Matemáticas 1

MATLAB:

Comandos y ejemplos

Elena Álvarez Sáiz

Dpto. Matemática Aplicada y C. Computación

Universidad de Cantabria

Para obtener un número con los decimales indicados en dígitos:

vpa(número, dígitos)

Ejemplo:

>> vpa(pi,30)

Operadores elementales:

Operador	Utilización	Ejemplo
+	Adición	2+3=5
-	Sustracción	2-3=-1
*	Multiplicación	2*3=6
/	División	2/3=0.6667
^	Potenciación	2^3=8

Operadores entre arrays	Utilización	Ejemplo
.*	Multiplicación término a término	[2 3] .* [2 4] = = [4 12]
./	División término a término	[2 3] ./ [2 4] = = [1 0.7500]
.^	Potenciación término a término	[2 3] .^ 2 = [4 9]

Funciones elementales:

Funciones	Utilización	Ejemplo
exp(x)	Exponencial de x	exp(1)=2.7183
log(x)	Logaritmo natural	log(2.7183)=1.0000
log10	Logaritmo en base 10	log10(350)=2.5441
sin(x)	Seno de x	sin(pi/6)=0.500
cos(x)	Coseno de x	cos(0)=1
tan(x)	Tangente de x	tan(pi/4)=1.000

asin(x)	Arco coseno de x con imagen en el rango [0,]	asin(1)=1.5708

acos(x)	Arco coseno de x con imagen en el rango [- ,]	acos(1)=-6.1257e-17
atan(x)	Arco tangente de x con imagen en el rango [- ,]	atan(1)=0.7854
atan2(y,x)	Arco tangente de y/x con imagen en el rango [- ,]	atan2(0,-1)=3.1416
sinh(x)	Seno hiperbólico de x	sinh(3)=10.0179
cosh(x)	Coseno hiperbólico de x	cosh(3)=10.0677
tanh(x)	Tangente hiperbólica de x	tanh(3)=0.9951

Para representar vectores:

plot(x,y)

dibuja un vector de abscisas "x" y ordenadas "y"

plot(y)

dibuja el vector "y" considerado como abscisas su índice. Si "y" es complejo es equivalente a dibujar plot(real(y),imag(y)).

plot(x,y,s)

Realiza el gráfico con el estilo indicado en "s". Para ello "s" debe ser una cadena de caracteres formada por uno o ningún elemento de las tres columnas siguientes:

y	yellow	.	point	-	solid
m	magenta	o	circle	:	dotted
c	cyan	x	x-mark	-.	dashdot
r	red	+	plus	--	dashed

g	green	*	star
b	blue	s	square
w	white	d	diamond
k	black	v	triangle (down)
		^	triangle (up)
		<	triangle (left)
		>	triangle (right)
		p	pentagram
		h	hexagram

Ejemplo:

```
n=1:10
a=2.^n;
plot(a,'bo')
%Para ver más opciones teclea la orden:
help plot
```

Para crear una ventana de dibujo:

figure(n)

Ejemplo:

```
>>  x=-pi : 0.1: pi;
>>  figure(1);
>>  plot(x,sin(x),'b. ');
>>  figure(2);
>>  plot(x,cos(x), 'gd-');
```

hold on

hold off

Permite dibujar dos gráficas en una misma ventana de dibujo.

Ejemplo:

```
>>      x=-pi : 0.1: pi;
>>      hold on
>>      figure(1);
>>      plot(x,sin(x),'b. ');
>>      plot(x,cos(x), 'gd-');
>>      hold off
```

Para manejar números complejos:

i

Es la unidad imaginaria en Matlab

abs(s)

Valor absoluto de los elementos de "s" o módulo en el caso de ser complejos.

Ejemplo:

```
>>      z=2+3i; w=5+7i;
>>      abs(z)          %       Devuelve
3.6056
>>      abs([z,w])      %       Devuelve
3.6056   86023
```

angle(h)

Retorno el ángulo de fase en radianes de cada elemento de la matriz h con elementos complejos.

Ejemplo:

```
>>      z=2+3i; w=5+7i;
```

```
>>  angle(z) % Devuelve  0.9828
>>  angle([z,w])  %  Devuelve
0.9828  0.9505
```

real(z)

Devuelve la parte real de z

Ejemplo:

```
>>  z=2+3i; w=5+7i;
>>  real(z)  % Devuelve  2
>>  real([z,w])  % Devuelve  2
5
```

imag(z)

Devuelve la parte imaginaria de z

Ejemplo:

```
>>  z=2+3i; w=5+7i;
>>  imag(z) % Devuelve  3
>>  imag([z,w])  % Devuelve  3
7
```

conj(z)

Devuelve el conjugado de z

Ejemplo:

```
>>  z=2+3i; w=5+7i
```

```
>> conj(z)
% Devuelve 2.0000-3.0000i
>>      conj([z,w])
% Devuelve      2.0000-3.0000i   5.000-7.000i
```

Para representar números complejos

plot(z)

Si z es un número complejo el comando plot dibuja el punto de coordenadas (real(z),imag(z)).

compass(z)

Representa el número complejo como una flecha que tiene su origen en el punto (0,0).

Ejemplo:

```
>>      z=3+2*i;
>>      figure(1);
>>      plot(z);
>>      figure(2);
>>      compass(z);
>>      % Esto es equivalente a:
>>      compass(real(z),imag(z));
```

Para manejar polinomios:

Los polinomios se representan en Matlab como un array de coeficientes. Por ejemplo, el polinomio:

$$p(x) = x^2 + 2x + 1$$

se escribirá en Matlab

```
>> p=[1 2 1];
```

roots(polinomio);

Calcula las raíces del polinomio. Es decir resuelve $x^2 + 2x + 1 = 0$.

Ejemplo:

```
>> r=roots(p);
```

poly(array)

Calcula los coeficientes del polinomio que tenga las raíces que se indiquen en array.

Ejemplo:

```
>> raices=[1 1 1];
>> p=poly(raices);
```

polyval(polinomio,array)

Evalúa el polinomio en cada uno de los puntos del array.

Ejemplo:

```
>> x=-3 : 0.1: 5;
>> y=polyval(p,x);
% Dibuja la gráfica de la función polinómica
% en el dominio indicado
>> plot(x,y)
```

conv(polinomio1,polinomio2)

Realiza el producto de los polinomios operando entre los arrays de los coeficientes.

Ejemplo:

```
>> p1=[ 2 3 1];
>> p2= [5 –2];
>> p3=conv(p1,p2);
```

deconv(polinomio1, polinomio2)

Realiza el cociente entre el primer y el segundo polinomio.

Ejemplo:

```
>> [cociente, resto] =deconv(p3,p1)
```

Para construir objetos simbólicos:

syms arg1 arg2 ...

Es la forma abreviada de escribir:

```
arg1 = sym('arg1');
arg2 = sym('arg2'); ...
```

Si se quiere indicar el tipo del objeto simbólico se puede escribir:

syms arg1 arg2 ... real

Es la forma abreviada de escribir:

```
arg1 = sym('arg1','real');
arg2 = sym('arg2','real'); ...
```

syms arg1 arg2 ... positive

Es la forma abreviada de escribir:

arg1 = sym('arg1','positive');

arg2 = sym('arg2','positive'); ...

syms arg1 arg2 ... unreal

Es la forma abreviada de escribir:

arg1 = sym('arg1','unreal');

arg2 = sym('arg2','unreal'); ...

