), plot(t,x3,'k'),title('graf #3');

subplot(2,3,4), plot(t,u1,'k'),title('graf #4');

subplot(2,3,5), plot(t,u2,'k'),title('graf #5');

subplot(2,3,6), plot(t,u3,'k'),title('graf #6');
```



1.5 Gráficos en el plano

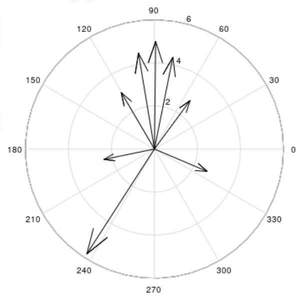
En el siguiente cuadro se mencionan los comandos que Octave tiene incorporados para dibujar funciones reales en el espacio bidimensional.

Comandos	**Definición**
feather(x_1, x_2)	La función feather dibuja vectores en el plano bidimensional con origen en el eje x_1. Además, su longitud y dirección vienen dadas por el módulo de x_2.
loglog(x_1, x_2)	La función loglog dibuja una curva en el plano bidimensional con escala logarítmica en los ejes $x_1 x_2$.
ezplot('f',[x_1, x_2])	La función ezplot dibuja en el plano bidimensional la función real F en el intervalo [x, y] y el título lo pone por defecto.
semilogy(x_1, x_2)	La función semilogy dibuja una curva el plano bidimensional. Por defecto presenta escala logarítmica en la segunda componente y escala normal en la primera componente.
semilogx(x_1, x_2)	La función semilogx dibuja una curva en el plano bidimensional con escala logarítmica en la primera componente y escala normal en la segunda componente.
fplot('F', [x_1, x_2])	La función fplot dibuja la función F en el intervalo [x_1, x_2].
fill(x_1, x_2, h)	La función fill grafica una región

	poligonal en el plano bidimensional, donde parámetro h contiene el color a graficar.
compass(x_1, x_2)	La función compass dibuja vectores tomando como punto de partida el origen de coordenadas.

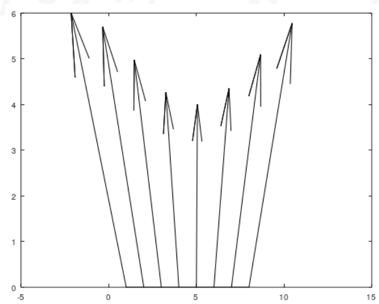
Ejemplo 1.17

```
x1=-pi:0.8:pi;
x2=5-x1.^2;
compass(x1,x2,'k')
```



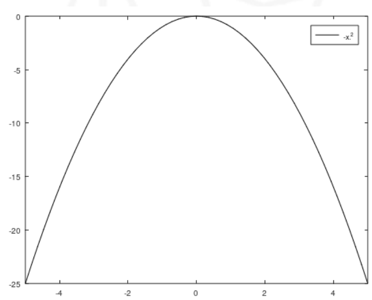
Ejemplo 1.18

```
x1=-pi:0.8:pi;
x2=5-cos(x1);
feather(x1,x2,'k')
```



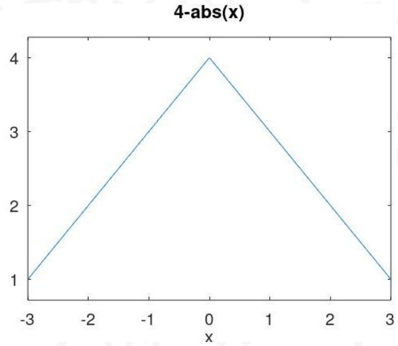
Ejemplo 1.19

```
fplot('-x.^2',[-5,5],'k')
```



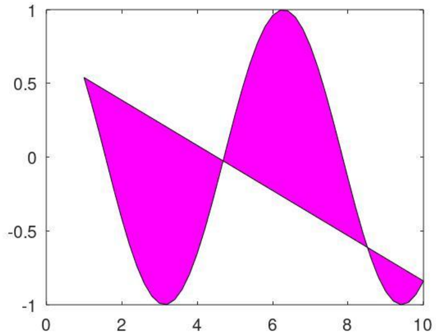
Ejemplo 1.20

