Prácticas Matlab

Práctica 1

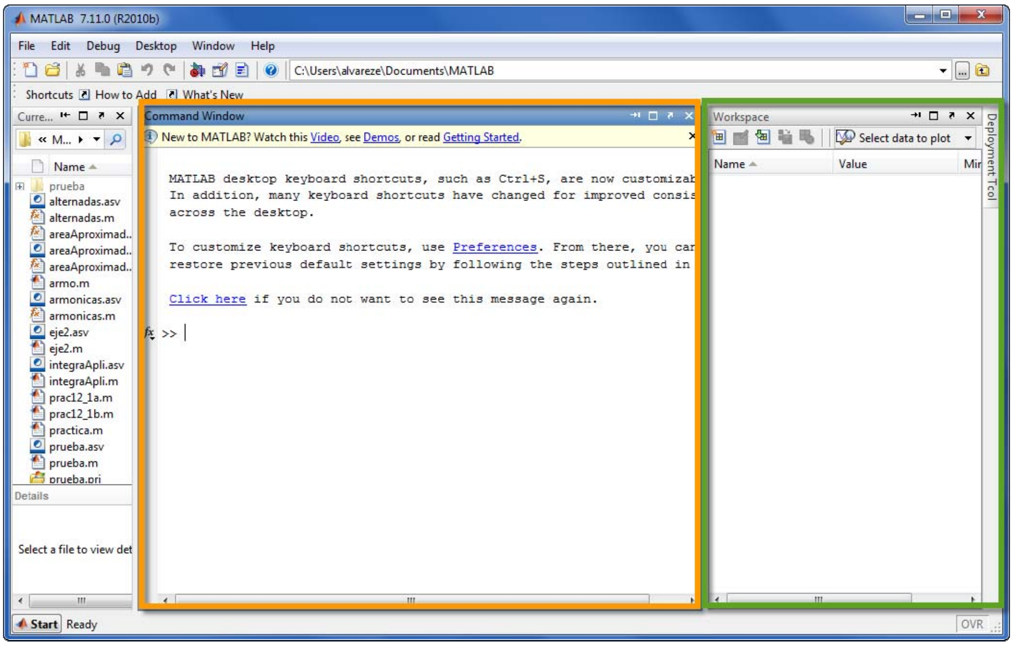
Objetivos

- Iniciarse en el uso de Matlab.
- Conocer comandos básicos de Matlab para realizar cálculos con números reales y números complejos.
- Realizar gráficos sencillos con el comando plot.

¿Qué es Matlab?

El nombre de Matlab es un acrónimo de MATrix LABoratory. Hoy en día Matlab es un programa muy potente con un entorno agradable, que incluye herramientas de visualización gráfica, así como un lenguaje de alto nivel.
La ventana de Matlab muestra un escritorio dividido en varias partes:

- Las órdenes se escriben en la ventana de comandos, *Command Window*.
- La ventana *Workspace* proporciona información sobre las variables utilizadas.



Inicio de sesión

- Introducir un pendrive y crear una carpeta de nombre "*practica1*".
- Entrar en Matlab y hacer que *Current Directory* sea la carpeta "*practica1*".

- Para conservar nuestra sesión de trabajo en un fichero de texto, empezar tecleando:
```
>>diary  practica1
```
- Al terminar la sesión teclear:
```
>>diary off
```

Operaciones elementales

La forma de representar números y de operar con Matlab es la misma que la de una calculadora de bolsillo.
```
>>3.2
```
Las operaciones básicas se hacen con los mismos símbolos y en la misma secuencia que las calculadoras.

Operador	Utilización	Ejemplo
+	Adición	2+3
-	Sustracción	2-3
*	Multiplicación	2*3
/	División	2/3
^	Potenciación	2^3

Para que Matlab ejecute una orden en la ventana de comandos es necesario pulsar la tecla *intro* o salto de línea.
```
>>3+5^3-2
ans=
      126
```

En el ejemplo anterior el resultado se ha guardado en la variable `ans`. Si al final de la orden se escribe un punto y coma (;) su resultado se calcula pero no se escribe en pantalla. Por ejemplo, si escribimos
```
>>3+5^3-2;
```

el valor de `ans` sería `126` pero no nos lo mostraría.
Una variable es un nombre que se da a una entidad que puede ser una matriz, un vector, un escalar. El valor de esa variable, e incluso el tipo de la entidad que representa, puede variar a lo largo de una sesión de Matlab. Para asignar un valor a una variable se escribirá:

```
nombreVariable=expresión
```

Si quisiéramos guardar el resultado en otra variable escribiríamos
```
>>s=3+5^3-2
```

El nuevo valor de la variable `s` es `126`.

Reglas para nombrar variables

- El nombre de una variable puede tener como máximo 63 caracteres que pueden ser letras, números y el guion de subrayar
- El primer carácter tiene que ser una letra, `modulo2` es un nombre válido, pero no lo es `2modulo`.

- Las mayúsculas y las minúsculas tienen valor distintivo. La variable `Modulo` es distinta de la variable `modulo`.
- Dentro de un nombre de variable no puede haber espacios en blanco, `modulo1` es un nombre de variable válido, pero no `modulo 1`.
- Existen nombres que deben evitarse porque tienen significado propio en Matlab: `ans, pi, Inf, i, . . .`

Ayuda de Matlab

En esta primera práctica trabajaremos únicamente en la *Command Window* (Ventana de Comandos) de Matlab.

1.- Cómo obtener ayuda desde la Command Window :

Ejecutar *help*, *lookfor*, *doc*, o *type* seguido del comando del que se requiere la ayuda.
Ejemplo

```
>>help plot
>>lookfor graph
>>doc plot
>>type linspace
```

2.- Cómo encontrar comandos de Matlab :

- Ir a *Product help* en el menú *Help* de la ventana principal de Matlab.
- En *MATLAB* buscar el tema que interese.

Algunas funciones matemáticas

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 seno de x con imagen en el rango [-π/2, π/2]	asin(1)=1.5708
acos(x)	Arco coseno de x con imagen en [-π/2, π/2]	acos(1)=-6.1257e-17
atan(x)	Arco tangente de x con imagen en el rango [-π/2, π/2]	atan(1)=0.7854
atan2(y,x)	Arco tangente de y/x con imagen en el rango [-π, π]	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

Comandos para trabajar con vectores

Matlab es un programa que trabaja fundamentalmente con vectores y matrices.
Para definir un **vector fila** se puede:

- o Introducir sus componentes separadas por un espacio o una coma
```
>> w=[1 4 9]
```
- o Introducir sus componentes, especificando el valor de cada componente
```
>> w[1]=1, w[2]=4, w[3]=9
```
- o Utilizar el operador, `colon (:).` El comando
```
    a:h:b
```
genera un vector fila de primer elemento a y los demás elementos aumentan de h en h hasta no superar b.

```
>> v=2:9
%  Devuelve   v = 2   3   4   5   6   7   8   9
>> w=-5:2:5
% Devuelve w = -5  -3   -1   1   3   5
```
- o Utilizar el comando
```
    linspace(a,b,n)
```
que genera un vector fila de n componentes cuyo primer elemento es a y el último b, siendo todos sus elementos equidistantes.

```
>>   w=linspace(-5,5,10)
% Devuelve w= -5.0000  -3.8889  -2.7778  -1.6667   -0.5556
0.5556   1.6667   2.7778   3.8889   5.0000
```
w es un vector de 10 números igualmente espaciados entre -5 y 5.

Si se quiere definir un vector columna basta hacer lo mismo que para un vector fila pero separando por un punto y coma cada fila.
```
>>w=[-1;2;3]
>>a=[1 2 3; 4 5 6] %matriz de 2 filas y 3 columnas
```

Si a y b son matrices y `r` es un escalar, la forma de indicar a Matlab que realice las operaciones algebraicas habituales es mediante los operadores ya vistos de suma (+), producto (*) y exponenciación (^). Para poder realizar estos cálculos únicamente es necesario que los vectores y matrices tengan la dimensión adecuada para que la operación pueda realizarse.
```
    >>a*b+r*a^2
```

Si queremos realizar estas operaciones algebraicas “elemento a elemento” los operadores que debemos utilizar son los que se muestran a continuación.

Operadores entre vectores	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]

Las funciones matemáticas que se han visto en el apartado anterior pueden aplicarse también a un vector. Por ejemplo, si se quiere calcular el seno a cada componente de un vector basta escribir

```
>>v=0:pi/4:pi;
>>sin(v)
```

Como hemos comentado anteriormente, uno de los aspectos más destacables de Matlab es su capacidad para trabajar con vectores y matrices y, en consecuencia, son muchos los comandos de los que se dispone para su manipulación. A modo de ejemplo:

```
>>v=1/2:1/3:3;
>>sum(v)      %suma las componentes del vector v
>>length(v)   %devuelve el número de elementos de v
```

Comandos para operar con números complejos

i (ó j)

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
```

```
>>  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
```

Comandos para representar puntos

plot(x,y)

dibuja una línea que une los puntos de abscisas el vector “x” y ordenadas“y”.

plot(y)

dibuja una línea que une los puntos del 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
```

figure(n)

Para crear una ventana de dibujo

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
```

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));
```

Comandos para construir variables simbólicas

sym(A)

Genera una expresión simbólica a partir de la expresión A. Si A es un valor numérico, el resultado es la representación simbólica de dicho valor.

Ejemplo:

```
>>  sym(angle(-1-i))       % Devuelve -3pi/4
```

Ejercicios

1

En este primer ejemplo veremos cómo utilizar Matlab como una calculadora.

a) Calcula el siguiente valor $(1+i)^4$
b) Calcula la suma de los primeros 100 números naturales.
c) Calcula la suma de los cubos de los primeros 100 números naturales.

Indicaciones

a)

Para hacer el cálculo a mano:

Pasa previamente el complejo $1+i$ a forma exponencial,

$(1+i)^4 =$

Para comprobar el resultado con Matlab:

- Escribe el complejo $1+i$ y utiliza el operador ^ para elevarlo a la potencia cuarta.

b)

Para hacer el cálculo a mano

Esta suma se puede representar mediante el sumatorio, $s=\sum_{k=1}^{100} k = 1+2+3+4+\ldots+100$

¿Cómo sumar a mano estos 100 primeros números naturales?

s=	1	+ 2	+ 3	+ 4	+...+	100
s=	100	+99	+98	+97	+...+	+ 1
2s=	101	101	101	101	...	101

$$s=\frac{100\cdot 101}{2}=5500$$

¿Sabrías obtener una fórmula general para calcular la suma de cualquier número "n" de números naturales?

$$s=\sum_{k=1}^{n} k=1+2+3+4+\ldots+n$$

Para comprobar el resultado con Matlab:

Utiliza los siguientes comandos ,

- `colon(:)`, para crear un array vector con los números desde 1 hasta 100.
- `sum`, para sumar los elementos del vector anterior.

c)

Para hacer el cálculo con Matlab:

Utiliza los siguientes comandos ,

- `colon(:)`, para crear un array vector con los números desde 1 hasta 100.
- El operador `.^`, para elevar al cubo cada elemento del vector anterior.
- `sum`, para sumar los elementos del último vector.

$$\sum_{k=1}^{100} k^3=1^3+2^3+3^3+\ldots+100^3=$$

2

Escribe en forma binómica, exponencial y trigonométrica los siguientes números complejos

a) $z_1=\frac{1}{\sqrt{3}}\cdot\left(\cos\frac{2\pi}{3}+i\,\mathrm{sen}\frac{2\pi}{3}\right)$

b) $z_2=\frac{(5i)\cdot\left(2_{\pi/6}\right)^4}{\left(4_{\pi/12}\right)^3\cdot\left(10_{5\pi/12}\right)}$

c) $z_3=4\sqrt{2}\cdot e^{-\frac{\pi}{6}i}$

d) $z_4=\left(\frac{10\sqrt{3}+10i}{5+5i}\right)^{-6}$

Indicaciones

Este ejercicio es el número 7 de los propuestos en el tema de complejos por lo que se asume que ya se ha resuelto a mano y solo resta comprobar los resultados con Matlab.

a)

Para comprobar el resultado con Matlab:

- Escribe el complejo `z1` directamente en forma trigonométrica, utilizando los operadores aritméticos y las funciones que se requieren.
- Escribe `sym(z1)` para obtener la representación simbólica del valor numérico de `z1`.

b)

Para comprobar el resultado con Matlab:

- Previamente pasa los complejos de forma polar a trigonométrica o exponencial.
- Haz por una parte las operaciones del numerador y guárdalas en la variable `w1`, y por otra las operaciones del denominador y guárdalas en la variable `w2`.
- Haz `>> z2=w1/w2`.
- Escribe `sym(z2)` para obtener la representación simbólica de z2. ¿Cómo es la parte real del resultado?

c)

Para comprobar el resultado con Matlab:

- Escribe el complejo `z3` directamente en forma exponencial, utilizando los operadores aritméticos y las funciones que se requieren.
- Escribe `sym(z3)` para obtener la representación simbólica de `z3`.

d)

Para comprobar el resultado con Matlab:

- Escribe el complejo `z4` directamente en la forma del enunciado, utilizando los operadores aritméticos y las funciones que se requieren.
- Escribe `sym(z4)` para obtener la representación simbólica de `z4`.

3

Sea la ecuación $w = Az + B \quad z \in \mathbb{C}$, $A = 1+i$, $B = 2+i$. Esta ecuación transforma puntos (x, y) en el plano complejo *z*, en puntos (u, v) en el plano complejo *w*.