Ejemplo:

```
>>  syms x
>>  y=sin(x)+3^x+8/(x+1)
```

Para hacer una sustitución simbólica simple de "var" en "valor" en la expresión "f":

subs(f,var,valor)

Ejemplo:

```
>>  syms x
>>  y=sin(x)+3^x+8/(x+1)
>>  subs(y, x, 2)
```

Para realizar la gráfica de una función simbólica en un dominio y en la ventana de dibujo indicada en fig:

ezplot(f, [a,b], fig)

Ejemplo:

```
>>  syms x
>>  y=sin(x)+3^x+8/(x+1)
>>%El segundo y el tercer parámetro son opcionales.
>>  ezplot(y, [-2,2])
```

Para resolver de forma simbólica ecuaciones algebraicas:

solve('eqn1','eqn2',...,'eqnn')

solve('eqn1','eqn2',...,'eqnn','var1,var2,...,varn')

solve('eqn1','eqn2',...,'eqnn','var1','var2',...'varn')

Ejemplo:

```
>>  % Calculamos las raíces de un polinomio genérico de grado 3.
>>  syms x a b c d
>>  v=solve(a*x^3+b*x^2+c*x+d)
>>  r=subexpr(v(1))
>>  s=subexpr(v(2))
>>  t=subexpr(v(3))
```

Para escribir simplificada o de forma más habitual una expresión:

pretty(expresion)

Ejemplo:

```
>>  syms x
>>  pretty(sin(x)^2+(cos(x)+3)/(sin(2*x)+5))
```

simplify(expresion)

Ejemplo:

```
>>  syms x
```

```
>> pretty(simplify(cos(x)*cos(x)-sin(x)*sin(x)))
```

Para construir objetos simbólicos:

syms arg1 arg2 ...

Es la forma abreviada de escribir:

arg1 = sym('arg1');
arg2 = sym('arg2'); ...

Si se quiere indicar el tipo del objeto simbólico se puede escribir:

syms arg1 arg2 ... real

Es la forma abreviada de escribir:

arg1 = sym('arg1','real');
arg2 = sym('arg2','real'); ...

syms arg1 arg2 ... positive

Es la forma abreviada de escribir:

arg1 = sym('arg1','positive');
arg2 = sym('arg2','positive'); ...

syms arg1 arg2 ... unreal

Es la forma abreviada de escribir:

arg1 = sym('arg1','unreal');
arg2 = sym('arg2','unreal'); ...

Ejemplo:

```
>>  syms x
>>  y=sin(x)+3^x+8/(x+1)
```

Para hacer una sustitución simbólica simple de "var" en "valor" en la expresión "f":

subs(f,var,valor)

Ejemplo:

```
>>  syms x
```

```
>>      y=sin(x)+3^x+8/(x+1)

>>      subs(y, x, 2)
```

Para realizar la gráfica de una función simbólica en un dominio y en la ventana de dibujo indicada en fig:

ezplot(f, [a,b], fig)

Ejemplo:

```
>>      syms x

>>      y=sin(x)+3^x+8/(x+1)

>>  % El segundo y el tercer parámetro son opcionales.

>>      ezplot(y, [-2,2])
```

Para resolver de forma simbólica ecuaciones algebraicas:

solve('eqn1','eqn2',...,'eqnn')

solve('eqn1','eqn2',...,'eqnn','var1,var2,...,varn')

solve('eqn1','eqn2',...,'eqnn','var1','var2',...'varn')

Ejemplo:

```
>>      % Calculamos las raíces de un polinomio genérico de grado 3.

>>      syms x a b c d

>>      v=solve(a*x^3+b*x^2+c*x+d)

>>      r=subexpr(v(1))

>>      s=subexpr(v(2))
```

```
>>      t=subexpr(v(3))
```

Para obtener el límite de una expresión simbólica "f" cuando la variable "n" tiende al valor "a"

limit(f,n,a)

Ejemplo:

```
>> syms n
>> limit(1/n,n,inf)
```

Para obtener la derivada de orden n una función simbólica respecto de la variable x.

diff(f,x,n)

Ejemplo:

```
>> syms x y
>> f=sin(x*y)/x; diff(f,x,3)
```

Las funciones que simplifican la forma de las expresiones simbólicas son:

collect (p)	Reúne los términos iguales
horner(p)	Cambia a la representación anidada o de Horner
expand(p)	Expande los productos en sumas
factor(p)	Factoriza la expresión (a veces) si el argumento es una función simbólica. Si se trata de un número proporciona la factorización en números primos.
simplify(p)	simplifica una expresión mediante la aplicación de diversas identidades algebraicas.

simple(p) Utiliza diferentes herramientas de simplificación y selecciona la forma que tiene el menor número de caracteres

pretty(p) Visualiza la expresión de una manera similar a la utilizada en la escritura habitual.

Para calcular la suma entre los valores a y b de la variable

symsum(f,a,b)

symsum(f,s,a,b)

Ejemplo:

```
>> syms n
>> symsum(1/n,1,inf)
```

Para descomponer un polinomio en fracciones simples

[R,P,K] = residue(B,A)

Encuentra la descomposición en fracciones simples de dos polinomios B(s)/A(s). Los vectores B y A contendrán los coeficientes del numerador y del denominador en potencias descendentes de s.

Si no hay raíces múltiples,

$$\frac{B(s)}{A(s)} = \frac{R(1)}{s - P(1)} + \frac{R(2)}{s - P(2)} + \ldots + \frac{R(n)}{s - P(n)} + K(s)$$

Si P(j) = ... = P(j+m-1) es un cero of multiplicidad m, entonces aparecen términos de la forma

$$R(j) \qquad R(j+1) \qquad R(j+m-1)$$

```
-------- + ------------   + ... + ------------
 s - P(j)   (s - P(j))^2          (s - P(j))^m
```

Ejemplo:

```
>> [R,P,K]=residue([1],[1 1 0])
     R =
        -1
         1
     P =
        -1
         0
     K =
         []
```

Para obtener el límite de una expresión simbólica "f" cuando la variable "n" tiende al valor "a"

limit(f,n,a)

Ejemplo:

```
>> syms n
>> limit(1/n,n,inf)
```

Para obtener la derivada de orden n una función simbólica respecto de la variable x.

diff(f,x,n)

Ejemplo:

```
>> syms x y
>> f=sin(x*y)/x; diff(f,x,3)
```

Para integrar una función simbólica

int(función,variable,LímiteInferior, LímiteSuperior)

Ejemplo:

```
>> syms x
>> int(1/x,x,1,4)
```

Funciones básicas elementales.