```
ezplot('4-abs(x)',
[-3,3])
```



Ejemplo 1.21

```
x=1:0.2:10;
y=cos(x);
fill(x,y,'m');
```



1.6 Gráficos en el sistema cartesiano tridimensional

Los dibujos en el sistema cartesiano tridimensional, de rectas y superficies, se pueden implementar en Octave como se muestra a continuación:

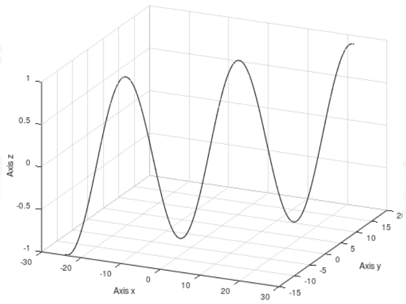
Comandos	**Definición**
plot3(x_1, x_2, x_3)	La función plot3 dibuja los puntos (x_1, x_2, x_3) en el sistema cartesiano tridimensional.
fill3(x_1, x_2, x_3, c)	La función fill3 dibuja la región poligonal cuyas coordenadas de los vértices son los elementos de los vectores columna x_1, x_2, x_3. El parámetro c contiene el color a graficar.
meshgrid(x_1, x_2)	La función meshgrid construye matrices bidimensionales a partir de los vectores x_1 e x_2, para poder construir la representación gráfica de una superficie explicita z=f(x_1, x_2).

mesh(x_1, x_2, x_3)	La función mesh construye una representación gráfica de una superficie explicita $x_3 = f(x_1, x_2)$ con vectores x_1, x_2, x_3.
meshc(x_1, x_2, x_3)	La función meshc construye una representación gráfica de una superficie explicita $x_3 = f(x_1, x_2)$ con los vectores x_1, x_2, x_3 ; proyectando las curvas de nivel en el plano $x_1 x_2$.
meshz(x_1, x_2, x_3)	La función meshz construye la representación gráfica de una superficie explicita $x_3 = f(x_1, x_2)$ con los arreglos x_1, x_2, x_3; cerrando la gráfica con las fronteras del dominio.
surf(x_1, x_2, x_3)	La función surf construye la representación gráfica de una superficie explicita $x_3 = f(x_1, x_2)$ con los vectores x_1, x_2, x_3; pintando cada una de las celdas.
surfc(x_1, x_2, x_3)	La función surfc construye la representación gráfica de una superficie explicita $x_3 = f(x_1, x_2)$ con los vectores x_1, x_2, x_3; proyectando las curvas de nivel en el plano xy.
surfl(x_1, x_2, x_3)	La función surfl construye la representación gráfica de una superficie explicita $x_3 = f(x_1, x_2)$ con los vectores x_1, x_2, x_3 ; considerando una iluminación en formato básico.
waterfall(x_1, x_2, x_3)	La función waterfall construye la representación gráfica de una superficie explicita $x_3 = f(x_1, x_2)$ en forma de cascada.
contour(x_1, x_2, x_3)	La función contour construye la representación gráfica de curvas de nivel de la superficie explicita $x_3 = f(x_1, x_2)$.

Los comandos mencionados anteriormente se utilizan para crear gráficos en tres dimensiones.

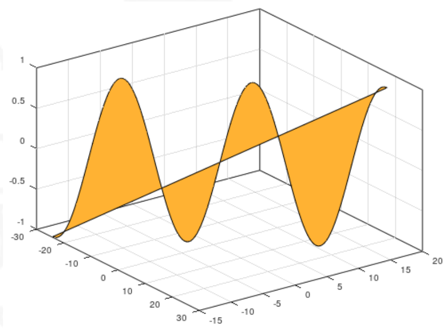
Ejemplo 1.22

```
t=-8:0.1:8;
x1=3*t; x2=1+2*t;
x3=sin(t);
plot3(x1,x2,x3,'k');
grid
xlabel('Axis x')
ylabel('Axis y')
zlabel('Axis z')
```



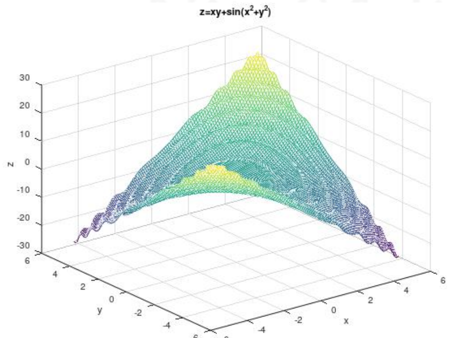
Ejemplo 1.23