(a) Realizar la transformación definida por *w* para los puntos del plano z.

$$z1 = 0, \quad z2 = 1+\sqrt{3}\,i, \quad z3 = 2$$

(b) Dibujar el triángulo definido por los puntos anteriores en color rojo.

(c) Dibujar el triángulo transformado en color verde y en la misma figura.

(d) Comprobar que tanto el triángulo en el plano *z* como el triángulo transformado son equiláteros.

Indicaciones

Este ejercicio es el número 3 de los propuestos en el tema de complejos por lo que se asume que ya se ha resuelto a mano y solo resta comprobar los resultados con Matlab.

Para comprobar el resultado con Matlab:

- Genera un vector fila al que llamarás z cuyos elementos sean z1, z2 y z3.
- Introduce las constantes A y B y define la transformación $w = Az + B$..
  a) Dibuja el triángulo definido por los puntos del vector z, utilizando el comando `plot(real(z),imag(z))`.
  b) Utiliza `hold on` para dibujar en la misma figura el triángulo definido por los puntos del vector w, con el comando `plot(real(w),imag(w))`.
  c) Para comprobar que los triángulos son equiláteros calcula las longitudes de los lados con el comando `abs` y comprueba que son iguales.

Práctica 2

Objetivos

- Dibujar gráficas de funciones definidas a trozos con el comando *Plot*.
- Dibujar funciones implícitas con el comando *ezplot*.
- Calcular límites de funciones en puntos concretos de su dominio.
- Representar gráficas de funciones

Comandos de Matlab

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 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.

Ejemplos resueltos

1

Representación de funciones.

(a) Forma explícita: $y=\left|x^2-9\right|+4$ en el intervalo [-5, 5]

(b) Forma implícita: $x^2+4y^2-3xy-5=0$

(c) Forma paramétrica: $\begin{cases} x(t)=a\cos(nt)\cos(t) \\ y(t)=a\cos(nt)sen(t) \end{cases}$ con el valor del parámetro t en un cierto intervalo, por ejemplo, $t\in[0,2\pi]$.

Según distintos valores de a y n se obtienen distintas curvas. Por ejemplo si n=1 la curva es una circunferencia. Puedes probar con los valores: n=1/2, n=2/3, n=9/2 y a=5.

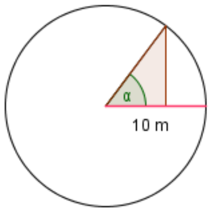
Solución:

```
% Apartado a
x1=-5:0.1:5;
y1=abs(x1.^2-9)+4;
plot(x1,y1)
axis([-5 5 -1 19])
grid on

% Apartado b
ezplot('x^2+4*y^2-3*x+y-5',[-5,5])
grid on

% Apartado c
% Linspace genera 100 puntos entre dos números dados,
% se puede indicar también el número de puntos de la
% forma linspace(x1,x2,n)
% Consideramos 200 puntos en el intervalo [0, 4*pi]
t=linspace(0,4*pi,200);
a=5;n=9/2;
x3=a*cos(n*t).*cos(t);
y3=a*cos(n*t).*sin(t);
plot(x3,y3)
```

```
axis([-10 10 -10 10])
```

2

Considera la región sombreada



a. Expresa el área de la región en función de α, $A=f(\alpha)$, cuando $\alpha\in[0,\pi]$.

b. Haz a mano una representación gráfica de la función $y=f(\alpha)$ en el intervalo $[0,\pi]$ a partir de la gráfica de la función seno y comprueba después con Matlab el resultado.

c. A la vista de la gráfica obtenida en el apartado b, ¿podrías hacer una representación aproximada de la función derivada?

d. Representa con Matlab en una misma figura $y=f(\alpha)$ y $y=f'(\alpha)$ en $\left[0,\frac{\pi}{2}\right]$. Indica qué información se puede obtener de la gráfica de f para representar f' y qué información se podría obtener de f' para representar a partir de ella la de f.

e. ¿Cuál es el valor máximo de esta área en el intervalo $\left[0,\frac{\pi}{2}\right]$

Solución

a) Teniendo en cuenta la figura, la función área será:

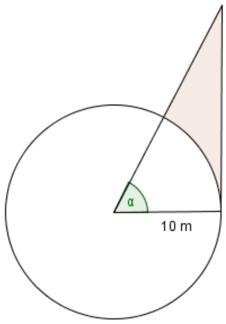
$$A=\begin{cases} f(\alpha)=\dfrac{10\text{sen}\alpha\, 10\cos\alpha}{2}=50\text{ sen}\alpha\ \cos\alpha=25\text{sen}(2\alpha) & si\,\alpha\in\left[0,\dfrac{\pi}{2}\right] \\ f(\pi-\alpha) & si\,\alpha\in\left[\dfrac{\pi}{2},\pi\right] \end{cases}$$

```
% Apartado b
clear all
t1=0:0.01:pi/2;
t2=pi/2:0.01:pi;
t=[t1 t2];
% f en el intervalo [0, pi/2]
f1=25*sin(2*t1);
% f en el intervalo [pi/2, pi]
f2=25*sin(2*(pi-t2));
plot(t1,f1,t2,f2,'b')

% Apartado d - Representación en [0, pi/2]
syms x
f=25*sin(2*x);
derif=diff(f,x);
plot(t1,f1,t1,subs(derif,x,t1))
legend('f', 'f`')
```

```
% Apartado d - Representación en [0 pi]
% Abrimos otra ventana de dibujo
figure(2)
% En este caso le indicamos la derivada
derf1=50*cos(2*t1);
derf2=-50*cos(2*(pi-t2));
plot(t1,derf1,t2,derf2,'b')


% Apartado e
xPunto=solve(derif)
```

3

Considerar la región que queda fuera del sector del círculo con radio 10 m y dentro del triángulo rectángulo de la figura



a) Expresar el área de la región en función de α, $A=f(\alpha)$. Determinar el dominio de la función. Recuerda que el área de un sector circular de radio r para un ángulo α medido en radianes es $S=\dfrac{r^2\alpha}{2}$

b) Utiliza Matlab para completar las siguientes tablas

α	$\frac{\pi}{6}$	$\frac{\pi}{5}$	$\frac{\pi}{4}$	$\frac{\pi}{3}$
$f(\alpha)$				

α	$\frac{\pi}{2}-10^{-1}$	$\frac{\pi}{2}-10^{-2}$	$\frac{\pi}{2}-10^{-3}$	$\frac{\pi}{2}-10^{-4}$
$f(\alpha)$				

c) Calcula el límite cuando $\alpha \to \left(\dfrac{\pi}{2}\right)^-$

Solución

a) Teniendo en cuenta la figura, la función área es:

$$A=f(\alpha)=\frac{100}{2}tg\alpha-\frac{100}{2}\alpha=50\,(tg\alpha-\alpha)$$

```
% Apartado b1
clear all
denom=6:-1:3;
t=pi./denom;
f1=50*(tan(t)-t);
disp('   alpha        f(alpha)')
disp([t' f1'])


% Apartado b2
t=pi/2-10.^(-1:-1:-4);
disp(' Más próximo a pi/2')
f1=50*(tan(t)-t);
disp('   alpha        f(alpha)')
format long
disp([t' f1'])

% Apartado c
syms x
limit(50*(tan(x)-x),x,pi/2,'left')
```

Ejercicios propuestos

1

Dibujar la gráfica de las siguientes funciones dadas en implícitas (ver ejercicios propuestos número 6j y 6k del tema 1 de Números complejos).

a) La elipse que tiene sus focos en (1, 0) y (-1,0) y viene dada por la expresión

$$|z-1|+|z+1|=4$$

$$\sqrt{(x-1)^2+y^2}+\sqrt{(x+1)^2+y^2}=4$$

b) La hipérbola que tiene sus focos en (1, 0) y (-1,0) y viene dada por la expresión

$$\left||z-1|-|z+1|\right|=\frac{1}{4}$$

$$\left|\sqrt{(x-1)^2+y^2}-\sqrt{(x+1)^2+y^2}\right|=\frac{1}{4}$$

2

Se considera la función $f(x)=\dfrac{10x+5}{x^4+x^3+2x-4}$.

a) ¿Cuál es su dominio?

b) Calcular los siguientes límites:

$$\lim_{x\to\infty} f(x);\ \lim_{x\to 0} f(x);\ \lim_{x\to 1^+} f(x);\ \lim_{x\to 1^-} f(x)$$

c) ¿Está acotada?

Prácticas Matlab

Práctica 3 (19/10/2012)

Objetivos

- Repasar, mediante ejemplos, la definición de polinomio de Taylor.
- Ayudar a comprender la aproximación local que proporcionan los polinomios de Taylor observando la incidencia que tiene en la aproximación el grado del polinomio de Taylor y la cercanía al punto en el que se hace el desarrollo.

Comandos de Matlab

1.- Para representar funciones

plot(x,y)

dibuja una línea que une los puntos de abscisas el vector "x" y ordenadas"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	

figure(n)

Para crear una ventana de dibujo

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 ver más opcioes teclea la orden:
>>  help plot
```

2.- 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)
```

3.- Para dibujar la gráfica de una función simbólica

```
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])
```

Ejemplo:

```
>>   %Representar una función implícita
```

>> % $xy^2 - x^2 - 2y = 0$, $x \in [-6,6]$

```
>>    ezplot('x*y^2-x^2-2*y',[-6,6])
```

4.- 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)
```

5.- 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)
```

6.- Para simplificar las expresiones simbólicas:

`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.

7. 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)
```

8.- Para calcular el polinomio de Taylor de grado n de la función f en el punto a