sin	Seno
sinh	Seno hiperbólico
asin	Arco seno
asinh	Arco seno hiperbólico
cos	Coseno
cosh	Coseno hiperbólico
acos	Arco coseno
acosh	Arco coseno hiperbólico
tan	Tangente
tanh	Tangente hiperbólica
atan	Arco tangente
atan2	Arco tangente en cuatro cuadrantes
atanh	Arco tangente hiperbólica
sec	Secante
sech	Secante hiperbólica
asec	Arco secante
asech	Arco secante hiperbólica
csc	Cosecante
csch	Cosecante hiperbólica
acsc	Arco cosecante
acsch	Arco cosecante hiperbólica
cot	Cotangente
coth	Cotangente hiperbólica
acot	Arco cotangente
acoth	Arco cotangente hiperbólica
exp	Exponencial
log	Logaritmo natural
log10	Logaritmo decimal
pow2	Potencia en base 2
sqrt	Raíz cuadrda

fix	Redondeo hacia cero
floor	Redondeo hacia menos infinito
ceil	Redondeo hacia más infinito
round	Redondeo hacia el entero más próximo
mod	Módulo (cociente entero de la división)
rem	Resto entero de la división
sign	Función signo

Para calcular el límite de una función simbólica de variable x cuando se tiende al valor a.

limit(función,x,a)

Ejemplo.-

```
>>syms x
>>f=sin(x)/x
>>limit(f,x,0)
```

Para calcular el límite de una función simbólica de variable x cuando se tiende al valor a por la derecha o por la izquierda.

limit(función,x,a,'right')
limit(función,x,a,'left')

Para obtener la derivada de orden n una función simbólica respecto de la variable x.

diff(f,x,n)

Ejemplo.-

```
>> syms x y
>> f=sin(x*y)/x
>>  diff(f,x,3)
```

Para calcular el polinomio de Taylor de orden n-1 de la función f en el punto "a"

taylor(f,a,n)

Ejemplo:

```
>> syms x
>> y=sin(x)+3^x+8/(x+1)
>> taylor(y,2,4)
```

Para realizar la gráfica de una función simbólica en un dominio y en la ventana de dibujo indicada en fig:

ezplot(f, [a,b], fig)

Ejemplo:

```
>>syms x
>>y=sin(x)+3^x+8/(x+1)
>>% El segundo y el tercer parámetro son opcionales.
>>ezplot(y, [-2,2])
```

Para representar un polinomio se considera un vector fila conteniendo todos los coeficientes en orden decreciente, incluyendo ceros.

Ejemplo:

```
>>P=[1 0 –1 3 4]
>>% Se trata del polinomio x^4 – x^2 + 3x + 4
```

Para manipular polinomios se tienen las siguientes funciones:

roots	Calcula las raíces de un polinomio
poly	Construye un polinomio con unas raíces específicas
polival	Evalúa un polinomio
residue	Desarrolla en fracciones simples
polyfit	Ajusta un polinomio a unos datos
polider	Derivada de un polinomio
conv	Multiplicación de polinomios
deconv	División de polinomios

Para representar un polinomio

Ejemplo:

```
>>x=linspace(-1,5)
>>vy=polyval(y,x)
>>plot(x,vy)
```

Para calcular la derivada de un polinomio definido como el vector de sus coeficientes

polyder(polinomio)

Ejemplo:

```
>> p=[2 3 4 –1]
>> polyder(p)
```

Función que determina si una expresión es infinito

isinf(Vector)

Devuelve uno donde el elemento de Vector es +Inf o –Inf y 0 donde no lo sea.

Ejemplo:

```
>> isinf([pi NaN Inf -Inf])
```

Para generar una malla de puntos en los que evaluar una función de dos variables.

```
meshgrid(x,y)
meshgrid(x)                    %Es equivalente a meshgrid(x,x)
```

Ejemplo.-

```
%Para evaluar la función f(x,y)=x^2*y en el dominio –2<x<2,
% -3<y<3
>>[X, Y]=meshgrid(-2:.2:2,-3:0.5:3);
>>Z=X.^2.* Y
```

Gráficos tridimensionales.

plot3(X,Y,Z,S)

Dibuja el conjunto de puntos (X,Y,Z) donde X, Y y Z son vectores fila y S son las opciones de dibujo.

plot3(X1,Y1,Z1,S1,X2,Y2,Z2,S2,...)

Dibuja sobre los mismos ejes los gráficos definidos por las tripletas (Xi,Yi,Zi) con las opciones de dibujo por Si.

Ejemplo.-

```
%Para evaluar la función f(x,y)=x^2*y en el dominio –2<x<2,
%  -3<y<3
>>[X, Y]=meshgrid(-2:.2:2,-3:0.5:3)
>>Z=X.^2.*Y
>>plot3(X,Y,Z)
```

Gráficos de superficie.

surf(X,Y,Z,C)

Representa el gráfico de superficie de la función z=f(x,y) con los colores especificados en C (este último parámetro se puede ignorar).

surfc(X,Y,Z,C)

Representa el gráfico de superficie de la función z=f(x,y) junto con el gráfico de contorno correspondiente (curvas de nivel)

Ejemplo.-

```
>>%Para evaluar la función f(x,y)=x^2*y en el dominio –2<x<2,
>> % -3<y<3
>>[X, Y]=meshgrid(-2:.2:2,-3:0.5:3);
>>Z=X.^2.*Y;
>>figure(1)
>>surf(X,Y,Z)
>>figure(2)
>>surfc(X,Y,Z)
```

Gráficos de malla.

mesh(X,Y,Z,C)

Representa el gráfico de malla de la función z=f(x,y) con los colores especificados en C (este último parámetro se puede ignorar).

meshc(X,Y,Z,C)

Representa el gráfico de malla de la función z=f(x,y) junto con el gráfico de contorno correspondiente (curvas de nivel)

meshz(X,Y,Z,C)

Representa el gráfico de malla de la función z=f(x,y) junto con una especie de cortina en la parte inferior.

Ejemplo.-

```
>>%Para evaluar la función f(x,y)=x^2*y en el dominio –2<x<2
>>% -3<y<3
>>[X, Y]=meshgrid(-2:.2:2,-3:0.5:3);
```

```
>>Z=X.^2.*Y;
>>figure(1)
>>mesh(X,Y,Z)
>>figure(2)
>>meshc(X,Y,Z)
>>figure(3)
>>meshz(X,Y,Z)
```

Gráficos de contorno (curvas de nivel).

contour(Z,n)

Representa el gráfico de contorno para la matriz Z usando n líneas. El segundo parámetro es opcional.

contour3(Z,n)

Representa el gráfico de contorno en tres dimensiones para la matriz Z usando n líneas. El segundo parámetro es opcional.

Ejemplo.-

```
>>%Para evaluar la función f(x,y)=x^2+y^2 en el dominio
>>  −2<x<2, -3<y<3
>>[X, Y]=meshgrid(-2:.2:2,-3:0.2:3);
>>Z=X.^2.+Y.^2;
>>figure(1)
>>contour(Z)
>>figure(2)
>>contour3(Z)
```

Gráficos de densidad

pcolor(X,Y,Z)

Representa el gráfico de contorno para la matriz (X,Y,Z) utilizando densidades de colores.

Ejemplo.-

```
>>%Para evaluar la función f(x,y)=x^2+y^2 en el dominio
>>%-2<x<2, -3<y<3
>>[X, Y]=meshgrid(-2:.2:2,-3:0.2:3);
>>Z=X.^2.+Y.^2;
>>pcolor(X,Y,Z)
```

Representación

view([x,y,z])

Sitúa el punto de vista de la figura en el indicado por las coordenadas (x,y,z).

ginput

Nos devuelve las coordenadas (x, y) del punto una vez seleccionado en la gráfica.