```
t=-8:0.1:8;
x=3*t;
y=1+2*t;
z=sin(t);
k=[1 0.7 0.2];
fill3(x,y,z,k);
grid
```



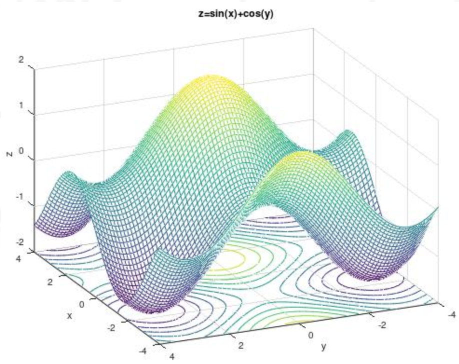
Ejemplo 1.24

```
[x1, x2]=meshgrid(-5:0.1:5);
x3=x1.*x2+sin(x1.^2+x2.^2);
mesh(x1,x2,x3); title('z=xy+sin(x^2+y^2)');
xlabel('x');ylabel('y');zlabel('z');
```



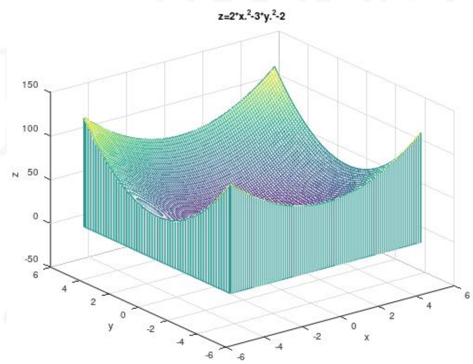
Ejemplo 1.25

```
[x1, x2]=meshgrid(-4:0.1:4); x3=sin(x1)+cos(x2); meshc(x1,x2,x3)
title('z=sin(x)+cos(y)'); xlabel('x');ylabel('y');zlabel('z');
```



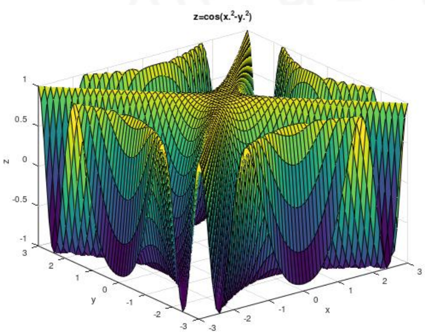
Ejemplo 1.26

[x1, x2]=meshgrid(-5:0.1:5); x3=2*x1.^2+3*x2.^2-2; meshz(x1,x2,x3); title('z=2*x.^2-3*y.^2-2'); xlabel('x');ylabel('y');zlabel('z');



Ejemplo 1.27

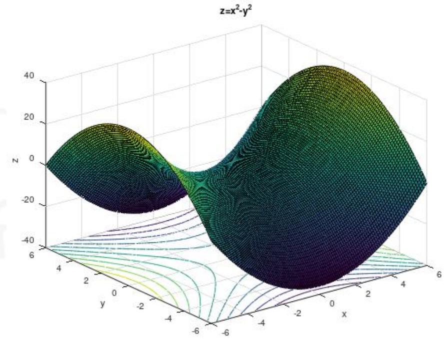
[x1,x2]=meshgrid(-3:0.1:3); x3=cos(x1.^2-x2.^2); surf(x1,x2,x3); title('z=cos(x.^2-y.^2)'); xlabel('x');ylabel('y');zlabel('z');



Ejemplo 1.28

[x1, x2]=meshgrid(-6:0.1:6); x3=x1.^2-x2.^2; surfc(x1,x2,x3);

title('z=x^2-y^2'); xlabel('x');ylabel('y');zlabel('z');

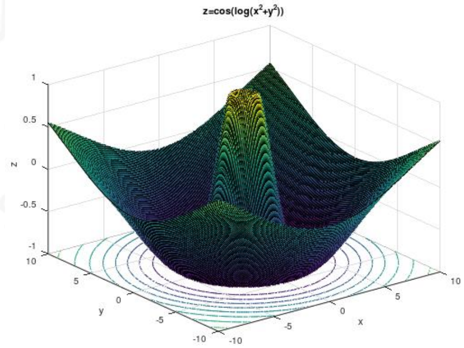


Ejemplo 1.29

[x1, x2]=meshgrid(-10:0.1:10); x3=cos(log(x1.^2+x2.^2));

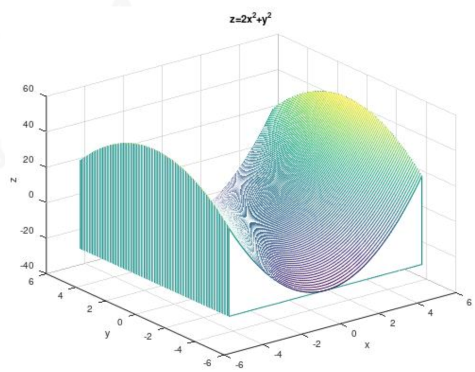
surfc(x1,x2,x3); title('z=cos(log(x^2+y^2))');

xlabel('x');ylabel('y');zlabel('z');



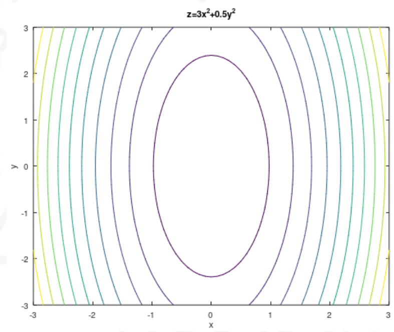
Ejemplo 1.30

[x1, x2]=meshgrid(-5:0.1:5); x3=2*x1.^2-x2.^2; waterfall(x1,x2,x3); title('z=2x^2+y^2');
xlabel('x');ylabel('y');zlabel('z');



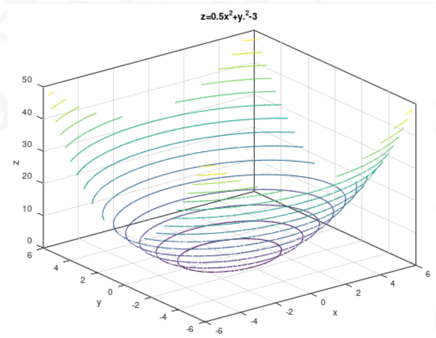
Ejemplo 1.31