```
taylor(f,n+1,a)
```

Ejemplo

```
>>syms x
>>f=x*sin(x+1);
>>taylor(f,5,0)
%Devuelve el polinomio de Taylor de f en el punto 0
%de grado 4.
```

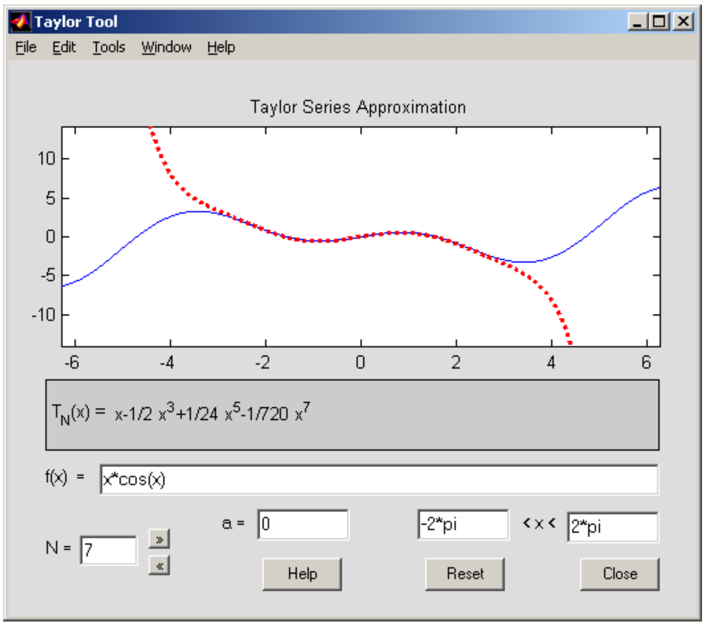
9.- Herramienta taylortool:

Matlab posee una herramienta que permite obtener el polinomio de Taylor de una función y la representación gráfica del polinomio junto con la de la función.

Ejecuta en la ventana de comandos la orden:

>> taylortool

Se abrirá una ventana (ver figura) en la que puedes introducir la función, el grado del polinomio y el intervalo en el que quieres representar la función y el correspondiente polinomio.



En el ejemplo de la figura se trata del polinomio de Taylor centrado en el punto $a = 0$ de grado 7 para la función $f\left(x\right) = x\cos x$ en el intervalo $\left[-2\pi, 2\pi\right]$.

Con los botones en forma de flecha puedes incrementar y/o disminuir el grado del polinomio.

Observa que a medida que el grado del polinomio aumenta el polinomio de Taylor aproxima mejor a la función y en un intervalo más grande.

Ejercicios

1	a) Obtener el polinomio de Taylor de grado 3 de la función $y=\sqrt{1-\frac{x}{2}}$ alrededor del punto *a*= 0. b) Obtener, mediante el polinomio anterior, un valor aproximado de $\sqrt{0{,}5}$. c) Hallar una cota del error cometido en dicha aproximación. d) Comprobar con Matlab cómo la aproximación de $\sqrt{0{,}5}$ es mejor cuánto mayor sea el grado del polinomio de Taylor que se utilice. Escribir una tabla con la aproximación que dan los polinomios de Taylor de grado 1, 3, 5, 20.

Indicaciones

Este ejercicio es el ejercicio propuesto nº11 del tema 2.

a) Calcula el polinomio de Taylor de grado 3 con el comando `Taylor` y guardarlo en una variable de nombre, por ejemplo, pol3.

b) Comprueba que el valor de x para el que se cumple $\sqrt{0{,}5}=\sqrt{1-\frac{x}{2}}$ es $x=1$. Sustituye después en la expresión de `pol3` ese valor con el comando `subs`.

c) Para acotar el error utiliza la fórmula de Lagrange.

EXPRESIONES DEL RESTO: Sea f es una función derivable $n+1$ veces en un intervalo abierto I, que contenga al punto a. Si $R_n\left[f(x);a\right]$ es el resto enésimo de Taylor correspondiente a la función f en el punto $x=a$ entonces:

$$R_n\left[f(x);a\right]=\frac{f^{(n+1)}(t)}{(n+1)!}(x-a)^{n+1}$$

siendo t un punto intermedio entre a y x.

En nuestro caso, la función que estamos considerando es $f(x)=\sqrt{1-\frac{x}{2}}$. Queremos aproximar para $x=1$ el valor de la función por el del polinomio de Taylor de grado $n=3$ desarrollado en el punto $a=0$, esto es,

$$f(1)\approx T_3(1)$$

Considerando el resto de Lagrange la diferencia entre estos dos valores es

$$f(1)-T_3(1)=\frac{f^{(4}(t)}{4!}(1-0)^4 \qquad (*)$$

siendo t un punto intermedio entre 0 y 1.

Realiza entonces los siguientes pasos con Matlab para obtener una cota de la aproximación $f(1)\approx T_3(1)$

1. Calcula la derivada cuarta de la función (comando `diff`) y represéntala (comando `plot`).
2. A partir del dibujo, encuentra un valor M que sea cota superior del valor absoluto de esta derivada en el intervalo [0, 1].
3. Escribe finalmente la cota del error utilizando la expresión del resto de Lagrange vista en (*) teniendo en cuenta que:

$$\left|f(1)-T_3(1)\right|=\left|\frac{f^{(4}(t)}{4!}(1-0)^4\right|\leq\frac{M}{4!}$$

d) Rellena la siguiente tabla

	Polinomio de Taylor $T_n(x)$	Aproximación $T_n(1)$	$f(1)-T_n(1)$
n=1			
n=3			
n=5			
n=20			

2

Se considera $f(x)=\log\left(\frac{x+2}{2x-2}\right)$. Se pide:

a) Representar el dominio de la función $f(x)$.

b) Calcular el polinomio de Taylor de grado n de $f(x)$ en $a=4$ y determina la expresión del resto enésimo de Lagrange de $f(x)$ en $a=4$.

c) Calcular una cota del error cuando queremos aproximar $\log\left(\frac{6'1}{6'2}\right)$ por el polinomio de grado 3.

d) ¿Cuál es el grado del polinomio de Taylor de $f(x)$ en $a=4$ que habría que considerar para aproximar $\log\left(\frac{6'1}{6'2}\right)$ con un error menor que 10^{-1}?

e) Calcular con Matlab el grado del polinomio de Taylor que aseguraría la aproximación de $\log\left(\frac{6'1}{6'2}\right)$ con un error menor que 10^{-6}

Indicaciones

Este ejercicio es el ejercicio propuesto nº12 del tema 2.

a) Calcula a mano los valores de x para los cuales $\frac{x+2}{2x-2}>0$. Recuerda que para que un cociente sea positivo el numerador y el denominador deben tener el mismo signo.

b) Calcula las primeras derivadas a mano teniendo en cuenta que

$$f(x)=\log\left(\frac{x+2}{2x-2}\right)=\log(x+2)+\log(2x-2) \quad si\ x>1$$

Puedes comprobar con Matlab que no te has confundido utilizando el comando `diff`.

Intenta obtener después una expresión para la derivada enésima.

Escribe a continuación la expresión del resto de Lagrange pedido.

c) Determina en primer lugar el valor de x que cumple que $f(x) = \log\left(\frac{6'1}{6'2}\right)$.

$x =$

Obtén después la expresión del resto de orden 1 (R_1) y acótalo

$R_1 =$

$|R_1| \leq$

Si la cota obtenida es menor que 10^{-1}, el grado del polinomio a considerar sería 1, en caso contrario, debes repetir el proceso considerando el polinomio de grado 2. Este proceso deberás repetirlo hasta encontrar un n de forma que el resto enésimo cumpla que es menor que 10^{-1}.

El valor de n es:

d) Repite los pasos seguidos en el apartado c) considerando ahora 10^{-6} en lugar de 10^{-1}.

El valor de n es:

Observaciones:

- La aproximación del polinomio de Taylor es local. En puntos alejados del punto en el que se hace el desarrollo del polinomio el valor de éste y la función pueden no ser próximos.
- Considerando un punto x suficientemente próximo al punto el que se hace el desarrollo, a, la aproximación de $f(x)$ por su polinomio de Taylor es mejor cuanto más grande sea el grado del polinomio, n.
- Considerando un valor n cualquiera, la aproximación de $f(x)$ por la del polinomio de Taylor con dicho grado es mejor cuánto más cerca esté x del punto en el que se hace el desarrollo, a.

Resumen de comandos

Estos son los comandos utilizados en esta práctica que se darán por conocidos en las prácticas siguientes y que conviene retener porque se podrán preguntar en las distintas pruebas de evaluación.

- Para operar con variables simbólicas: `syms, diff, subs`
- Para representar funciones por puntos: `plot`
- Para calcular el polinomio de Taylor: `taylor`

Prácticas Matlab

Práctica 4 (26/10/2012)

Objetivos

- Aproximar el valor de las derivadas primera y segunda en un punto, utilizando fórmulas de derivación numérica obtenidas a partir de la Fórmula de Taylor.

Comandos de Matlab

Todos los comandos que se utilizarán en esta sesión se han explicado en prácticas anteriores.

Derivación numérica

Considerando que se tiene el valor de una función en n puntos equidistantes, x_1, ..., x_n (llamamos $h = x_{i+1} - x_i$)

x	x_1	x_2	...	x_n
$y = f(x)$	$f_1 = f(x_1)$	$f_2 = f(x_2)$	...	$f_n = f(x_n)$

estamos interesados en calcular una aproximación de la derivada primera, segunda, etc. en dichos puntos

x	x_1	x_2	...	x_n
$y = f'(x)$	$f_1' = f'(x_1)$	$f_2' = f'(x_2)$	...	$f_n' = f'(x_n)$
$y = f''(x)$	$f_1'' = f''(x_1)$	$f_2'' = f''(x_2)$	...	$f_n' = f''(x_n)$
...	...	...	...	...

Para calcular la derivada primera, consideramos el polinomio de Taylor de f en $a = x_i$ para el punto $x = x_{i+1}$. Con la notación anterior se tendría

$$f_{i+1} = f_i + hf_i' + \frac{h^2}{2} f_i'' + \frac{h^3}{6} f_i''' + \frac{h^4}{24} f_i^{iv} + \dots \qquad (1)$$

Despejando f_i' la expresion de la diferencia progresiva será

$$f_i' = \frac{f_{i+1} - f_i}{h} + E, \quad E \approx -\frac{1}{2} h f_i''$$

siendo el error de la aproximación, E, del orden de h.

Escribiendo ahora el polinomio de Taylor en el punto $a = x_i$ para $x = x_{i-1}$ se obtendría

$$f_{i-1} = f_i - hf_i' + \frac{h^2}{2} f_i'' - \frac{h^3}{6} f_i''' + \frac{h^4}{24} f_i^{iv} - \ldots \qquad (2)$$

Despejando f_i' la aproximación de la diferencia regresiva será

$$f_i' = \frac{f_i - f_{i-1}}{h} + E, \quad E \approx \frac{1}{2} h f_i''$$

siendo el error de la aproximación del orden de h también.

Restando finalmente las dos expresiones (1) y (2), se llegaría a la aproximación por diferencias centrales

$$f_i' = \frac{f_{i+1} - f_{i-1}}{2h} + E, \quad E \approx -\frac{1}{6} h^2 f_i'''$$

consiguiendo que el error de la aproximación sea del orden de h^2.

Siguiendo un proceso análogo se podrían obtener distintas fórmulas que aproximarían tanto la derivada primera como las derivadas sucesivas basándose en la utilización de los polinomios de Taylor. Al final de la práctica se han incluido algunas de estas fórmulas.

Ejercicios

1

Dada la siguiente tabla de datos:

x	0	0.25	0.5	0.75	1.0	1.25	1.5
y	1.2	1.1035	0.9250	0.6363	0.20	-0.4309	-1.3125

SE PIDE:

1.- Calcular una aproximación de la derivada en $x = 0.5$ utilizando las diferencias de dos puntos siguientes:

a) Diferencia progresiva con h=0.5 y h=0.25.

b) Diferencia regresiva considerando h=0. 5 y h=0.25.

c) Diferencia central considerando h=0.5 y h=0.25.

2.- Sabiendo que los datos se tomaron de la función,

$$f(x) = -0.1x^4 - 0.15x^3 - 0.5x^2 - 0.25x + 1.2$$

comparar los valores obtenidos mediante diferencias con la expresión que se obtendría derivando directamente la función y sustituyendo x=0.5 ($f'(0.5) = -0.9125$).

3.- Calcular una aproximación de la derivada en todos los puntos considerando h=0.25.

Indicaciones

Este ejercicio es el ejercicio propuesto nº21 del tema 2.

Para contestar a los dos primeros apartados del ejercicio utiliza Matlab y completa la siguiente tabla:

	Valor aproximado de $f'(0.5)$	$\dfrac{\lvert valor\ aproximado - valor\ real \rvert}{\lvert valor\ real \rvert} \cdot 100$
h=0.5		
h=0.25		

Código Matlab

Con ayuda de Matlab aproxima la derivada primera utilizando diferencias progresivas, regresivas o centrales en todos los puntos considerando h=0.25 y rellena después la siguiente tabla.

i	x_i	f_i	Progresiva $f_i' \approx \frac{f_{i+1}-f_i}{x_{i+1}-x_i}$	Regresiva $f_i' \approx \frac{f_{i-1}-f_i}{x_{i-1}-x_i}$	Central $f_i' \approx \frac{f_{i+1}-f_{i-1}}{2h}$
1	0				
2	0.25				
3	0.5				
4	0.75				
5	1				
6	1.25				
7	1.75				

Para rellenar la segunda fila, puedes utilizar el siguiente código

```
puntos=[0 0.25 0.5 0.75 1 1.25 1.75];
valores=[1.2 1.035 0.9250 0.6363 0.20 -0.4309 -1.3125];
%Para i=2
%Diferencia progresiva
derProg=(valores(3)-valores(2))/(puntos(3)-puntos(2))
%Diferencia regresiva
derReg=(valores(1)-valores(2))/(puntos(1)-puntos(2))
%Diferencia central
derCent=(derProg+derReg)/2
```

Haz los cambios necesarios para rellenar las filas 3, 4, 5 y 6. Las filas 1 y 7 se rellenarían únicamente con la única aproximación que es posible.

Estos cálculos se pueden obtener de forma más rápida utilizando un ciclo for. En Matlab esta estructura se escribe, en su versión más simple, de la forma siguiente

```
for i=valorInicial:valorFinal
  % instrucciones matlab a realizar
end
```

Resumen de comandos

Estos son los comandos utilizados en esta práctica que se darán por conocidos en las prácticas siguientes y que conviene retener porque se podrán preguntar en las distintas pruebas de evaluación.

- Para calcular el valor de una función simbólica en un punto: `syms, subs`
- Para obtener la derivada de una función simbólica en un punto: `diff`

Fórmulas de Derivación Numérica para la derivación primera

(a) Aproximación por diferencia progresiva

Dos puntos: $$f_i' = \frac{f_{i+1} - f_i}{h} + E, \qquad E \approx -\frac{1}{2} h f_i''$$

Tres puntos: $$f_i' = \frac{-f_{i+2} + 4f_{i+1} - 3f_i}{2h} + E, \qquad E \approx \frac{1}{3} h^2 f_i'''$$

(b) Aproximación por diferencias regresivas

Dos puntos: $$f_i' = \frac{f_i - f_{i-1}}{h} + E, \qquad E \approx \frac{1}{2} h f_i''$$

Tres puntos: $$f_i' = \frac{3f_i - 4f_{i-1} + f_{i-2}}{2h} + E, \qquad E \approx \frac{1}{3} h^2 f_i'''$$

(c) Aproximación por diferencias centradas

Dos puntos: $$f_i' = \frac{f_{i+1} - f_{i-1}}{2h} + E, \qquad E \approx -\frac{1}{6} h^2 f_i'''$$

Cuatro puntos: $$f_i' = \frac{-f_{i+2} + 8f_{i+1} - 8f_{i-1} + f_{i-2}}{12h} + E, \qquad E \approx \frac{1}{30} h^4 f_i^{(5}$$

Fórmulas de Derivación Numérica para la derivación segunda

(a) Aproximación por diferencias progresivas

$$f_i'' = \frac{f_{i+2} - 2f_{i+1} + f_i}{h^2} + E, \qquad E \approx -h f_i'''$$

(b) Aproximación por diferencias regresivas

$$f_i'' = \frac{f_i - 2f_{i-1} + f_{i-2}}{h^2} + E, \qquad E \approx h f_i'''$$

(c) Aproximación por diferencias centradas

$$f_i'' = \frac{f_{i+1} - 2f_i + f_{i-1}}{h^2} + E, \qquad E \approx -\frac{1}{12} h^2 f_i^{(4}$$

Prácticas Matlab

Práctica 5 (2/11/2012)

Objetivos

- Profundizar en el concepto de suma enésima y resolver casos prácticos con ayuda de Matlab.
- Analizar la convergencia de una serie numérica con ayuda de Matlab y contrastar los resultados experimentales con los teóricos.
- Sumar series numéricas.

Comandos de Matlab

1.- Para obtener el limite de una expresión simbólica cuando la variable n tiende al valor a

```
limit(f,n,a)
```

Ejemplo:

```
>> syms x
>> f=sin(5*x)/x; limit(f,x,3); limit(f,x,inf)
```

2.- Para calcular la suma de una expresión simbólica entre dos valores dados a y b.