Para calcular los límites iterados de una función de dos variables en el punto (a, b)

Ejemplo.-

```
>>syms x y
>>f=x^2+y^2
>>limit(limit(f,x,a),y,b)
>>limit(limit(f,y,b),x,a)
```

Para calcular el límite de una función según una dirección x=g(t), y=h(t) cuando t tiende a cero.

Ejemplo.-

```
>>syms x y t
>>f=x^2+y^2;
```

```
>>nf=subs(f,{x,y},{g(t),h(t)})
>>limit(nf,t,0)
```

Para calcular el límite de una función en coordenadas polares cuando nos aproximamos al punto (a, b)

Ejemplo.-

```
>>syms x y
>>f=x^2+y^2;
>>syms r theta
>>polar=subs(f,{x,y},{a+r*cos(theta),b+r*sin(theta)})
>>limit(polar,r,0)
```

NÚMEROS COMPLEJOS: EJEMPLOS Y PROGRAMAS

1 Operaciones con números complejos

```
x=-4:0.1:4;
y=-4*ones(1,length(x));
z=x+i*y;
%%A=((5.9997)*(z +(0.1667-0.1667i))) ./ (z +(6.9997+1.9999i))
A=1./z;
%%A=((0.4667+2.0667i)*(z +(0.3465+0.4654i))) ./ (z +(1.4+2.2i));
a1=real(A);
b1=imag(A);
hold on
plot(a1,b1)
%%
x=-4*ones(1,length(x));
y=-4:0.1:4;
z=x+i*y;
A=1./z;
a1=real(A);
b1=imag(A);
plot(a1,b1)
%%
x=-4:0.1:4;
y=4*ones(1,length(x));
z=x+i*y;
A=1./z;
a1=real(A);
b1=imag(A);
plot(a1,b1)
%%
x=4*ones(1,length(x));
y=-4:0.1:4;
z=x+i*y;
A=1./z;
a1=real(A);
b1=imag(A);
plot(a1,b1)
%
%%
t=-2*pi:0.1:2*pi;
x=3*cos(t);
y=3*sin(t);
z=x+i*y;
A=1./z;
a1=real(A);
b1=imag(A);
plot(a1,b1)
%%
t=-2*pi:0.1:2*pi;
x=1+(1/3)*cos(t);
```

```
y=1.5+(1/3)*sin(t);
z=x+i*y;
A=1./z;
a1=real(A);
b1=imag(A);
plot(a1,b1)
%%
%%
t=-2*pi:0.1:2*pi;
x=-1+(1/3)*cos(t);
y=1.5+(1/3)*sin(t);
z=x+i*y;
A=1./z;
a1=real(A);
b1=imag(A);
plot(a1,b1)
%%
t=-pi:0.1:0;
x=2*cos(t);
y=2*sin(t);
z=x+i*y;
A=1./z;
a1=real(A);
b1=imag(A);
plot(a1,b1)
```

2 Potencias de números complejos: Dado z=i

(a) Calcula $w = z^n$ para n=1,...,10

(b) Representa dichos valores en el plano complejo.

(c) ¿Qué ocurre con w cuando n crece indefinidamente?.

(d) Repite los apartados anteriores con z=0.99 i y z=1.1i

```
%Dado z=i
%   (a) calcular z^n para n=1,2...10
for n=1:10
   v(n)=i^n;
end
%    (b) Representar estos números complejos
plot(v)
%     ¿Qué ocurre cuando n tiende a infinito?
%    (c) Repetir los resultados cuando z=0.99 i y z=1.1 i
for n=1:10
   v(n)=(1.1*i)^n;
end
plot(v)
```

SUCESIONES Y SERIES: EJEMPLOS Y PROGRAMAS

1 Convergencia de una sucesión

```
clear;
%Definimos el número de términos de la sucesión
%y los extremos inferior y superior donde queremos
%representar la sucesión
nInferior=input('Da el valor de n a partir del cual quieres dibujar
la sucesión:    ');
terminos=input('Da el número de términos a dibujar:    ');
nSuperior=nInferior+terminos-1;
x=linspace(nInferior,nSuperior,terminos);
syms n
suc=input('Da la sucesión en función de n:    ');
for k=1:terminos
   an(k)=double(subs(suc,n,nInferior+k-1));
end
%Dibujamos la sucesión en verde y con puntos
plot(an,'g.')
%Damos el valor de epsilon
epsilon=input('Da un valor para epsilon:    ');
%Dibujamos la recta y=limite+epsilon, y=limite-epsilon
limite=limit(suc,n,inf)
y1=ones(terminos)*double(limite+epsilon);
y2=ones(terminos)*double(limite-epsilon);
hold on
plot(y1,'b')
plot(y2,'b')
```

2 Representación de las sumas parciales enésimas de una serie y cálculo de su límite.

```
%Fichero "SumaParcial.m"
clear;
%Definimos el número de términos de la sucesión
%y los extremos inferior y superior donde queremos
%representar la sucesión
disp(' ');
syms n
suc=input('Da el término general de la serie en función de n:    ');
nInferior=input('Da el valor de n a partir del cual quieres dibujar
la sucesión de sumas parciales:    ');
```

```
terminos=input('Da el número de términos a dibujar:     ');
nSuperior=nInferior+terminos-1;
x=linspace(nInferior,nSuperior,terminos);
valorAnterior=nInferior-1;
suma=double(symsum(suc,1,nInferior-1));
disp('         ');
disp('El valor de las sumas parciales es:');
disp('-----------------------------------');
for k=1:terminos
   an(k)=double(subs(suc,n,nInferior+k-1));
   suma=suma+an(k);
   sumParcial(k)=suma;
   disp(['Sn para n = ' num2str(nInferior-1+k) ' ----> '
num2str(sumParcial(k))])
end
disp('     ');
%Dibujamos la sucesión
disp('Da un número para indicar en que ventana de dibujo ');
valor=input('quieres pintar la sucesión y la suma parcial n-ésima:
');
figure(valor)
hold on
plot(an,'g.')
text(2.5,an(2),'Sucesión');
%Dibujamos la suma parcial en rojo y con *
plot(sumParcial,'r*')
text(2.5,sumParcial(2),'Suma parcial');
%El valor de la suma parcial n-esima es:
disp('         ');
disp('El valor de la suma parcial n-ésima es:    ');
disp('---------------------------------------');
syms m
pretty(subs(symsum(suc,1,m),m,n))
%La suma de la serie es:
disp('         ');
disp('La suma de la serie es:    ');
disp('-----------------------');
pretty(symsum(suc,1,inf))
```