```
[x1, x2]=meshgrid(-3:0.1:3); x3=3*x1.^2+0.5*x2.^2; contour(x1,x2,x3);
title('z=3x^2+0.5y^2'); xlabel('x'); ylabel('y'); zlabel('z');
```



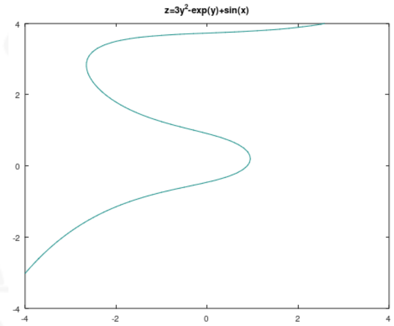
Ejemplo 1.32

```
[x1, x2]=meshgrid(-6:0.1:6); x3=0.5*x1.^2+x2.^2-3; contour3(x1,x2,x3,15);
title('z=0.5x^2+y.^2-3'); xlabel('x');ylabel('y');zlabel('z');
```



Ejemplo 1.33

```
[x1, x2]=meshgrid(-4:0.1:4);

x3=3*x2.^2-exp(x2)+sinh(x1); contour(x1,x2,x3,[0,0]);title('z=3y^2exp(y)+sin(x)');
```



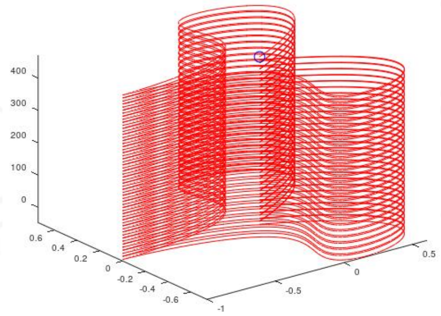
1.7 Gráficos especiales

La tabla muestra los comandos que Octave puede usar para representar superficies.

Comandos	Definición
cylinder(f)	La función cylinder construye la representación gráfica de una superficie de revolución.
sphere	Dibuja la esfera unitaria usando 20x20 caras.
ribbon(x, y, z, c)	La función ribbon construye la representación gráfica de una superficie explicita z=f(x, y) en el espacio tridimensional. Además, c nos da el ancho de la superficie.
stem3(x, y, z)	La función stem3 construye la representación gráfica de puntos de una curva paramétrica x=x(t), y=y(t), z=z(t), representándolo con líneas verticales tipo bastones.
pie3(x)	La función pie3 construye la representación gráfica de sectores en el espacio tridimensional.
quiver3(x, y, z, u, v, w, c)	La función quiver3 construye la representación gráfica de vectores de componentes (u, v, w) en los puntos (x, y, z), además c indica el tamaño de los vectores.
comet3(x, y, z)	La función comet construye la representación gráfica de una curva paramétrica, además genera un movimiento dicha curva.

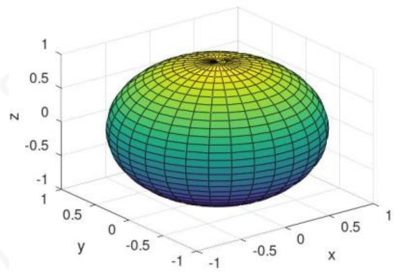
Ejemplo 1.34