```
symsum(f,a,b)
```

Ejemplo:

```
>> syms n
>> symsum(1/n,5,8);symsum(1/n,1,inf);
```

3. Para calcular la suma de las componentes de un vector

```
sum(vector)
```

Ejemplo:

```
>> vector=1:100
>> sum(vector);
```

4. Para dibujar en el plano los puntos de una sucesión

```
plot(n,an,'s')
```

Los valores posibles del parámetro “s” se indicaron en la práctica 2.

Ejemplo:

```
>> n=1:10;
>> an=1./n;
>> plot(n,an,'or') %dibuja puntos con círculos rojos
```

Ejercicios

1

Sumas enésimas.

(a) Calcular la suma de los n primeros números naturales y comprobar que $S_n = 1+2+\ldots+n = \dfrac{n(n+1)}{2}$.

(b) Calcular la suma de los cuadrados de los n primeros números naturales y comprobar que $S_n = 1+2^2+\ldots+n^2 = \dfrac{n(n+1)(2n+1)}{6}$

(c) Calcular la suma de los n primeros términos de la sucesión $a_n = \dfrac{(-1)^{n-1}}{2^n}$, $n \geq 1$, y comprobar que

$$S_n = \frac{1}{2} - \frac{1}{2^2} + \frac{1}{2^3} - \frac{1}{2^4} + \ldots + \frac{(-1)^{n-1}}{2^n} = \frac{1}{3}\left(1 - \frac{(-1)^n}{2^n}\right)$$

Representar conjuntamente, los diez primeros términos de las sucesiones a_n y S_n

(d) ¿Crees que alguna de las sumas de los apartados anteriores tendrá un valor finito cuando se sumen un número infinito de términos? Comprueba tu respuesta con Matlab.

Indicaciones

Sucesión aritmética:

Se define como, $a_1 = a, \quad a_n = a_{n-1} + d, \quad n \geq 2$

La suma de sus n primeros términos es,

$$a_1 + a_2 + \ldots + a_n = \frac{n(a_1 + a_n)}{2}$$

Sucesión geométrica:

Se define como, $a_1 = a, \quad a_n = r a_{n-1}, \quad n \geq 2$

La suma de sus n primeros términos es, $a_1 + a_2 + \ldots + a_n = a_1 \dfrac{1-r^n}{1-r}$

(a) Comprueba que la sucesión de los números naturales forman una sucesión aritmética. Escribe el primer término, y calcula la distancia. Finalmente escribe el término general en función de n y la suma enésima en forma de sumatorio

$a =$ $d =$ $a_n =$ $S_n =$

Escribe el código Matlab para comprobar que la suma enésima es $S_n = \frac{n(n+1)}{2}$

```
syms k n
%Apartado a
Sn1=symsum(k,1,n)
factor(Sn1) %para factorizar, visto en la práctica 2.
```

(b) Escribe el código Matlab para comprobar que la suma enésima es $S_n = \frac{n(n+1)(2n+1)}{6}$

(c) Comprueba que se trata de una sucesión geométrica. Calcula el primer término y la razón y la suma enésima en forma de sumatorio.

$a_1 =$ $r = \frac{a_n}{a_{n-1}} =$ $S_n =$

Escribe el código Matlab para comprobar que la suma enésima es

$$S_n = \frac{1}{2} - \frac{1}{2^2} + \frac{1}{2^3} - \frac{1}{2^4} + \ldots + \frac{(-1)^{n-1}}{2^n} = \frac{1}{3}\left(1 - \frac{(-1)^n}{2^n}\right)$$

Utiliza el comando plot para hacer la representación gráfica de los veinte primeros términos de las dos sucesiones, en la misma figura. Utiliza colores diferentes para distinguir los términos de ambas sucesiones.

2

Series geométricas.

(a) Comprobar si las siguientes series son geométricas, determinar su convergencia y calcular su suma.

a1) $\sum_{n=1}^{\infty} \frac{1}{2^n} = 1$ a2) $\sum_{n=1}^{\infty} \frac{4(-2)^{n+6}}{-5 \cdot 3^{(2n-3)}}$ a3) $\sum_{n=0}^{\infty} \frac{3^{n-5}}{6(-2)^{3n+2}}$ a4) $\sum_{n=0}^{\infty} \frac{1}{n!}$

(b) Justificar la convergencia y hallar la suma de la serie siguiente:

$$20+4+0.8+0.16+0.032+\ldots$$

Indicaciones

Serie geométrica: Es de la forma $\sum_{n=0}^{\infty} ar^{n-1} = a + ar + ar^2 + \ldots + ar^{n-1} + \ldots$

Esta serie es

- Convergente si $|r|<1$, siendo su suma $S=\dfrac{a}{1-r}$
- Divergente u oscilante si $r>1$

Calcular, en todos los casos, el primer término y la razón si existe.

Para las series geométricas, calcular la suma a mano y comprobar el resultado con Matlab. Para las que no sean geométricas calcular la suma únicamente con Matlab cuando sea posible.

3

Suma de series.

Calcula la suma de las siguientes series y comprueba el resultado con Matlab:

a) $\displaystyle\sum_{n=1}^{\infty} \frac{2}{n(n+1)}$ b) $\displaystyle\sum_{n=2}^{\infty} \frac{1}{n^2-1}$ c) $\displaystyle\sum_{n=1}^{\infty} \left(\frac{1}{e^n}-\frac{1}{n}\right)$

Indicaciones

Este ejercicio es el ejercicio propuesto nº6 del tema 3. Se calculará la suma a mano cuando sea posible, y se comprobarán los resultados con Matlab.

Resumen de comandos

Estos son los comandos utilizados en esta práctica que se darán por conocidos en las prácticas siguientes y que conviene retener porque se podrán preguntar en las distintas pruebas de evaluación.

- Para calcular el límite de una expresión simbólica: `limit`
- Para calcular una suma simbólica: `symsum`
- Para calcular la suma de las elementos de un vector numérico : `sum`
- Para factorizar una expresión simbólica: `factor`

Prácticas Matlab

Práctica 6 (9/11/2012)

Objetivos

- Profundizar en el concepto de suma enésima y resolver casos prácticos con ayuda de Matlab.
- Analizar la convergencia de una serie numérica con ayuda de Matlab y contrastar los resultados experimentales con los teóricos.
- Sumar series numéricas.
- Introducir las series de potencias.

Comandos de Matlab

Los comandos que se utilizarán en esta práctica ya han sido explicados en prácticas anteriores.

Ejercicios

1

Series geométricas.

(a) Comprobar si las siguientes series son geométricas, determinar su convergencia y calcular su suma.

a1) $\sum_{n=1}^{\infty}\frac{1}{2^n}=1$ a2) $\sum_{n=1}^{\infty}\frac{4(-2)^{n+6}}{-5\cdot 3^{(2n-3)}}$ a3) $\sum_{n=0}^{\infty}\frac{3^{n-5}}{6(-2)^{3n+2}}$ a4) $\sum_{n=1}^{\infty}\frac{1}{n(n+1)}$

(b) Justificar la convergencia y hallar la suma de la serie siguiente:

$$20+4+0.8+0.16+0.032+\dots$$

Indicaciones

Se debe calcular la suma a mano y se comprobarán los resultados con Matlab.

Para calcular la suma a mano en el ejercicio a4) se debe descomponer el término $a_n = \dfrac{1}{n(n+1)}$ en fracciones simples. A partir de esta expresión de deberá escribir la de la suma parcial enésima S_n y calcular luego su límite.

2

Series de potencias

a) Considera la función $f(x) = \dfrac{2}{2-x}$. ¿Cuál es su dominio? Representa la función en el intervalo [-3,3]

b) Comprueba a mano que la expresión del polinomio de Taylor de grado n en el punto 0 es $T_n(x) = 1 + \dfrac{x}{2} + \dfrac{x^2}{2^2} + \ldots + \dfrac{x^n}{2^n}$

c) Considera para los valores de x siguientes:

$$-2.25, -1.75 \; -1.25, \; -0.75, \; -0.25, \; 0.25, \; 0.75, \; 1.25, \; 1.75, \; 2.25$$

la serie numérica $\sum_{n=1}^{\infty} \dfrac{x^n}{2^n}$.

1) Obtén la expresión de la suma parcial enésima para cada x.

2) Para cada x representa la suma parcial enésima considerando distintos valores de n. ¿Observas alguna tendencia respecto al valor al que podrían converger estas sumas parciales?

3) Calcula finalmente el valor de la suma de esta serie para cada x cuando sea posible.

d) ¿En qué puntos coincide la serie $\sum_{n=1}^{\infty} \dfrac{x^n}{2^n}$ con $f(x) = \dfrac{2}{2-x}$?

Indicaciones

a) Utiliza el comando `plot` para representar la función en el intervalo [-3, 3]

b) Calcula la derivada enésima de $f(x)=\frac{2}{2-x}$ y utiliza la siguiente fórmula para calcular su polinomio de Taylor en el punto a=0:

$$T_n(x)=f(0)+\frac{f'(0)}{1!}x+\ldots+\frac{f^{(n}(0)}{n!}x^n$$

c) Introduce los valores de x en un vector y calcula la suma con el comando `symsum`.

Rellena la siguiente tabla para contestar a este apartado

x	-2.25	-1.75	-1.25	-0.75	-0.25
$S(x)=\sum_{n=1}^{\infty}\frac{x^n}{2^n}$					
$f(x)=\frac{2}{2-x}$					

x	0.25	0.75	1.25	1.75	-2.25
$S(x)=\sum_{n=1}^{\infty}\frac{x^n}{2^n}$					
$f(x)=\frac{2}{2-x}$					

Resumen de comandos

Estos son los comandos utilizados en esta práctica que se darán por conocidos en las prácticas siguientes y que conviene retener porque se podrán preguntar en las distintas pruebas de evaluación.

- Para representar una función `plot`
- Para calcular el límite de una expresión simbólica: `limit`
- Para calcular una suma simbólica: `symsum`

Prácticas Matlab

Práctica 7 (16/11/2012)

Objetivos

- Aproximar el valor de una función mediante una serie de potencias.
- Acotar el error de la aproximación de la función en un punto por una serie de potencias.

Comandos de Matlab

1.- Para calcular la integral indefinida de una expresión simbólica S respecto de la variable x.

```
int(S,x)
```

Ejemplo:

```
>> syms x
>> int(-2*x/(1+x^2)^2,x)
```

2.- Para calcular la integral definida de una expresión simbólica S respecto de la variable x en el intervalo [a,b].

```
int(S,x,a,b)
```

Ejemplo:

```
>> syms x
>> int(x*log(1+x),0,1)
```

Ejercicios

1

Aproximación mediante Series de Taylor

a) Calcula con error de una milésima el valor de la integral $\int_0^1 e^{-x^2}dx$ utilizando series de potencias centradas en el origen.

Observa que no se puede encontrar una primitiva en términos de funciones elementales pero que sí se puede obtener a partir de la serie de Taylor de

e^{-x^2} .

b) La función error, también denominada como función error de Gauss, es una función que podemos encontrar en distintos campos como: estadística, probabilidad (la campana de Gauss) y las ecuaciones en derivadas parciales. Se define como

$$erf\left(x\right)=\frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}dt$$

Esta función no puede evaluarse en términos de funciones elementales pero se puede obtener integrando la serie de Taylor de e^{-x^2}

$$erf\left(x\right)=\frac{2}{\sqrt{\pi}}\sum_{n=0}^{\infty}\frac{\left(-1\right)^n x^{2n+1}}{n!\left(2n+1\right)}$$

1. Calcula los valores que devuelve Matlab cuando se considera 0, 0.5, 1, 1.5 y 2 utilizando la función `erf(x)`.
2. Obtén después una aproximación de estos valores utilizando la serie de Taylor con un error menor que 10^{-4} .

Indicaciones

Apartado a)

Si se escribe el siguiente código Matlab para calcular la integral pedida

```
>> syms x
>> valorint= int(exp(-x^2),0,1)
```

el valor que da se expresa en términos de la función `erf` (esta función se define en el segundo apartado de este ejercicio):_

```
valorint =
(pi^(1/2)*erf(1))/2
```

Para obtener el valor numérico de la integral escribimos

```
>> format long
>> double(valorint)
ans =
     0.746824132812427
```

En este ejercicio vamos a obtener una aproximación de la integral utilizando series de potencias. El error que queremos lograr en la aproximación debe ser menor que una milésima. Los pasos que debemos dar son los siguientes:

Paso 1. Calculamos a mano el desarrollo en serie de Taylor de la función $f(x)=e^{-x^2}$ en el punto 0 y obtenemos también su campo de convergencia.

$$f(x)=e^{-x^2}=\sum_{n=0}^{\infty}\frac{(-1)^n x^{2n}}{n!} \qquad x\in\mathbb{R}$$

Paso 2: Teniendo en cuenta que el intervalo [0, 1] es un intervalo cerrado contenido en su campo de convergencia podemos integrar término a término la serie obtenida en el paso anterior. De esta manera se obtendrá el valor exacto de $\int_0^1 e^{-x^2}dx$ que será una serie numérica.

$$\int_0^1 e^{-x^2}dx=\sum_{n=0}^{\infty}\frac{(-1)^n}{n!\,(2n+1)}=S$$

Paso 3. Al ser el valor exacto de la integral una serie alternada convergente por Leibniz podemos calcular la suma aproximada con cualquier aproximación. En nuestro caso el error que se pide en la aproximación es menor que una milésima por lo que basta obtener el número n que cumpla:

$$\frac{1}{(2n+1)\,n!}<10^{-3}$$

Podemos utilizar el siguiente código Matlab

El valor de n es:

Paso 4. Para calcular finalmente el valor aproximado de $\int_0^1 e^{-x^2}dx$ debemos sumar los n primeros términos de la serie $\sum_{n=0}^{\infty}\frac{(-1)^n}{n!\,(2n+1)}$ siendo n el valor obtenido en el paso 3.

El valor aproximado de la integral es:

Apartado b)

En Matlab la función error de Gauss e escribe como `erf(x)`. Para obtener unos valores de esta función se puede escribir el siguiente código