3 Clasificación de una serie numérica

```
%clasificacion.m
disp(' ');
disp('""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
"""""""')
disp('""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
"""""""')
syms n positive
suc=input('Da el término general "an" de la serie en función de n:
');
if abs(suc) == suc
    disp('-------------------------------------------------------------
------------')
   disp('Se trata de una serie de términos positivos.')
```

```
   disp('Por lo tanto es una serie o bien convergente o bien
divergente.')
   disp('    ')
    disp('-------------------------------------------------------------
------------')
   disp('El limite del termino general an es')
   limite=limit(suc,n,inf,'left');
   pretty(limite)
   if limite == 0
       disp('    ')
    disp('-------------------------------------------------------------
------------')
      disp('La condición necesaria de convergencia no da
información.')
      disp('  ');

disp('.....................................................................
........')
      disp('   ');
      disp('Utiliza el fichero "criterios.m" para determinar si es
posible su caracter.')
   else
      disp('    ')
    disp('-------------------------------------------------------------
------------')
      disp('La condición necesaria de convergencia indica que la
serie es divergente')
   end
else
   if abs(suc) == (-1)*suc
    disp('-------------------------------------------------------------
------------')
      disp('Se trata de una serie de términos negativos.')
      disp('Por lo tanto es una serie o bien convergente o bien
divergente.')
      disp('    ')
    disp('-------------------------------------------------------------
------------')
      disp('El limite del termino general es')
      limite=limit(suc,n,inf,'left');
      pretty(limite)
      if limite == 0
          disp('    ')
    disp('-------------------------------------------------------------
------------')
        disp('La condición necesaria de convergencia no da
información.')
         disp('  ');

disp('.....................................................................
........')
        disp('   ');
        disp('Utiliza el fichero "criterios.m" para determinar si es
posible su caracter.')
      else
          disp('    ')
    disp('-------------------------------------------------------------
------------')
```

```
        disp('La condición necesaria de convergencia indica que la
serie es divergente')
      end
   else
      disp('-------------------------------------------------------
------')
      disp('La serie es alternada o de términos cualesquiera');
      disp(' ');
      disp('-------------------------------------------------------
------')
        disp('El limite del termino general es')
      limite=limit(suc,n,inf,'left');
      pretty(limite)
      if limite == 0
          disp('    ')
   disp('-------------------------------------------------------------
---')

         disp('La condición necesaria de convergencia no da
información.')
         disp('  ');

disp('.....................................................................
........')
        disp('   ');
         disp('Estudia la convergencia absoluta o si se trata de ')
         disp('una serie alternada la convergencia por el criterio de
Leibniz')
      else
          disp('    ')
   disp('-------------------------------------------------------------
---')
disp('La condición necesaria de convergencia indica que la serie no
es convergente')
      end
   end
end
disp('  ')
disp('  ')
```

4 Criterios de convergencia de una serie

```
clear;
%Definimos el número de términos de la sucesión
%y los extremos inferior y superior donde queremos
%representar la sucesión
disp(' ');
syms n m positive
suc=input('Da el término general "an" de la serie en función de n:
');
    disp('        ');
    disp('Escribe 1 si deseas utilizar el criterio del cociente:');
    disp('Escribe 2 si deseas utilizar el criterio de la raiz:');
    disp('Escribe 3 si deseas utilizar el criterio de comparación:');
    disp('Escribe 4 si deseas utilizar el criterio integral:');
    disp('  ');
   valor=input('.......   ');
```

```
disp(' ')
 if valor==1
    disp('El cociente a(n)/a(n-1) es ')
 an1=subs(suc,n,n-1);
     cociente=(suc/an1);
   pretty(simple(cociente))
   disp(' ');
 disp('El limite de a(n)/a(n-1) es ')
 limite=limit(cociente,n,inf,'left');
   pretty(limite)
   disp(' ')
 if double(limite)<1
     disp('La serie es convergente')
     else
     if double(limite)>1
         disp('La serie es divergente')
     else
        disp('Como el límite es 1 utilizo el criterio de Raabe')
        disp('y calculamos lim(n*(1-a(n)/a(n-1))) cuyo valor es');
        limiteRaabe=limit(n*(1-cociente),n,inf,'left');
      pretty(simple(limiteRaabe))
      disp('  ');
         if double(limiteRaabe)>1
            disp('La serie es convergente')
         else
           if double(limiteRaabe)<1
              disp('La serie es divergente')
          else
             disp('Duda por el caso de Raabe')
          end
         end
     end
     end
 else
 if (valor == 2)
     disp('La raiz n-ésima de a(n) ')
         raiz=suc^(1/n);
     pretty(simple(raiz))
     disp('El limite de la raiz n-ésima de a(n) es ')
     limite=limit(raiz,n,inf,'left');
     pretty(limite)
     if double(limite)<1
         disp('La serie es convergente')
         else
             if double(limite)>1
                 disp('La serie es divergente')
             else
           disp('Utiliza otro criterio')
         end
         end
  else
      if (valor == 3)
             disp('  ')
             disp('Escribe:')
           disp('   1 para comparar con la serie armónica
generalizada.')
            disp('   2 para comparar con cualquier otra serie.')
            opcion=input('.......   ');
            disp('  ')
            if opcion == 1
```

```
                    disp('    Introduce el valor de p para comparar')
                    valorp=input('    con la serie de término general
1/n^p:    ');
                    armonica=1/(n^valorp);
                     limite=limit(suc/armonica,n,inf,'left');
                     disp(' ')
                        disp('El límite (n^p)*an es ')
                        pretty(limite)
                        disp(' ')
                        if and(limite~=0,isinf(double(limite))== 0)
                            disp('Como el límite es distinto de cero
y de infinito ...')
                            if valorp>1
                            disp('La serie es convergente ');
                            else
                                 disp('La serie es divergente ')
                            end
                        end
                 else
                    disp('    Introduce el término general de la
sucesión ')
                    bn=input('    con la que deseas comparar en
función de n:    ');
                    limite=limit(suc/bn,n,inf,'left');
                    disp(' ')
                        disp('El límite an/bn es ')
                     pretty(limite)
                     disp('  ')
                        if and(limite~=0,isinf(double(limite))== 0)
                            disp('Como el límite es distinto de cero
y de infinito ...')
                        disp('  ')
                        disp('Las dos series tiene el mismo
carácter')
                        end

                    end
          else
             if (valor == 4)
                disp('  ')
                disp('Escribe:')
                disp('   1 si la sucesión an es monótona
decreciente.')
                disp('   2 si la sucesión an es monótona creciente.')
                opcion=input('.......   ');
                disp(' ')
                if opcion == 1
                   disp('Una sucesión equivalente a la suma parcial n-
ésima es:')
                   equivalente=subs(int(suc,1,m),m,n) ;
                   pretty(simple(equivalente))
                   disp(' ')
                   disp('y su limite cuando n tiende a infinito es  ')
                   aa=int(suc,1,inf);
                   pretty(simple(aa))
                   if isinf(double(aa)) == 0
                      disp('Es una serie convergente.')
                   else
                      disp('Es una serie divergente.')
```

```
                end
            else
                disp('La sucesión que acota inferiormente a la suma
parcial n-ésima es')
                inferior=subs(int(suc,1,m),m,n);
                pretty(inferior)
                disp('y su limite es ')
                limite=limit(inferior,n,inf,'left');
                pretty(limite)
                 disp('La sucesión que acota superiormente a la
suma parcial n-ésima es')
                superior=subs(int(suc,1,m)+suc,m,n);
                pretty(superior);
                disp('y su limite es ')
                limite=limit(superior,n,inf,'left');
                pretty(limite)

            end
        else
        end
    end
    end
end
disp(' ')
disp('El valor que da Matlab para la suma de la serie es: ')
 vpa(limit(symsum(suc,1,m),m,inf,'left'))
```