```
t=-5*pi:pi/150:50*pi; x=(sin(3*t).^2).*cos(2*t); y=(sin(2*t).^2).*cos(t); z=3*t;
comet3(x,y,z)
```



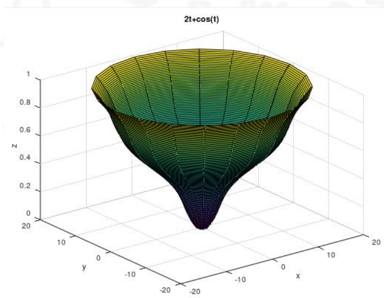
Ejemplo 1.35

```
sphere(20)
xlabel('x');
ylabel('y');
zlabel('z');
```



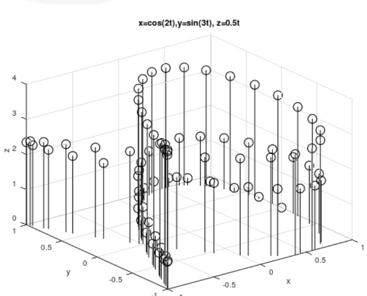
Ejemplo 1.36

```
t=0:0.1:10;
f=2*t+cos(t);
cylinder(f);
title('2t+cos(t)');
xlabel('x');
ylabel('y');
zlabel('z');
```



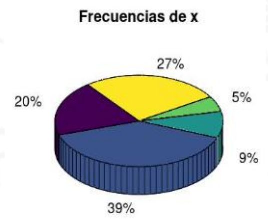
Ejemplo 1.37

t=0:0.1:8; x=cos(2*t); y=sin(3*t); z=0.5*t; stem3(x,y,z,'k'); title('x=cos(2t),y=sin(3t), z=0.5t'); xlabel('x'); ylabel('y'); zlabel('z');



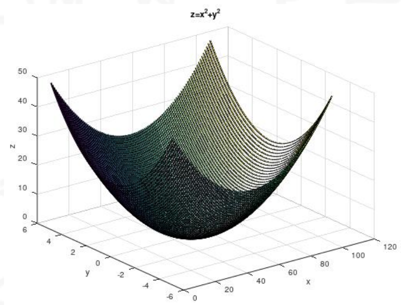
Ejemplo 1.38

```
x=[25 50 12 7 34];
pie3(x);
title('Frecuencias de x');
xlabel('x');
ylabel('y');
```



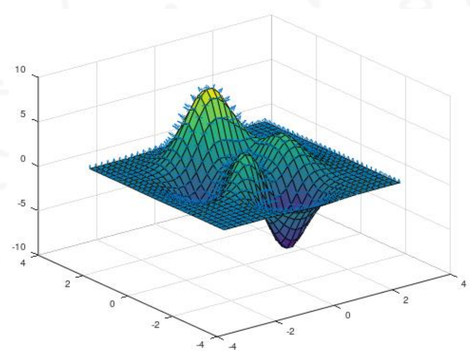
Ejemplo 1.39

```
[x,y]=meshgrid(-5:0.1:5); z=x.^2+y.^2; ribbon(y,z,0.5);
title('z=x^2+y^2');xlabel('x');ylabel('y');zlabel('z');
```



Ejemplo 1.40

```
[x, y, z] = peaks(30); surf(x,y,z); hold on
[u,v,w]=surfnorm(x,y,z/15); h = quiver3(x,y,z,u,v,w);
set(h,'maxheadsize',0.50);
```

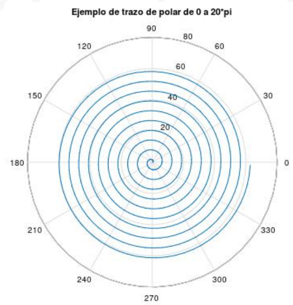


1.8 Gráficas en coordenadas polares

El trazado de datos en coordenadas polares se facilita con la función polar. Sin embargo, las coordenadas de visualización todavía son lineales y rectangulares.

Ejemplo 1.41