```
>> valores=0:0.5:2;
>>format long
>>double(erf(valores))
ans =
  Columns 1 through 3
    0   0.520499877813047  0.842700792949715
  Columns 4 through 5
   0.966105146475311   0.995322265018953
```

Podemos también dibujar esta función en el intervalo [0, 2]

```
>> xv=0:0.1:2;
>> plot(xv,erf(xv))
```

En este ejercicio se pide utilizar series de Taylor para obtener los valores de la función `erf(x)`.

Paso 1. Comprobamos a mano que $erf(x)=\frac{2}{\sqrt{\pi}}\sum_{n=0}^{\infty}\frac{(-1)^n x^{2n+1}}{n!(2n+1)}$

Paso 2. Observamos que para cada x de los considerados en el apartado b.1 la serie $erf\left(x\right)=\frac{2}{\sqrt{\pi}}\sum_{n=0}^{\infty}\frac{\left(-1\right)^{n}x^{2n+1}}{n!\left(2n+1\right)}$ es una serie alternada.

En la aproximación de $erf\left(x\right)$ por la suma de los n primeros términos se cometerá un error menor que $error<\frac{x^{2n+1}}{n!\left(2n+1\right)}$. Si queremos que $\frac{x^{2n+1}}{n!\left(2n+1\right)}<10^{-4}$, basta considerar el valor de n que verifique esta desigualdad.

Realizando los cálculos con Matlab.

Paso 3. Calculamos finalmente la suma de los términos de la serie adecuados para conseguir la aproximación pedida.

Completa con los resultados obtenidos la siguiente tabla.

x	n	$erf\left(x\right)$

2

Otra función que aparece con frecuencia en tratamiento de señales es la función integral senoidal que se define como

$$si(x) = \int_0^x \frac{sen(t)}{t} dt$$

Se pide obtener $si(\pi)$ con un error menor que 10^{-6}.

Indicaciones

Repetir los pasos seguidos en el apartado b) del ejercicio anterior.

Resumen de comandos

Estos son los comandos utilizados en esta práctica que se darán por conocidos en las prácticas siguientes y que conviene retener porque se podrán preguntar en las distintas pruebas de evaluación.

- Para calcular la suma de las componentes de un vector `sum`
- Para integrar una función simbólica: `int`

Prácticas Matlab

Práctica 8 (23/11/2012)

Objetivos

- Representar gráficamente una función de dos variables.
- Representar curvas de nivel de una función de dos variables.
- Practicar con ecuaciones paramétricas de superficies.

Comandos de Matlab

1.- Para generar una malla de puntos en los cuales 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
```

2.- Para hacer gráficas tridimensionales.

```
plot3(X,Y,Z,S)
```

Dibuja las curvas o 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 dibujar curvas sobre la superficie 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)
```

3.- Para representar superficies.

`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 dibujar la superficie definida por
>> %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)
```

4.- Para hacer 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 dibujar el gráfico de malla de 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)
```

5.- Para representar 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 dibujar curvas de nivel de 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)
```

6.- Para representar 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)
```

7.- Para controlar otros parámetros de la representación

`view([x,y,z])`

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

`ginput`

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

Ejercicios

1

Representación de superficies y curvas de nivel

(a) Representa la superficie de ecuación $z = x^2 + y^2$ en el rectángulo $R = [-3,3]\times[-3,3]$. Dibuja en una misma figura cuatro gráficas probando los comandos: `surf`, `surfc`, `mesh` y `meshc`.

(b) En otra figura representa 6 curvas de nivel igualmente espaciadas utilizando los comandos `contour` y `contour3`.

(c) Representa la misma superficie utilizando las ecuaciones paramétricas,

$$x = u\cos v, \quad y = u\,\mathrm{sen}\, v, \quad z = u^2$$

para $0 \le u \le 5, \quad 0 \le v \le 2\pi$.

(d) Repite el apartado (b) utilizando la malla de puntos obtenida con las ecuaciones paramétricas.

A la vista de las gráficas obtenidas, responde a estas preguntas:

1. ¿Qué ecuaciones te parecen más adecuadas para la representación de este paraboloide? , ¿Por qué?
2. A la vista de la gráfica, define el dominio y la imagen de la función $z = x^2 + y^2$.
3. ¿Qué ecuación tienen las 6 curvas de nivel representadas?, ¿Qué curvas son?

Indicaciones

Damos resuelto este primer ejercicio para que sirva como modelo para otros similares.

(a) Representamos en primer lugar la superficie definida por su ecuación cartesiana explícita, sobre el dominio rectangular del enunciado.

Comandos Matlab:

```
[X,Y]=meshgrid(-3:.2:3,-3:0.2:3);
Z=X.^2.+Y.^2;
subplot(2,2,1); surf(X,Y,Z)
subplot(2,2,2); mesh(X,Y,Z)
subplot(2,2,3); surfc(X,Y,Z)
subplot(2,2,4); meshc(X,Y,Z)
```

(b) Utilizando la malla tridimensional de puntos obtenida en el apartado anterior dibujamos las curvas de nivel en el espacio y en el plano

Comandos Matlab:

```
figure(2)
subplot(2,1,1) ; contour3(X,Y,Z,6)
subplot(2,1,2) ; contour(X,Y,Z,6)
```

(c) Representamos ahora la superficie definida por las ecuaciones paramétricas.

Comandos Matlab:

```
u=0:.1:5;v=0:pi/50:2*pi;;
[U,V]=meshgrid(u,v);
X=U.*cos(V); Y=U.*sin(V); Z=U.^2;
subplot(2,2,1); surf(X,Y,Z)
subplot(2,2,2); mesh(X,Y,Z)
subplot(2,2,3); surfc(X,Y,Z)
subplot(2,2,4); meshc(X,Y,Z)
```

(d) Finalmente, dibujamos las curvas de nivel, utilizando la malla de puntos obtenida mediante las ecuaciones paramétricas.

Comandos Matlab:

```
figure(2)
subplot(2,1,1) ; contour3(X,Y,Z,6) ;axis square
subplot(2,1,2) ; contour(X,Y,Z,6) ;axis square
```

Escribe aquí la respuesta a las preguntas :

1.-

2.-

3.-

2

Representación de funciones de dos variables sobre un rectángulo

Representar, utilizando el comando **`surf`**, las siguientes funciones:

(a) $$z = \frac{\operatorname{sen}\sqrt{x^2+y^2}}{\sqrt{x^2+y^2}} \quad \text{en} \quad [-8,8]\times[-6,6]$$

(b) $$z = \frac{x^2-y^2}{x^2+y^2} \quad \text{en} \quad [-8,8]\times[-3,3]$$

A la vista de las gráficas contestar a las siguientes preguntas:

1. ¿Cuál es el dominio de estas funciones?, ¿y la imagen?
2. ¿Son funciones continuas sobre el rectángulo de la representación?, en caso de no serlo, ¿se pueden redefinir de forma que lo sean?

Indicaciones

Para escribir el código Matlab, utiliza como modelo la resolución del apartado a) del ejercicio anterior.

a) *Comandos Matlab:*

b) *Comandos Matlab:*

Escribe aquí la respuesta a las preguntas :

1.-

2.-

3

Representación de funciones de dos variables en paramétricas

Representar, utilizando el comando **`surf`**, las funciones del ejercicio 2, expresadas en paramétricas. Las ecuaciones paramétricas se obtendrán mediante el siguiente cambio de variables:

$$x = r\cos t\,, \quad y = r\,\mathrm{sen}\,t$$

Para la representación se tomará el dominio

$$0 \le r \le 8, \quad 0 \le t \le 2\pi$$

Indicaciones

Para escribir el código Matlab, utiliza como modelo la resolución del apartado c) del ejercicio número 1.

a) *Comandos Matlab:*

b) *Comandos Matlab:*

4

Curvas de nivel

Dada una función $z = f(x,y)$ las curvas de nivel son las curvas planas de ecuación $f(x,y) = k$, siendo k un punto del rango de f. Son, por tanto, las trazas de la gráfica de f con los planos horizontales $z = k$.

Utiliza el comando `surf` para representar las superficies, y el comando `contour` para representar las curvas de nivel, pedidas en los siguientes apartados.

(a) Representa en una matriz de gráficos 1x2 la superficie en el lado izquierdo y diez curvas de nivel de $f(x,y) = \sqrt{64 - 4x^2 + y^2}$ en la parte derecha. Tomar como dominio el rectángulo $x \in [-2,2]$, $y \in [-3,3]$.

(b) Representa en una matriz de gráficos 1x2 la superficie en el lado izquierdo y las curvas de nivel de $f(x,y) = \dfrac{\frac{5}{4} + \cos(5.4y)}{6 + 6(3x-1)^2}$ en la parte derecha, para k=0.1, 0.2 y 0.3. Tomar como dominio el rectángulo $x \in [-1,1]$, $y \in [-1,1]$.

Indicaciones

Para representar n curvas de nivel igualmente espaciadas , utilizaremos el comando

```
contour(X,Y,f(X,Y),n)
```

Tomar como referencia el apartado (b) del ejercicio número 1.

Para representar las líneas de nivel para k=0.1, 0.2 y 0.3, se sustituirá el argumento n por un vector con los valores de la constante.

Codigo Matlab:

Resumen de comandos

Estos son los comandos utilizados en esta práctica que se darán por conocidos en las prácticas siguientes y que conviene retener porque se podrán preguntar en las distintas pruebas de evaluación.

- Para generar una malla de puntos en el plano `meshgrid`
- Para representar una función de dos variables `surf, mesh`
- Para representar curvas de nivel : `contour, contour3`
- Para generar una matriz de gráficos: `subplot`

Prácticas Matlab

Práctica 8 (23/11/2012)

Objetivos

- Representar gráficamente una función de dos variables.
- Representar curvas de nivel de una función de dos variables.
- Practicar con ecuaciones paramétricas de superficies.

Comandos de Matlab

1.- Para generar una malla de puntos en los cuales 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
```

2.- Para hacer gráficas tridimensionales.

```
plot3(X,Y,Z,S)
```

Dibuja las curvas o 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 dibujar curvas sobre la superficie 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)
```

3.- Para representar superficies.

`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 dibujar la superficie definida por
>> %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)
```

4.- Para hacer 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 dibujar el gráfico de malla de 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)
```

5.- Para representar 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 dibujar curvas de nivel de 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)
```

6.- Para representar 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)
```

7.- Para controlar otros parámetros de la representación

`view([x,y,z])`

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

`ginput`

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

Ejercicios

1

Representación de superficies y curvas de nivel

(a) Representa la superficie de ecuación $z = x^2 + y^2$ en el rectángulo $R = [-3,3] \times [-3,3]$. Dibuja en una misma figura cuatro gráficas probando los comandos: `surf`, `surfc`, `mesh` y `meshc`.

(b) En otra figura representa 6 curvas de nivel igualmente espaciadas utilizando los comandos `contour` y `contour3`.

(c) Representa la misma superficie utilizando las ecuaciones paramétricas,

$$x = u\cos v, \quad y = u\,\mathrm{sen}\, v, \quad z = u^2$$

para $0 \le u \le 5, \quad 0 \le v \le 2\pi$.

(d) Repite el apartado (b) utilizando la malla de puntos obtenida con las ecuaciones paramétricas.

A la vista de las gráficas obtenidas, responde a estas preguntas:

1. ¿Qué ecuaciones te parecen más adecuadas para la representación de este paraboloide? , ¿Por qué?
2. A la vista de la gráfica, define el dominio y la imagen de la función $z = x^2 + y^2$.
3. ¿Qué ecuación tienen las 6 curvas de nivel representadas?, ¿Qué curvas son?

Indicaciones

Damos resuelto este primer ejercicio para que sirva como modelo para otros similares.

(a) Representamos en primer lugar la superficie definida por su ecuación cartesiana explícita, sobre el dominio rectangular del enunciado.

Comandos Matlab:

```
[X,Y]=meshgrid(-3:.2:3,-3:0.2:3);
Z=X.^2.+Y.^2;
subplot(2,2,1); surf(X,Y,Z)
subplot(2,2,2); mesh(X,Y,Z)
subplot(2,2,3); surfc(X,Y,Z)
subplot(2,2,4); meshc(X,Y,Z)
```

(b) Utilizando la malla tridimensional de puntos obtenida en el apartado anterior dibujamos las curvas de nivel en el espacio y en el plano

Comandos Matlab:

```
figure(2)
subplot(2,1,1) ; contour3(X,Y,Z,6)
subplot(2,1,2) ; contour(X,Y,Z,6)
```

(c) Representamos ahora la superficie definida por las ecuaciones paramétricas.

Comandos Matlab:

```
u=0:.1:5;v=0:pi/50:2*pi;;
[U,V]=meshgrid(u,v);
X=U.*cos(V); Y=U.*sin(V); Z=U.^2;
subplot(2,2,1); surf(X,Y,Z)
subplot(2,2,2); mesh(X,Y,Z)
subplot(2,2,3); surfc(X,Y,Z)
subplot(2,2,4); meshc(X,Y,Z)
```

(d) Finalmente, dibujamos las curvas de nivel, utilizando la malla de puntos obtenida mediante las ecuaciones paramétricas.

Comandos Matlab:

```
figure(2)
subplot(2,1,1) ; contour3(X,Y,Z,6) ;axis square
subplot(2,1,2) ; contour(X,Y,Z,6) ;axis square
```

Escribe aquí la respuesta a las preguntas :

1.-

2.-

3.-

2

Representación de funciones de dos variables sobre un rectángulo

Representar, utilizando el comando **`surf`**, las siguientes funciones:

(a) $z = \dfrac{\operatorname{sen}\sqrt{x^2+y^2}}{\sqrt{x^2+y^2}}$ en $[-8,8]\times[-6,6]$

(b) $z = \dfrac{x^2-y^2}{x^2+y^2}$ en $[-8,8]\times[-3,3]$

A la vista de las gráficas contestar a las siguientes preguntas:

1. ¿Cuál es el dominio de estas funciones?, ¿y la imagen?
2. ¿Son funciones continuas sobre el rectángulo de la representación?, en caso de no serlo, ¿se pueden redefinir de forma que lo sean?

Indicaciones

Para escribir el código Matlab, utiliza como modelo la resolución del apartado a) del ejercicio anterior.

a) *Comandos Matlab:*

b) *Comandos Matlab:*

Escribe aquí la respuesta a las preguntas :

1.-

2.-

3

Representación de funciones de dos variables en paramétricas

Representar, utilizando el comando **surf**, las funciones del ejercicio 2, expresadas en paramétricas. Las ecuaciones paramétricas se obtendrán mediante el siguiente cambio de variables:

$$x = r\cos t\,,\quad y = r\,\mathrm{sen}\,t$$

Para la representación se tomará el dominio

$$0 \le r \le 8,\quad 0 \le t \le 2\pi$$

Indicaciones

Para escribir el código Matlab, utiliza como modelo la resolución del apartado c) del ejercicio número 1.

a) *Comandos Matlab:*

b) *Comandos Matlab:*

4

Curvas de nivel

Dada una función $z = f(x,y)$ las curvas de nivel son las curvas planas de ecuación $f(x,y) = k$, siendo k un punto del rango de f. Son, por tanto, las trazas de la gráfica de f con los planos horizontales $z = k$.

Utiliza el comando `surf` para representar las superficies, y el comando `contour` para representar las curvas de nivel, pedidas en los siguientes apartados.

(a) Representa en una matriz de gráficos 1x2 la superficie en el lado izquierdo y diez curvas de nivel de $f(x,y) = \sqrt{64 - 4x^2 + y^2}$ en la parte derecha. Tomar como dominio el rectángulo $x \in [-2,2]$, $y \in [-3,3]$.

(b) Representa en una matriz de gráficos 1x2 la superficie en el lado izquierdo y las curvas de nivel de $f(x,y) = \dfrac{\frac{5}{4} + \cos(5.4y)}{6 + 6(3x-1)^2}$ en la parte derecha, para k=0.1, 0.2 y 0.3. Tomar como dominio el rectángulo $x \in [-1,1]$, $y \in [-1,1]$.

Indicaciones

Para representar n curvas de nivel igualmente espaciadas , utilizaremos el comando

```
contour(X,Y,f(X,Y),n)
```

Tomar como referencia el apartado (b) del ejercicio número 1.

Para representar las líneas de nivel para k=0.1, 0.2 y 0.3, se sustituirá el argumento n por un vector con los valores de la constante.

Codigo Matlab:

Resumen de comandos

Estos son los comandos utilizados en esta práctica que se darán por conocidos en las prácticas siguientes y que conviene retener porque se podrán preguntar en las distintas pruebas de evaluación.

- Para generar una malla de puntos en el plano `meshgrid`
- Para representar una función de dos variables `surf, mesh`
- Para representar curvas de nivel : `contour, contour3`
- Para generar una matriz de gráficos: `subplot`

Prácticas Matlab

Práctica 9 (30/11/2012)

Objetivos

- Determinar las derivadas parciales de una función de forma simbólica.
- Representar el campo gradiente junto con las curvas de nivel. Analizar geométricamente algunas propiedades del gradiente.

Comandos de Matlab

1.- Para dibujar vectores en el plano a partir de las coordenadas de los puntos iniciales y finales

```
quiver(pix,piy,pfx,pfy)
```

Ejemplo:

```
>> [X,Y]=meshgrid(-1:0.5:1);
>> U=2*X;V=-X+Y;
>> quiver(X,Y,U,W)
```

2.- Para calcular el gradiente de una matriz utilizando hx y hy como distancia entre los puntos en la dirección x e y respectivamente

```
gradient(Z,hx,hy)
```

Ejemplo:

```
>>[X,Y] = meshgrid(-2:.2:2, -2:.2:2);
>> Z = X .* exp(-X.^2-Y.^2);
>>[px,py] = gradient(Z,.2,.2);
```

3- Para etiquetar las curvas de nivel

```
clabel(c,h)
```

Ejemplo:

```
>> [X,Y]=meshgrid(-1:0.5:1);
>> Z=X+Y;
>>[c,h]=contour(X,Y,Z);
>>clabel(c,h)
```

Ejercicios

1

Cálculo de la derivada parcial en forma simbólica

Dada la función $f(x,y) = sen(xy) + \cos(xy^2)$ se pide calcular:

$$\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial x \partial y}, \frac{\partial f}{\partial y \partial x}$$

Recuerda que las derivadas parciales de segundo orden se definen:

$$\frac{\partial}{\partial x}\left(\frac{\partial x}{\partial x}\right) = \frac{\partial^2 z}{\partial x^2} = z''_{xx}(x,y) = f''_{xx}(x,y)$$

$$\frac{\partial}{\partial y}\left(\frac{\partial x}{\partial x}\right) = \frac{\partial^2 z}{\partial x \partial y} = z''_{xy}(x,y) = f''_{xy}(x,y)$$

$$\frac{\partial}{\partial y}\left(\frac{\partial z}{\partial y}\right) = \frac{\partial^2 z}{\partial y^2} = z''_{yy}(x,y) = f_{yy}(x,y)$$

$$\frac{\partial}{\partial x}\left(\frac{\partial x}{\partial y}\right) = \frac{\partial^2 z}{\partial y \partial x} = z''_{yx}(x,y) = f_{yx}(x,y)$$

(b) Comprueba que se verifica el teorema de Schwarz en $\mathbb{R}^2$

Indicaciones

Utiliza el comando `diff` para calcular las derivadas parciales. Por ejemplo para obtener las derivadas de primer orden las instrucciones serán:

```
syms x y
f=sin(x*y)+cos(x*y^2);
fx=diff(f,1,x)
fy=diff(f,1,y)
```

2

El vector gradiente y la curvas de nivel

Considerar la función $z = f(x,y) = -3x^2 - 4y^2$ sobre la región $D = \{(x,y) / -3 \leq x \leq 3, -3 \leq y \leq 3\}$.

(a) Calcula el vector gradiente en dicho punto.

(b) Representa la superficie S que es gráfica de la función f en una figura.

(c) Representa en otra figura la curva de nivel que pasa por el punto $P(1,2)$ y el vector gradiente en dicho punto $\vec{v} = \nabla f(1,2)$.

(d) Comprueba gráficamente que el vector gradiente, $\vec{v} = \nabla f(1,2)$, es ortogonal a la curva de nivel que pasa por el punto $P(1,2)$. Para ello representa el vector tangente a la curva de nivel en la misma figura que has utilizado en el apartado (c).

(e) Dibujar en la otra figura distintas curvas de nivel y en cada punto de la malla utilizada para representar la función los vectores tangentes y vectores perpendiculares a ellos.

Indicaciones

a) Utilizar el comando `diff` para calcular las componentes del vector gradiente en cualquier punto (x,y) y el comando `subs` para particularizar de este vector en el punto P(1, 2).

```
syms x y
a=1;
b=2;
f=-3*x^2-4*y^2;
fx=diff(f,x);
fy=diff(f,y);
gx=subs(fx,{x,y},{a,b});
gy=subs(fy,{x,y},{a,b});
```

b) Puedes utilizar el comando `surf` para representar la superficie S.

```
[X,Y]=meshgrid(-3:0.2:3,-3:0.2:3);
Z=-3*X.^2-4*Y.^2;
figure(1)
surf(X,Y,Z)
```

c) Representamos varias curvas de nivel. La correspondiente al valor $c = f(1, 2)$ será la que pasa por el punto $P(1, 2)$

```
figure(2)
c=subs(f,{x,y},{a,b});
valores=c-12:4:c+12;
hold on
```

```
cn=contour(X,Y,Z,valores)
clabel(cn)
axis equal
%Representamos el punto
plot(a,b,'o')
%Representación del vector gradiente normalizado
modulo=sqrt(gx^2+gy^2);
quiver(a,b,gx/modulo,gy/modulo)
```

d) Para representar el vector tangente

```
%Representación de un vector tangente a la curva de nivel
quiver(a,b,-gy/modulo,gx/modulo)
```

e) Puedes utilizar el comando gradient para calcular el gradiente en cada punto de la malla y el comando quiver para representar los vectores gradientes y los vectores tangentes a las curvas de nivel en los puntos de la malla utilizada para represenar la función

```
figure(3)
cn=contour(X,Y,Z,10)
clabel(cn,'manual')
axis equal
[gxt,gyt]=gradient(Z,0.3,0.3)
hold on
quiver(X,Y,gxt,gyt)
quiver(X,Y,-gyt,gxt)
```

3

El vector gradiente y la derivada direccional

Consideramos la función $z = f(x,y) = -2x^2 - 4y^2$.

(a) Calcula las derivadas direccionales de la función en el punto $P(1,2)$ en las direcciones siguientes:

a. del vector $\vec{u} = (\cos(\varphi), \operatorname{sen}(\varphi))$ para $\varphi = \dfrac{\pi}{3}$

b. del vector $\vec{v} = \nabla f(1,2)$

c. del vector $\overrightarrow{w}$ que une los puntos $P(1,2)$ con $R(3,5)$

(b) **Resultado**: Demuestra que si $z = f(x,y)$ es una función continua en un punto (a,b) que tiene derivadas parciales primeras en dicho punto, siendo al menos una de ellas continua en (a,b), la derivada direccional máxima de $f(x,y)$ en el punto (a,b) se alcanza en la dirección del gradiente.

(c) Calcula la derivada direccional máxima de $z = f(x,y) = -2x^2 - 4y^2$ en el punto $P(1,2)$.

(d) **El resultado del apartado (b) indica que situados en el punto**

(a,b) **se alcanzaría el valor máximo de z más rápidamente si en principio nos movemos en la dirección del vector gradiente de la función en dicho punto,** $\nabla f(a,b)$.

Teniendo en cuenta este resultado pretendemos alcanzar el valor máximo de la función $z = f(x,y) = -2x^2 - 4y^2$ describiendo una trayectoria que empiece en el punto $P(1,2)$ y que nos lleve siempre por la máxima pendiente de f.

Como primera aproximación a este idea realizaremos los siguientes pasos:

a. Desde el punto $P(1,2)$ nos movemos en la dirección del vector gradiente $\vec{u} = \nabla f(1,2)$ una distancia determinada para llegar al punto Q. En ese punto nos planteamos nuevamente elegir la dirección que nos proporcionaría el máximo crecimiento de f que será $\vec{v_1} = \nabla f(Q)$

b. Considerando ahora el punto de partida Q y la dirección $\vec{v_1} = \nabla f(Q)$ repetiríamos el proceso.

Nota: La trayectoria que seguiría P si se moviera de forma continua en la dirección de máximo crecimiento de la función es la curva de ecuación: $x(t) = e^{-4t},\ y(t) = 2e^{-8t}$

a) Como la función y sus derivadas parciales son continuas, la derivada direccional se puede calcular como el producto escalar del gradiente por la dirección. Para el primer caso, se tendrá:

```
phi=pi/3;
u1=cos(phi);
u2=sin(phi);
%Si en gx, gy almacenamos las coordenadas del gradiente
%en el punto
ddirec=gx*u1+gy*u2
```

b) La derivada direccional máxima se alcanza en la dirección del gradiente y su valor es su módulo.