5 Criterio integral

```
clear;
%Definimos el número de términos de la sucesión
%y los extremos inferior y superior donde queremos
%representar la sucesión
disp(' ');
syms n
suc=input('Da el término general de la serie en función de n:    ');
terminos=input('Da el número de términos a considerar:    ');
disp('        ');
disp('El valor de las sumas parciales es:');
disp('-----------------------------------');
an(1)=double(subs(suc,n,1));
sumParcial(1)=an(1);
disp(['Sn para n = 1 '  ' ----> ' num2str(sumParcial(1))])
figure(1)
hold on
plot([0 1 1 0],[0 0 an(1) an(1)],'r')
for k=2:terminos
   an(k)=double(subs(suc,n,k));
   sumParcial(k)=sumParcial(k-1)+an(k);
   disp(['Sn para n = ' num2str(k) ' ----> '
num2str(sumParcial(k))]);
   plot([k-1 k-1  k k],[0 an(k)  an(k) 0],'r')
end
plot([terminos 0],[0 0],'r');
%Dibujamos la sucesión
plot(an,'go')
```

```
for k=1:terminos
   text(k+0.1,an(k),strcat('an(',num2str(k),')'))
end
ezplot(suc,[0,terminos])
title('Cota superior de la suma parcial n-ésima')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure(2)
hold on
for k=2:terminos
    plot([k-1 k-1 k k],[0 an(k-1) an(k-1) 0],'r')
 end
 plot([terminos 0],[0 0],'r');
%Dibujamos la sucesión
plot(an,'g.')
ezplot(suc,[0,terminos])
for k=1:terminos
   text(k-0.5,an(k),strcat('an(',num2str(k),')'))
end
title('Cota inferior de la suma parcial n-ésima')
%El valor de la suma parcial n-esima es:
disp('        ');
disp('El valor de la suma parcial n-ésima es:   ');
disp('-----------------------------------');
pretty(symsum(suc))
%Una sucesión que acota a la suma parcial n-ésima es:
disp('        ');
disp('Una sucesión que acota inferiormente a la suma parcial n-ésima
es:   ');
syms m
disp('-------------------------------------------------------');
pretty(subs(int(suc,1,m),n,m))
disp('Una sucesión que acota superiormente a la suma parcial n-ésima
es:   ');
syms m
disp('-------------------------------------------------------');
pretty(subs(int(suc,1,m)+an(1),n,m))
%La suma de la serie es:
disp('        ');
disp('La suma de la serie es:   ');
disp('----------------------');
pretty(symsum(suc,1,Inf))
```

6 Número de términos que es necesario considerar para calcular la suma de una serie con un error dado.

```
%Fichero "SumaAproximada2.m"
clear
syms x m
disp(' ');
disp('-----------------------------------------------------------------
----------');
disp('-----------------------------------------------------------------
----------');
syms n m
suc=input('Da el término general "an" de la serie en función de n:
');
```

```
disp(' ');
error=input('Da el error con el que quieres obtener la suma de la
serie:    ');
cotaError=int(suc,n,m,inf);
disp(' ');
disp('""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
""""""""""')
disp('Una cota superior del error es: ' )
pretty(subs(cotaError,m,n))
disp('-------------------------------------------------------------------
----------');
j=1;
aux=subs(cotaError,m,j);
while aux>error
   j=j+1;
   aux=subs(cotaError,m,j);
end
disp('   ')
disp (['El número de términos a coger es  '  num2str(j)])
aproximado=double(symsum(suc,n,1,j));
disp('-------------------------------------------------------------------
----------');
disp(['La suma parcial de orden ' num2str(j) ' es:   '
num2str(aproximado,12)])
disp('-------------------------------------------------------------------
----------');
disp('El valor exacto que da Matlab es:  ')
suma=symsum(suc,n,1,inf);
pretty(suma)
disp(['de valor  '  num2str(double(suma),12) '...'])
disp(' ')
disp('-------------------------------------------------------------------
----------');
disp(['La diferencia entre el valor exacto y el aproximado es '
num2str(double(suma-aproximado),12)])
disp(' ')
```

FUNCIONES DE UNA VARIABLE: EJEMPLOS Y PROGRAMAS

1 Polinomio de Taylor

```
clear;
syms x
disp('  ');
disp('  ');
f='sin(x)';
punto=0;
salir=0;
while salir == 0
   disp(' ')
   orden=input('Da el orden del polinomio de Taylor:  ');
   figura=input('Da el numero de la ventana de dibujo:  ');
   extremoInferior=input('Da el extremo inferior del intervalo donde
quieres la representación:  ');
   extremoSuperior=input('Da el extremo superior del intervalo donde
quieres la representación:  ');
   figure(figura)
   hold on
   puntos=linspace(extremoInferior,extremoSuperior,80);
   poli=taylor(sin(x),orden+1);
   resto2=abs(sin(x)-poli);
   %gamma(n+1)=n!
   estimacion=((abs(x))^(orden + 1))/gamma(orden+2);
   restoNsimo=subs(resto2,puntos);
   estimacionNsima=subs(estimacion,puntos);
   restoNsimo
   plot(puntos,restoNsimo,'r');
   plot(puntos,estimacionNsima,'b');
   title('El resto en rojo, la estimación en azul');
   disp('-----------------------------------------------------------
--------------------')
   disp('Escribe 0 si deseas continuar con otro polinomio');
   disp('Escribe 1 si deseas terminar');
   salir=input(' .... ');
 end
```

2 Acotación del resto de Taylor

```
clear
syms x m
disp(' ');
```

```
disp('-----------------------------------------------------------------
----------');
disp('-----------------------------------------------------------------
----------');
syms n m
suc=input('Da el término general "an" de la serie en función de n:
');
disp(' ');
numTerminos=input('Cuantos términos quieres sumar:    ');
disp('"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
""""""""""')
aproximado=double(symsum(suc,n,1,numTerminos));
disp(' ')
disp(['El valor aproximado es  ' num2str(aproximado,12)])
disp('-----------------------------------------------------------------
----------');
disp('El valor exacto es  ')
suma=symsum(suc,n,1,inf);
pretty(suma)
disp(['de valor ' num2str(double(suma),12) '...'])
disp(' ')
disp('-----------------------------------------------------------------
----------');
cotaErrorInferior=int(suc,n,numTerminos+1,inf);
cotaErrorSuperior=int(suc,n,numTerminos,inf);
disp(' ');
disp(['El error cometido es un valor verificando '
num2str(double(cotaErrorInferior),12) ' <= error <= '
num2str(double(cotaErrorSuperior),12)])
disp('-----------------------------------------------------------------
----------');
disp(' ')
disp(['La diferencia entre el valor exacto y el aproximado es '
num2str(double(suma-aproximado),20)])
disp(' ')
```