```
polar (0:0.1:20*pi, 0:0.1:20*pi);
title ('Ejemplo de trazo de polar de 0 a 20*pi')
```



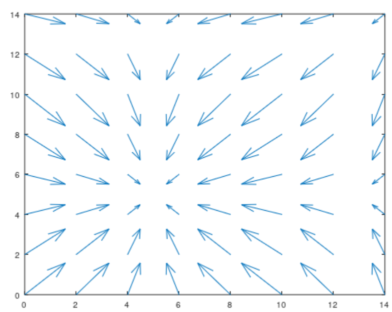
1.9 Gráfica de un campo vectorial 2D.
Los comandos quiver (x1, y1, u1, v1) y quiver(u1, v1), dibujan las componentes (u1, v1) de un campo de vectores en los puntos de la rejilla definidos por (x1, y1). Si la rejilla es uniforme, entonces x1 e y1 se pueden expresar como vectores y el comando meshgrid se usa para crear la rejilla 2-D.

Si x1 e y1 no se definen, se supone que son (1:m, 1:n) donde [m, n] = size(u1).

Ejemplo 1.42

[x1, y1] = meshgrid (0:2:15);

quiver(x1, y1, cos(2*pi*x1/20), cos(2*pi*y1/20))



Taller N° 01 gráficos en 2D y 3D

1. Representa gráficamente la función $y = 1 + x$.

2. Representa la función $y = 2x^2 + x^2 + 1$ y resuelve el valor de y si $x = 10$.

3. Dibuja la función $y = 2 + \sin(2x)$ de color rojo y con circulo.

4. Dibuja la función $y = \cos x$, con $z = sen\ x$, una de color amarillo y otra de color negro.

5. Crea cuatro subplot de las siguientes funciones: seno, coseno, tangente y la cotangente de x.

6. Dibuja en una misma grafica las funciones

$$y = x^2,\ \ z = x^3 + 2.$$

7. Use octave para hallar el coeficiente de Rayleigh y grafique a partir de la siguiente formula:

$$rs = a.fray.(\frac{1}{\frac{landa}{500}})^{-4}$$

$$a = 48; b = 0.7; fray = 0.41;$$

$$landa = 400 - 1200\ nm$$

8. Usar octave para hallar el índice de refracción a partir de la siguiente formula y compáralo con la longitud de onda propuesta:

$$n = \frac{A + B}{\lambda^2} + \frac{C}{\lambda^4}$$

$$A = 13696;\ \ B = 3916.8;\ \ C = 2558.8; landa = -1300$$

9. Representa en la misma gráfica:

$x = [1, 100]$, $y = x^2 + x^3 + 2x + 6$, $z = \sin x^2$,

$w = \frac{\sin x}{\cos x}$.

10. Represente la función:

$$y = \frac{x^2+x+1}{x+1}.$$

11. Dada la función:

$$x = \cos(at) - cos^3(bt)$$

$$y = \sin(ct) - sin^3(dt)$$

construya una función en el sistema cartesiano bidimensional que reciba los parámetros a, b, c, d y haga una representación gráfica de la función dada. Evalué la función en los siguientes valores dados: [1,80,1,80], [80,1,80,1], [80,1,1,80] y [1,100,1,50].

12. Las ecuaciones paramétricas de un hiperboloide de una hoja son:

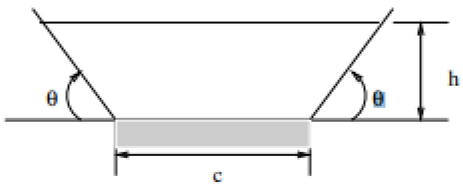
$$x = \mathrm{a}.\cosh(\theta)\cos(\phi)$$

$$y = b.cosh(\theta)\sin(\phi)$$

$$z = c.sinh(\theta)$$

Se pide representar la función con $\theta \in \mathbb{R}$, $0 < \phi \leq 2\pi.$

13. Se a diseñado un canal, como el que se muestra en la siguiente figura, se pretende que la velocidad de flujo sea máxima.



Las variables para considerar a la hora de diseñar el canal son su altura h, la anchura de la base c, y el Angulo lateral θ. Puede demostrarse que la velocidad de flujo es inversamente proporcional al perímetro, cuya expresión es:

$$p = c + \frac{2h}{\sin(\theta)}$$

Mientras que el área total de flujo es

$$A = ch + h^2\cot(\theta)$$

Considerando que el canal debe tener 200 cms. De área transversal (A=200), la función 1/p únicamente dependerá de h y θ. Represente la superficie $\frac{1}{p} = f(h,\theta)$, mostrando el diagrama de contorno simultáneamente.

Solución del taller N° 01 gráficos en 2D y 3D