```
ddirecMax=sqrt(gx^2+gy^2)
```

c) Demostración:

d) Podemos utilizar el siguiente código:

```
%Definimos la función
syms x y
f=-2*x^2-4*y^2;
%Dibujamos 10 curvas de nivel
[X,Y]=meshgrid(-2.5:0.2:2.5,-2.5:0.2:2.5);
Z=-2*X.^2-4*Y.^2;
contour(X,Y,Z,10);
axis equal
%Elegimos como punto de partida P y lo dibujamos
%junto con el punto (0,0)
a=1;
b=2;
hold on
plot(a, b,'o')
plot(0,0,'o')
%******************** Pasos ********************
%La función maxpend calcula el punto Q resultado de
%avanzar una distancia alfa desde el punto (a,b) siguiendo
%la dirección del gradiente de f en el punto (a,b).
[a1,b1]=maxpend(a,b,0.8)
%Desde el punto obtenido volvemos a realizar
%el mismo proceso
[a1,b1]=maxpend(a1,b1,0.5)
%******************** Trayectoria **************
%Si este proceso se hiciera de forma continua
%describiríamos la siguiente curva
t=0:0.1:10;
xt=exp(-4*t);
yt=2*exp(-8*t);
plot(xt,yt,'Color',[0.87,0.49,0],'LineWidth',2)
```

Resumen de comandos

Estos son los comandos utilizados en esta práctica que se darán por conocidos en las prácticas siguientes y que conviene retener porque se podrán preguntar en las distintas pruebas de evaluación.

- Para calcular la derivada de una función `diff`
- Para sustituir en una expresión simbólica as variables por otros valores `subs`
- Para obtener el gradiente númerico `gradient`
- Para dibujar un vector en el plano `quiver`
- Para representar una función de dos variables `surf`
- Para representar las curvas de nivel `contour`
- Para etiquetar las curvas de nivel `clabel`

Prácticas Matlab

Práctica 10 (11/1/2013)

Objetivos

- Profundizar en la comprensión del concepto de integración
- Utilizar el programa Matlab para calcular primitivas y para calcular valores aproximados y exactos de integrales definidas.

Comandos de Matlab

1.- Para calcular la integral de forma simbólica de una función f en un intervalo [a,b]

```
int(f,a,b)
```

Ejemplo:

```
>> syms x
>> int(x^2)     %integral indefinida
>> int(x^2,2,3) %integral definida
```

2.- Aproximar la integral de f mediante sumas de Riemann

```
rsums(f,a,b)
```

Ejemplo:

```
>> syms x
>> rsums(exp(-x^2),0,1)
```

Nota: Matlab posee los comandos `quad` y `trapz` que permiten realizar, respectivamente, la integración numérica de funciones utilizando el método de cuadratura adaptativa de Simpson y la regla de los trapecios.

Ejercicios

1

Cálculo de primitivas de una función

Calcular:

(a) $\int sen(ax)\cos(bx)\,dx$ (b) $\int \cos(\log x)\,dx$ (c) $\int e^{-x^2}\,dx$

Indicaciones

Apartado a). Utilizaremos el comando `int` y escribiremos:

```
syms a b x
f=sin(a*x)*cos(b*x);
integral=int(f,x);
pretty(integral)
```

Apartado b). La integral es un proceso difícil y puede suceder que Matlab no encuentre la primitiva de una función. En estos casos devuelve un mensaje del tipo *Explicit integral could not be found* como es el caso de la integral del apartado (b).

Apartado c). En este caso el valor que devuelve Matlab como primitiva de e^{-x^2} es:

(pi^(1/2)*erf(x))/2

La función `erf`, que se conoce con el nombre de *función error*, se define de la manera siguiente:

$$erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$$

Podemos representar su gráfica con Matlab escribiendo

```
vectorx=0:0.1:1;
plot(vectorx,erf(vectorx))
```

o tecleando: `ezplot('erf(x)',[0,1])`

Nota: Observa que por el Teorema Fundamental del Cálculo sabemos que

$$erf'(x) = \frac{2}{\sqrt{\pi}} e^{-x^2}$$

2

Cálculo del área limitada por dos curvas

Dadas las parejas de funciones y el intervalo que se especifica:

(a) $f(x) = 3\,\text{sen}\left(\frac{x}{4}\right) - 1,\; g(x) = 0,\; x \in [-10,17]$

(b) $f(x) = 3\,\text{sen}\left(\frac{x}{4}\right) - 1,\; g(x) = 0,\; x \in [-10+8\pi, 17+8\pi]$

(c) $f(x) = 3\,\text{sen}\left(\frac{x}{4}\right) + 13,\; g(x) = 4\cos\left(\frac{x}{3}\right) + 13,\; x \in [-10,17]$

(d) $f(x) = 3\,\text{sen}\left(\frac{x}{4}\right) - 5,\; g(x) = 4\cos\left(\frac{x}{3}\right) - 5,\; x \in [-10,17]$

(e) $f(x) = 3\,\text{sen}\left(\frac{x-2}{4}\right) - 5,\; g(x) = 4\cos\left(\frac{x-2}{3}\right) - 5,\; x \in [-10,17]$

se pide calcular el área encerrada por sus gráficas para los valores de x comprendidos en el intervalo correspondiente utilizando integrales definidas para obtener el valor exacto y sumas de Riemann para obtener un valor aproximado.

Indicaciones

Para realizar este ejercicio accederemos a la página

http://www.giematic.unican.es/integra/material-interactivo

y haremos clic sobre el enlace *Laboratorio: Aplicación. Área entre curvas* que se encuentra en el apartado *Aplicaciones*.

Una vez cargado, para cada apartado de este ejercicio realizaremos los siguientes pasos:

Paso1. Teclearemos en el campo correspondiente

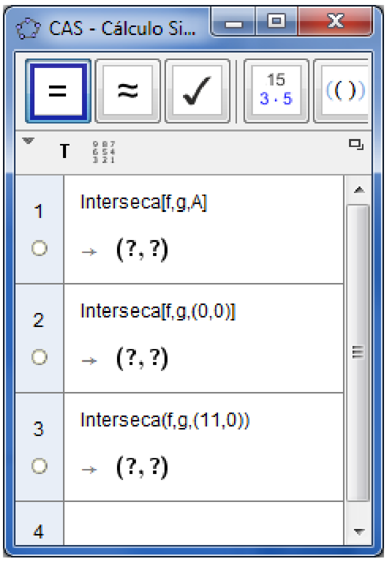
- La expresión de f y g
- Los valores a y b correspondientes al intervalo $[a,b]$

Paso 2. Observaremos:

- la gráfica de estas funciones y la de la función h que es el valor absoluto de su diferencia de f y g.
- que calculando la integral de h entre a y b obtendríamos el área entre estas dos curvas.

¿Cómo calcular esta área de forma exacta?

1. En primer lugar calculando los puntos de corte entre ambas curvas. Puedes ayudarte de la ventana **CAS** a modo de calculadora.



Tecleando en esta ventana Interseca[f,g,P] y pulsando después la tecla de salto de línea, el laboratorio devolverá las coordenadas del punto de corte de ambas curvas que esté más próximo a P.

2. Obtenidos los puntos de corte bastará calcular las integrales definidas en los subintervalos que correspondan.

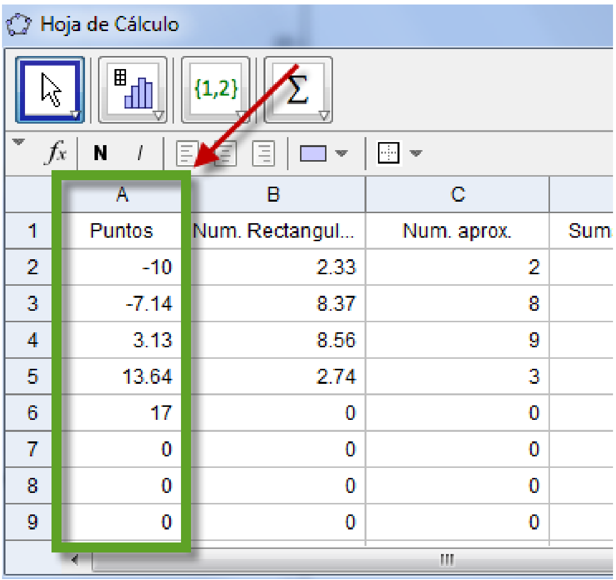
   Puedes usar Matlab para realizar los cálculos o utilizar de nuevo la ventana CAS. Para ello bastará teclear: Integral[funcion,extInf,extSup] donde funcion será la expresión de la función a integrar y extInf y extSup los extremos inferior y superior de integración.

¿Cómo calcular esta área de forma aproximada?

Utilizaremos sumas de Riemann realizando una partición del intervalo $[a, b]$ en n subintervalos de igual longitud y considerando rectángulos aproximantes. Para realizar los cálculos utilizaremos la ventana **Hoja de Cálculo**.

1. Una vez calculados los puntos de corte de las dos curvas en el intervalo $[a, b]$ deberás completar la columna A.

   En esa columna el contenido de la fila 2 será el valor a, en las siguientes filas incluirás ordenadamente la abscisa (coordenada x) de cada punto de corte utilizando una fila para cada punto. Finalmente incluirás en la última fila no nula el valor de b.



	A	B	C	
1	Puntos	Num. Rectangul...	Num. aprox.	Sum
2	-10	2.33	2	
3	-7.14	8.37	8	
4	3.13	8.56	9	
5	13.64	2.74	3	
6	17	0	0	
7	0	0	0	
8	0	0	0	
9	0	0	0	

2. Completada la columna A se actualizarán automáticamente las celdas de esta hoja:

   a. En la columna C se mostrará el número de rectángulos aproximantes que se considerarán en cada subintervalo.

   b. En la columna F se mostrarán las sumas de Riemann de f en cada subintervalo.

c. En las columnas F y H podrás comprobar:

- El valor de las áreas de los rectángulos aproximantes en cada subintervalo (columna F).
- El valor de la integral que calcularía el área comprendida entre ambas curvas en cada subintervalo (columna H).

Solución apartados:

Apartado	Área encerrada por las curvas	Área aproximada			¿Qué suma de Rieman aproxima mejor?
		n	Suma	Valor	
a)		20	Suma superior		
		20	Suma inferior		
		20	Suma centrada		
b)					¿Se obtiene el mismo resultado que en a)? ¿Por qué?
c)					
d)					¿Se obtiene el mismo resultado que en c)? ¿Por qué?

e)					¿Se obtiene el mismo resultado que en c)? ¿Por qué?
Comentarios:					

Resumen de comandos

Estos son los comandos utilizados en esta práctica que se darán por conocidos en las prácticas siguientes y que conviene retener porque se podrán preguntar en las distintas pruebas de evaluación.

- Para calcular la integral de una función `int`
- Para obtener la aproximación por sumas de Riemann `rsums`

Práctica 10: Derivada direccional. Plano tangente

Objetivos

- Mostrar la interpretación geométrica de la derivada direccional.
- Interpretar el plano tangente como la mejor aproximación lineal en las cercanías de un punto.

Comandos de Matlab

line

Dibuja una o más líneas que unen los puntos cuyas coordenadas se indiquen como argumentos. Se puede especificar el color, tipo de trazo, ...

Ejemplo:

```
line([xini, xend]', [yini,yend]',
        'color','r','LineWidth',4,'MarkerSize',12,'LineStyle','-
        ','Marker','*')
```

quiver3

Dibuja los vectores U, V, W con flechas en los puntos X, Y,Z. Las matrices X, Y, U, V deben tener el mismo tamaño.

Ejemplo:

```
[X,Y,Z]=meshgrid(-1:0.5:1);
U=2*X;V=-X+Y;W=2+0*X;
quiver3(X,Y,Z,U,V,W)
```

Ejercicios resueltos

1

Considera la función $f(x,y)=9-x^2-y^2$ y el punto P(1,1)

a) Dibuja la función en [-2,2]x[-2,2]
b) Considerar el punto (1,2) y la dirección u=cosf con f=45º. Dibujar los puntos del dominio que están en la recta anterior.
c) ¿Qué ocurre cuando evaluamos la función en los puntos de esa curva? Dibuja la curva imagen.
d) Dibujar la recta tangente a la curva anterior en el punto P(1,1,f(1,1))

Solución

(a) Código Matlab

```
[X,Y]=meshgrid(-2:.25:2);
Z=9-X.^2-Y.^2;
h1=surf(X,Y,Z);
%Dibujamos la superficie en color magenta, con transparencia 0.5
%y color de la retícula en blanco
set(h1,'FaceColor','magenta','FaceAlpha',0.5,'EdgeColor','w')

%Etiquetamos los ejes
xlabel('Eje X')
ylabel('Eje Y')
zlabel('Eje Z')
title('Gráfica de f(x,y) = 9 - x^2 - y^2')
view(150,20)
```

(b) Código Matlab

```
hold on
%Dibujamos el punto (1,1,0) y su imagen
plot3(1,1,0,'bo')
plot3(1,1,7,'bo')
%Parametrizamos la recta en el plano z=0 que pasa por (1,2,0)
%y tiene por vector director u=(cos(pi/4),sen(pi/4))
t=linspace(-6/sqrt(2),2/sqrt(2));
X1=1+t*sqrt(2)/2;
Y1=1+t*sqrt(2)/2;
Z1=0*t;
%La dibujamos con grosor 2 y color azul
line(X1,Y1,Z1,'linewidth',2,'color','blue')
```

(c) Código Matlab

```
%Dibujamos la curva C imagen de la recta anterior
Z1=9-X1.^2-Y1.^2;
line(X1,Y1,Z1,'linewidth',2,'color','blue')
```

(d) Código Matlab

```
%Dibujamos la recta tangente a C en el punto (1,1,f(1,1))
lambda=linspace(-1,1);
X3=1+lambda*sqrt(2)/2;
Y3=1+lambda*sqrt(2)/2;
Z3=7-2*lambda*sqrt(2);
line(X3,Y3,Z3,'LineWidth',2,'color','black')
```

2

Considerar la función $f(x,y) = -x^2 - y^2$ y el punto (1, 2)

a) Dibujar la función en [-1,3]x[0,4]
b) Dibujar el plano tangente en el punto P(1,2,f(1,2))
c) Dibujar un vector director del plano tangente calculado en el apartado b) contenido en el plano x=1.
d) Dibujar un vector director del plano tangente calculado en el apartado b) contenido en el plano y=2.

e) Dibujar un vector normal al plano tangente a la superficie en el punto P.

Solución

(a) Código Matlab

```
clear all
syms x y
funcion=-x^2-y^2;
a=1;
b=2;
%Generamos una malla en el dominio
[X1, Y1]=meshgrid(a+(-2:0.5:2),b+(-2:0.5:2));
Z1=subs(funcion,{x,y},{X1,Y1});
h1=surf(X1,Y1,Z1);
set(h1,'FaceColor','magenta','FaceAlpha',0.25,'EdgeColor','w')
view(-39,58)
%Etiquetamos los ejes
xlabel('Eje X')
ylabel('Eje Y')
zlabel('Eje Z')
```

(b) Código Matlab