FUNCIONES DE VARIAS VARIABLES: EJEMPLOS Y PROGRAMAS

1 Límites direccionales

```
clear
clf;
%PUNTO DONDE SE ESTUDIARA EL LIMITE
disp('    ')
syms x y
funcion=input('Da la expresión de la función con x e y como variables
independientes:    ');
a=input('Da la abscisa del punto:    ');
b=input('Da la ordenada del punto:    ');
disp(' ')
disp('------------------------------------------------------------')
disp('En la figura 1 se muestra el gráfico de superficie ')
disp('En la figura 2 el gráfico de contorno ')
disp(' ')
disp('------------------------------------------------------------')
disp(' ')
%Representamos la gráfica de la función
[X, Y]=meshgrid(a+(-1:.03:1),b+(-1:0.03:1));
Z=subs(funcion,{x,y},{X,Y});
figure(1)
colormap('default')
surfc(X,Y,Z)
shading interp
title('Gráfico de la superficie')
%Las líneas de contorno
figure(2)
colormap gray
contour3(X,Y,Z,20)
title('Lineas de contorno')
disp(' ')
disp('============================================================')
disp(' ')
%LIMITES RADIALES
syms m
radiales=limit(subs(funcion,y,b+m*(x-a)),x,a);
disp(['Los límites radiales son ' num2str(double(radiales))])
%Devuelve 0
%DIBUJO DE UN CAMINO y=F(x)
disp(' ')
disp('------------------------------------------------------------')
disp('pulsa una tecla para ver un camino hacia un punto')
disp(' ')
pause
curva=input('Da la expresión de la curva en función de x:  ');
vx=a+(-1:0.03:1);
vy=subs(curva,x,vx);
vz=subs(funcion,{x,y},{vx,vy});
```

```
figure(1);
colormap gray
hold on
plot(vx,vy,'r')
plot3(vx,vy,vz)
hold off
%LIMITE DE LA FUNCION SEGUN UN CAMINO
disp(' ')
dire=subs(funcion,y,curva);
direccion=limit(dire,x,a);
disp(['El límite siguiendo la dirección dada por la curva es '
num2str(double(direccion))])
%LIMITE "en polares"
disp(' ')
disp('------------------------------------------------------------')
disp(' ')
syms r theta
pol=subs(funcion,{x,y},{a+r*cos(theta),b+r*sin(theta)});
polares=limit(pol,r,0);
disp(['El límite en polares es ' num2str(double(polares))])
```

2 Representación de una función de dos variables. Curvas de nivel y gradiente.

```
clear
clf;
%PUNTO DONDE SE ESTUDIARA EL LIMITE
disp('   ')
syms x y
funcion=input('Da la expresión de la función con x e y como variables
independientes:   ');
xMin=input('Da el valor mínimo para x:   ');
xMax=input('Da el valor máximo para x:   ');
yMin=input('Da el valor mínimo para y:   ');
yMax=input('Da el valor máximo para y:   ');
disp(' ')
disp('------------------------------------------------------------')
disp('En la figura 1 se muestra el gráfico de superficie ')
disp('En la figura 2 el gráfico de contorno ')
disp('En la figura 3 las curvas de nivel y el campo gradiente')
disp(' ')
disp('------------------------------------------------------------')
disp(' ')
%Representamos la gráfica de la función
[X, Y]=meshgrid(xMin:.1:xMax,yMin:0.1:yMax);
Z=double(subs(funcion,{x,y},{X,Y}));
figure(1)
colormap('default')
surfc(X,Y,Z)
shading interp
title('Gráfico de la superficie')
%Las líneas de contorno
figure(2)
%colormap gray
contour3(X,Y,Z,20)
title('Lineas de contorno')
%Las líneas de contorno
figure(3)
```

```
cs=contour(X,Y,Z,20);clabel(cs);
[X1, Y1]=meshgrid(xMin:0.5:xMax,yMin:0.5:yMax);
Z1=double(subs(funcion,{x,y},{X1,Y1}));
[px,py]=gradient(Z1,0.5,0.5);
hold on
quiver(X1,Y1,px,py)
hold off
title('Curvas de nivel y gradiente')
disp(' ')
disp('============================================================')
```

3 Extremos de funciones de dos variables. Método del hessiano.

```
clear
clf;
clear
%Trabajamos en lo que sigue con la función en
%forma simbólica.
syms x y
f=input('Da la expresión de la función con x e y como variables
independientes: ');
disp('**************************************************************
*****')
disp('    PASO 1')
disp('-----------')
disp('La derivada parcial de f respecto a x es:')
fx=diff(f,x);
pretty(fx)
disp(' ')
disp('La derivada parcial de f respecto a y es:')
fy=diff(f,y);
pretty(fy)
disp(' ')
%Resolvemos el sistema fx=0, fy=0.
disp('**************************************************************
*****')
disp('    PASO 2: Resolvemos el sistema fx=0, fy=0. La solución es:')
disp('-----------')
[coordx,coordy]=solve(fx,fy);
numPuntos=length(coordx);
for k=1:numPuntos
   disp(['Punto ' num2str(k) ' = (' num2str(double(coordx(k))) ','
num2str(double(coordy(k))) ')'])
end
%Calculamos las derivadas parciales segundas.
fxx=diff(fx,x);
fxy=diff(fx,y);
fyx=diff(fy,x);
fyy=diff(fy,y);
%Construimos la matriz hessiana y el hessiano.
hessiana=[fxx fxy;fyx fyy];
disp(' ')
```

```
disp('**************************************************************
*****')
disp('    PASO 3: Construimos la matriz hessiana:')
disp('-----------')
pretty(hessiana)
hessiano=det(hessiana);
disp('    ')
disp('**************************************************************
*****')
disp('    PASO 4: Estudiamos para cada punto si es definida positiva
o definida negativa:')
disp('-----------')
disp('    ')
%Estudiamos los puntos con gradiente nulo
for k=1:numPuntos
      px=coordx(k);
      py=coordy(k);
      if and(imag(px)==0,imag(py)==0)
      disp(['PUNTO ' num2str(k) ':  (' num2str(double(px)) ','
num2str(double(py)) ') : '])
      disp(' ')
      disp('    La matriz hessiana en ese punto es ')
      pretty(subs(hessiana,{x,y},{px,py}))
     %Estudiamos el hessiano en el punto
     hes=double(subs(hessiano,{x,y},{px,py}));
      %Evaluamos fxx en el punto.
  val=double(subs(hessiana(1,1),{x,y},{px,py}));
     if and(hes>0,val>0)
          disp('    Es un mínimo relativo.')
     else
          if and(hes>0,val<0)
             disp('   Es un máximo relativo.')
          else
             if hes<0
                 disp('    Es un punto de silla.')
             else
               disp('    Hay que estudiar el signo de la diferencial
segunda.')
             end
            end
     end
     disp('-----------------------------------')
     end
end
disp( ' ')
disp(' ')
```