```
#Ejercicio 1
ezplot(@(x) 1+x)

# Ejercicio 2
f = @(x) 2*x.^2 + x.^2 +1
ezplot(f)
fprintf("f evaluado en x = 10\n")
feval(f,10)

# Ejercicio 3
f = @(x) sin(x)
h = ezplot(f);
set (h,'color','g','marker','*','markersize',3)

# Ejercicio 4
f = @(x) sin(x)
g = @(x) cos(x)
h1 = ezplot(f);
hold on
grid on
h2 = ezplot(g);
set (h1,'color','g')
set (h2,'color','r')
title("");

# Ejercicio 5
subplot(2,2,1)
f = @(x) sin(x);
h1 = ezplot(f);
title("Funcion seno")
subplot(2,2,2)
f = @(x) cos(x);
h2 = ezplot(f);
title("Funcion coseno")
subplot(2,2,3)
f = @(x) tan(x);
h3 = ezplot(f);
title("Funcion tangente")
subplot(2,2,4)
f = @(x) cot(x); h4 = ezplot(f);
title("Funcion cotangente")

# Ejercicio 6
f = @(x) x.^2; g = @(x) x.^3 + 2;
h1 = ezplot(f);
```

```
hold on
grid on
h2 = ezplot(g);
set (h1,'color','g')
set (h2,'color','r')
title("");

# Ejercicio 7
a = 48; b = 0.7;
fray = 0.41;
landa = 400:1:1200;
f = a.*fray.*(1./(landa./500)).^(-4);
plot(landa,f);
fprintf('%10.1f %10.1f\n',landa,f)

# Ejercicio 8
A = 13696; B = 3916.8; C = 2558.8;
lambda = 300:1:1600;
n = (A+B)./lambda.^2 + C./lambda.^4;
plot(lambda,n);
fprintf('%10.5f %10.5f\n',lambda,n)

# Ejercicio 9
x = [1:100]; y = x.^2 + x.^3 + 2.*x + 6;
z = 5000.*sin(x.^2); w = 5000.*sin(x)./cos(x);
hold on
plot(x, y, x, z, x, w)
# Ejercicio 10
f = @(x) (x.^2 + x + 1)./(x + 1)
h = ezplot(f);

# Ejercicio 11
function z = grafica(a, b, c, d)
  t = linspace(0,10*pi,1000);
  x = cos(a.*t) - (cos(b.*t)).^3;
  y = sin(c.*t) - (sin(d.*t)).^3;
  plot(t,x,'g',t,y,'r')
endfunction

# Ejercicio 12
theta = [0:0.1:2*pi];
phi = [0:0.1:2*pi];
a = 1; b = 1; c = 1;
x = a.*cosh(theta).*cos(phi);
y = b.*cosh(theta).*sin(phi);
z = c.*sinh(theta);
plot3(x,y,z)
```

```
# Ejercicio 13
x = 2:1:30;
# altura desde 2 para poder ver las curvas de nivel.
y = 0:0.1:pi*150/180;
# no debe llegar a 180° porque "p" se hace infinito
# generamos puntos con x,y que podamos utilizar en las # ecuaciones para "p"
[h,theta] = meshgrid(x,y);
c = (200 - h.^2.*cot(theta))./h;
p = c + 2.*h./sin(theta); k = 1./p;
# graficamos la malla y las curvas de nivel
meshc(h, theta, k)
```

Objetivo:

Calcular límites, derivadas e integrales con los comandos apropiados que posee Octave.

CONTENIDO:

- ✓ Límite de una función.
- ✓ Límite de una función seccionada.
- ✓ Límite al infinito.
- ✓ Derivada de una función.
- ✓ Integrales indefinidas.
- ✓ Integrales definidas.

Capítulo

2