```
%Calculamos el plano tangente
fx=diff(funcion,x);
fy=diff(funcion,y);
fab=subs(funcion,{x,y},{a,b});
fxab=subs(fx,{x,y},{a,b});
fyab=subs(fy,{x,y},{a,b});
ztangente=fab+fxab*(x-a)+fyab*(y-b);

%Generamos una malla en las proximidades del punto
hold on
[X2,Y2]=meshgrid(a+(-0.8:.2:0.8),b+(-0.8:0.2:0.8));
Z2=subs(ztangente,{x,y},{X2,Y2});
%En el caso en el que ztangente=0 se pone: Z2=0*X2;
h2=surf(X2,Y2,Z2);
set(h2,'FaceColor','white','FaceAlpha',0.5,'EdgeColor','b')
plot3(a,b,fab,'ko','LineWidth',2,'MarkerFaceColor','k','MarkerSize',5)
```

(c) Código Matlab

```
quiver3(a,b,fab,1,0,fxab,'LineWidth',3)
```

(d) Código Matlab

```
quiver3(a,b,fab,0,1,fyab,'LineWidth',3)
```

(e) Código Matlab

```
quiver3(a,b,fab,-fxab,-fyab,1,'LineWidth',2)
hold off
```

Ejercicios propuestos

1

Considerando la función $f(x,y)=9-x^2-y^2$ se pide:

(a) Dibujar la superficie S que es gráfica de f.
(b) Encontrar la ecuación de la recta que pasa por el punto (-1,1) en la dirección del vector u=(2, 1). Representar esta recta en el plano z=0.

Nota: No olvidar normalizar el vector u.

(c) Dibujar la curva en la superficie S que es imagen por f de los puntos de la recta del apartado (b).
(d) Representar la recta tangente a la superficie S en el punto $\left(-1,1,f(1,1)\right)$ en la dirección del vector u.

2

De la misma forma que se utiliza la recta tangente para aproximar el valor de una función $y=f(x)$ en las proximidades de un punto a

$$f(x)\approx f(a)+f'(a)(x-a) \qquad si \;\; x\to a$$

se puede utilizar el plano tangente para aproximar el valor de una función $y=f(x,y)$ en las proximidades de un punto (a,b)

$$f(x,y)\approx f(a,b)+\frac{\partial f}{\partial x}(a,b)\cdot(x-a)+\frac{\partial f}{\partial y}(a,b)\cdot(y-a) \quad si \;\; (x,y)\to(a,b)$$

o también

$$f(a+\Delta x,b+\Delta y)\approx f(a,b)+\frac{\partial f}{\partial x}(a,b)\cdot\Delta x+\frac{\partial f}{\partial y}(a,b)\cdot\Delta y \;\; si \;\; (\Delta x,\Delta x)\to(0,0)$$

Calcular, utilizando la diferencial como la aproximación que proporciona el plano tangente, una estimación de $\sqrt{3{,}98^2+3{,}01^2}$

Práctica 11

Objetivos

- Calcular integrales definidas de forma aproximada, utilizando sumas de Riemann.
- Profundizar en la comprensión del concepto de integración.

Comandos de Matlab

int

Calcula de manera simbólica la integral indefinida de una función.

Ejemplo:

```
syms x
int(x^2/(x^6-8))
```

rsums

Aproxima la integral de f mediante sumas de Riemann y realiza una representación gráfica de los rectángulos.

Ejemplo:

```
syms x
rsums exp(-x^2)
```

Definiciones

Definición (*Partición*).- Dados dos números reales tales que $a < b$, recibe el nombre de partición del intervalo cerrado $[a,b]$ todo conjunto finito de puntos de $[a,b]$ de la forma:

$$P = \{x_0, x_1, x_2, \ldots, x_n \mid a = x_0 < x_1 < x_2 < \ldots < x_n = b\}$$

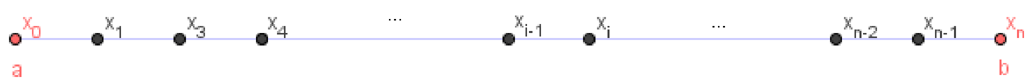


Figura 1.- Ejemplo de una partición de [a, b]

Si todos los puntos de la partición son equidistantes, se habla de *partición regular*. En este caso se cumple $\|P\| = \Delta x$ donde

$$\|P\| = \max_{1 \le i \le n} |x_i - x_{i-1}|$$

se llama *norma de la partición*.

Definición (*Integral definida*).- Dada una función $y = f(x)$ acotada en un intervalo cerrado $[a,b]$, se dice que es integrable en este intervalo $[a,b]$ si para cualquier partición P, existe el límite

$$\lim_{\substack{\|P\| \to 0 \\ (n \to \infty)}} \left(\sum_{i=1}^{n} f(c_i) \Delta x_i \right) \quad \text{con} \quad c_i \in [x_{i-1}, x_i]$$

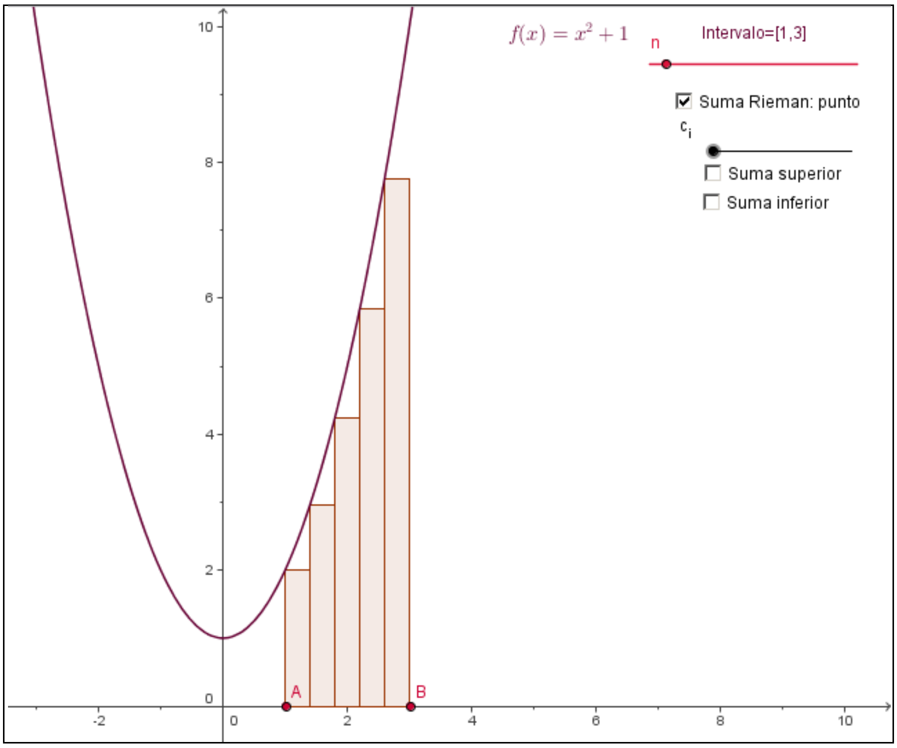
En este caso, el valor del límite recibe el nombre de *integral definida o integral de Riemann* de f sobre $[a,b]$ y se denota por

$$\int_a^b f(x)dx$$

En esta expresión, los números a y b se llaman respectivamente límite inferior y límite superior de integración.

1

1. Accede a la página http://www.giematic.com/integralDef/
2. Haz clic sobre el enlace **Material Interactivo**
3. Haz clic sobre el enlace **Laboratorios: Representación y Cálculo** del apartado **Sumas de Rieman**
4. Comprueba para diferentes valores de **n** (número de términos de la partición) y distintos valores posibles de $c_i \in [x_{i-1}, x_i]$ la suma de las áreas de los rectángulos cuya base es Δx_i y altura $f(c_i)$.
5. ¿A qué valor convergerá $\sum_{i=1}^{n} f(c_i) \Delta x_i$ si n tiende a infinito? En el caso de esta función que es positiva en el intervalo $[-1,1]$, ¿qué representará el límite obtenido?



2

Considerar la función $f(x) = x^2 + 1$ en el intervalo $[1,3]$.

a) Representar gráficamente $f(x)$ en el intervalo $[1,3]$ y destacar sobre la gráfica la región del plano cuyo área viene dado por $\int_1^3 (x^2+1)dx$.

b) Aproximar el área anterior mediante sumas de Riemann, utilizando *n* rectángulos de la misma base y de altura el valor de *f* en el extremo izquierdo de cada uno de ellos. Tomar los siguientes valores de *n*:

b1) $n = 10$ b2) $n = 20$

c) Obtener una fórmula general que proporcione una estimación del área tomando *n* rectángulos como los anteriores.

SUMAS ENÉSIMAS FRECUENTES:

$$\sum_{k=1}^{n} 1 = n$$

$$\sum_{k=1}^{n} k = 1+2+3+\ldots+n = \frac{n(n+1)}{2}$$

$$\sum_{k=1}^{n} k^2 = 1^2+2^2+3^2+\ldots+n^2 = \frac{n(n+1)(2n+1)}{6}$$

$$\sum_{k=1}^{n} k^3 = 1^3+2^3+3^3+\ldots+n^3 = \frac{n^2(n+1)^2}{4}$$

d) Calcular el valor exacto del área como límite de la expresión anterior para *n* tendiendo a infinito.

e) Calcular el error cometido en las aproximaciones calculadas en el apartado b).

f) Demuestra que si se considera la función $F(x)=\frac{x^3}{3}+x$ (función que cumple que $F'(x)=f(x)$, es decir que su derivada es la función integrando) se verifica:

$$\int_a^b f(x)\,dx = F(b)-F(a)$$

Nota: Una función $F(x)$ es primitiva de otra función $f(x)$ si cumple $F'(x)=f(x)$. Se puede demostrar que si $F(x)$ es primitiva de la función $f(x)$ cualquier otra primitiva es de la forma $F(x)+C$ siendo C una constante.

Solución

a) *Comandos Matlab*

```
x=1:.05:3;
y=x.^2+1;
plot(x,y,'r','LineWidth',2)
hold on
area(x,y)
```

b) Observa que la suma de Riemann será

$$\int_0^1 \left(x^2+1\right)dx \approx \sum_{i=1}^{10} f(c_i)\Delta x = \left\{ \begin{array}{l} \Delta x = \dfrac{2}{10} = \dfrac{1}{5} \\ c_i = 1+\left(i-1\right)\Delta x = 1+\left(i-1\right)\dfrac{1}{5} \end{array} \right\} = \frac{1}{5}\sum_{i=1}^{10}\left[\left(1+\frac{\left(i-1\right)}{5}\right)^2+1\right]$$

Comandos matlab

```
%De forma simbólica
syms k
S=symsum((1/5)*((1+(k-1)/5)^2+1),k,1,10);
double(S)
%De forma numérica
m=1:10;
am=(1/5)*((1+(m-1)/5).^2+1);
S=sum(am)
```

c) Considerando ahora una partición de n subintervalos, la suma de Riemann será

$$\int_0^1 \left(x^2+1\right)dx \approx \sum_{i=1}^{n} f(c_i)\Delta x = \left\{ \begin{array}{l} \Delta x = \dfrac{2}{n} \\ c_i = 1+\left(i-1\right)\Delta x = 1+\left(i-1\right)\dfrac{2}{n} \end{array} \right\} = \frac{2}{n}\sum_{i=1}^{n}\left[\left(1+\frac{2\left(i-1\right)}{n}\right)^2+1\right]$$

Comandos matlab

```
%Cálculo del área de los n rectángulos
syms k n
SumaN=symsum((2/n)*((1+(k-1)*2/n)^2+1),k,1,n)
pretty(simplify(SumaN))
```

d) El área de la región del plano por debajo de la gráfica de la función y delimitada por el eje de abscisas y las rectas $x=1$ y $x=3$ será:

$$\sum_{i=1}^{n} f(c_i)\Delta x = \frac{2}{n}\sum_{i=1}^{n}\left[\left(1+\frac{2\left(i-1\right)}{n}\right)^2+1\right] \quad \underset{n\to\infty}{\rightarrow} \quad \int_1^3 \left(x^2+1\right)dx$$

Comandos matlab

```
%Cálculo del límite de la suma de Riemann
Areaf=limit(SumaN,n,inf);
```

d) *Comandos Matlab*

```
%Cálculo del área menos el área aproximada
term=[10,20]
error=Areaf-subs(SumaN,n,term)
double(error)
```

f) *Comando Matlab*

```
%Cálculo de la integral en un intervalo [a, b]
syms k n a b
SumaN=symsum(((b-a)/n)*((a+(k-1)*(b-a)/n)^2+1),k,1,n);
pretty(simplify(SumaN))
Areaf =limit(SumaN,n,inf);
pretty(expand(areaF))
```

3

Repetir el ejercicio anterior tomando como altura de cada rectángulo

a) el valor de $f(x)$ en el extremo derecho de cada uno de ellos.

b) el valor de $f(x)$ en el punto medio de cada uno de ellos.

Ejercicios propuestos

1

Repetir los apartados a), b) y c) de los ejercicios 2 y 3 considerando la función $f(x)=\frac{senx}{x}$ y el intervalo $[0,\pi/2]$

2

La función definida de la forma $f(x)=ae^{\frac{-(x-b)^2}{2c^2}}$ se denomina función gaussiana y su gráfica tiene forma de campana. Para ciertos valores a,b,c esta función es la función de densidad de una variable aleatoria normal. Calcular el área bajo la curva $f(x)=\frac{1}{\sqrt{2\pi}}e^{\frac{-x^2}{2}}$ en el intervalo $[-1,1]$.