4 Extremos condicionados

```
clear
clf;
%PUNTO DONDE SE ESTUDIARA EL LIMITE
disp('   ')
syms x y
```

```
funcion=input('Da la expresión de la función con x e y como variables
independientes:   ');
a=input('Da la abscisa del punto:   ');
b=input('Da la ordenada del punto:   ');
disp(' ')
disp('-----------------------------------------------------------')
disp('En la figura 1 se muestra el gráfico de superficie ')
disp('En la figura 2 el gráfico de contorno ')
disp(' ')
disp('-----------------------------------------------------------')
disp(' ')
numPuntos=30;
extremoSuperiorX=2;
extremoInferiorX=-2;
extremoSuperiorY=2;
extremoInferiorY=-2;
incrementoX=(extremoSuperiorX-extremoInferiorX)/numPuntos;
incrementoY=(extremoSuperiorY-extremoInferiorY)/numPuntos;
%Representamos la gráfica de la función
[X,
Y]=meshgrid(extremoInferiorX:incrementoX:extremoSuperiorX,extremoInfe
riorY:incrementoY:extremoSuperiorY);
Z=subs(funcion,{x,y},{X,Y});
figure(1)
colormap('default')
surfc(X,Y,Z)
title('Gráfico de la superficie')
%DIBUJO DE UN CAMINO y=F(x)
disp(' ')
disp('-----------------------------------------------------------')
disp('pulsa una tecla para ver un camino hacia un punto')
disp(' ')
pause
curva=input('Da la expresión de la curva en función de x:  ');
vx=extremoInferiorX:incrementoX:extremoSuperiorX;
vy=subs(curva,x,vx);
vz=subs(funcion,{x,y},{vx,vy});
figure(2);
hold on
plot(vx,vy,'r')
plot3(vx,vy,vz)
colormap gray
[X, Y]=meshgrid(-2:0.1:2,-6:0.1:6);
Z=subs(funcion,{x,y},{X,Y});
surfc(X,Y,Z)
shading interp
hold off
```

5 Plano tangente

```
clear;
clf;
syms x y
disp(' ')
funcion=input('Da la función de dos variables con x e y como
variables independientes:   ');
```

```
puntoX=input('Da la abscisa del punto donde quieres calcular el plano
tangente:   ');
puntoY=input('Da la ordenada del punto donde quieres calcular el
plano tangente:  ');
fx=diff(funcion,x);
fy=diff(funcion,y);
fab=subs(funcion,{x,y},{puntoX,puntoY});
fxab=subs(fx,{x,y},{puntoX,puntoY});
fyab=subs(fy,{x,y},{puntoX,puntoY});
ztangente=fab+fxab*(x-puntoX)+fyab*(y-puntoY);
disp('=================================================')
disp('La ecuación del plano tangente es: ')
pretty(ztangente)
[X, Y]=meshgrid(puntoX+(-0.4:.05:0.4),puntoY+(-0.4:0.05:0.4));
Z1=subs(funcion,{x,y},{X,Y});
Z2=subs(ztangente,{x,y},{X,Y});
num=length(X);
disp(' ')
disp('=================================================')
disp(' ')
disp('Escribe 1 para dar un punto donde evaluar la función y el plano
tangente:')
disp('Escribe 2 para terminar')
valor=input('..... ');
while valor==1
disp('Da un punto "próximo" al punto en el que desarrollas')
abscisa=input('   la abscisa es:   ');
ordenada=input('   la ordenada es:   ');
disp(' ')
disp('-----------------------------------------------------------------
-------')
disp(['El valor de la función en ese punto es '
num2str(double(subs(funcion,{x,y},{abscisa,ordenada})))])
disp(['El valor de ztan en ese punto es '
num2str(double(subs(ztangente,{x,y},{abscisa,ordenada})))])
   disp('Escribe 1 para dar un punto donde evaluar la función y el
plano tangente:')
disp('Escribe 2 para terminar')
valor=disp('..... ')
disp(' ')
end
%for k=1:num
  % for k1=1:num
  % disp(' ')
  % disp('-----------------------------------------------------------
------')
  % disp(['El valor de la función en el punto '
num2str(double(X(k,k1)),4) ',' num2str(double(Y(k,k1)),4) ' es '
num2str(double(Z1(k,k1)),4)])
  % disp(['El valor de ztan en el punto ' num2str(double(X(k,k1)),4)
',' num2str(double(Y(k,k1)),4) ' es ' num2str(double(Z2(k,k1)),4)])
  % end
%end
 %disp(' ')
 %disp('-----------------------------------------------------------
----')
 %disp('En forma de tabla ')
 %disp(' ')
 %disp('El valor de la función es ')
 %double(Z1)
```

```
 %disp('El valor de ztan es ')
 %double(Z2)
 figure(1)
 hold on
colormap('default')
surfc(X,Y,Z1)
%figure(2)
surfc(X,Y,double(Z2))
hold off
```

6 Aproximación por el poliniomio de Taylor de funciones de dos variables

```
clear;
clf;
syms x y
disp(' ')
disp(' ')
funcion=input('Da la función de dos variables con x e y como
variables independientes:   ');
puntoX=input('Da la abscisa del punto:   ');
puntoY=input('Da la ordenada del punto:  ');
disp('')
fx=diff(funcion,x);
fy=diff(funcion,y);
fxx=diff(fx,x);
fxy=diff(fx,y);
fyy=diff(fy,y);
fab=subs(funcion,{x,y},{puntoX,puntoY});
fxab=subs(fx,{x,y},{puntoX,puntoY});
fyab=subs(fy,{x,y},{puntoX,puntoY});
fxxab=subs(fxx,{x,y},{puntoX,puntoY});
fxyab=subs(fxy,{x,y},{puntoX,puntoY});
fyyab=subs(fyy,{x,y},{puntoX,puntoY});
taylorVar=fab+fxab*(x-puntoX)+fyab*(y-puntoY)+1/2*(fxxab*(x-
puntoX)^2+fyyab*(y-puntoY)^2+2*fxyab*(x-puntoX)*(y-puntoY));
disp('-----------------------------------------------------------------
')
disp('  ');
disp(['El polinomio de Taylor de grado 2 centrado en el punto: ('
num2str(puntoX) ',' num2str(puntoY) ') es: '])
pretty(taylorVar)
disp('')
[X, Y]=meshgrid(puntoX+(-1:.06:1),puntoY+(-1:0.06:1));
Z1=double(subs(funcion,{x,y},{X,Y}));
figure(1)
colormap('default')
hold on
surfc(X,Y,Z1)
%figure(2)
Z2=double(subs(taylorVar,{x,y},{X,Y}));
colormap('default')
surfc(X,Y,Z2)
hold off
figure(2)
colormap('default')
```

```
surfc(X,Y,Z2)
title('Polinomio de Taylor de orden 2')
hold off
disp(' ')
valor=1;
disp('==========================================================
')
while valor==1
disp('Da un punto "próximo" al punto en el que desarrollas')
abscisa=input('   la abscisa es:   ');
ordenada=input('   la ordenada es:   ');
disp(' ')
disp('----------------------------------------------------------
')
disp(['El valor de la función en ese punto es '
num2str(double(subs(funcion,{x,y},{abscisa,ordenada})))])
disp(['El valor del polinomio de Taylor en ese punto es '
num2str(double(subs(taylorVar,{x,y},{abscisa,ordenada})))])
disp('   ')
disp('Escribe:')
disp('     1 si deseas evaluar la función y el polinomio de Taylor en
un punto')
disp('     2 para terminar')
valor=input('.........  ');
disp(' ')
end
```