CÁLCULO NUMÉRICO Y COMPUTACIÓN.

GUÍA DE ESTUDIO.

Las Guías de Estudio 2016 tienen a finalidad de brindar una orientación de la secuencia y profundidad de los distintos temas. Están ordenadas según el cronograma a seguir en el presente año lectivo 2016. La complejidad de los ejercicios es creciente. Cada Guía de Estudio tiene indicado al principio los objetivos de la misma, una síntesis de la teoría necesaria (en algunas no), la bibliografía recomendada; y termina con ejercicios integradores.
La secuencia lógica de los ejercicios es desarrollar manualmente los primeros ejercicios que siempre son simples. Cuando se entiende y conoce el algoritmo del método, se pasa a ejercicios para desarrollar códigos propios en MATLAB.
Finalmente, los ejercicios integradores suelen necesitar de conocer más de una unidad temática de la materia; o más de una guía de estudio para poder resolverlos.

Los temas desarrollados son los siguientes:

Algoritmia. Conceptos Básicos
Solución Numérica de Raíces de Ecuaciones No Lineales
Sistemas de Ecuaciones Lineales
Valores y Vectores Propios
Interpolación y Aproximación de funciones discretas
Integración Numérica
Derivación Numérica, con aplicación a la solución de Ecuaciones Diferenciales con Valores de Contorno
Solución Numérica de Ecuaciones Diferenciales Ordinarias con Valores Iniciales

El ordenamiento elegido está según el cronograma del año 2016.

Guía de Estudio para Algoritmos

Contenido

Guía de Estudio para Algoritmos 1
Bibliografía recomendada: 1
Objetivo: 1
Estructura Secuencial 2
Ejercicios de Estructuras Secuenciales 3
Estructuras de Decisión 4
Ejercicios sobre Estructura de Decisión 4
Estructura Variar 5
Ejercicios sobre Estructura Variar 6
Estructura Iterativa 7
Ejercicios sobre Estructura Iterativa 7
Arreglos 8
Ejercicios de Arreglos 8
Subprogramas 9
Ejercicios de subprogramas 10
Algoritmos sobre Métodos de Cálculo Numérico 10
Ejercicios Integradores 11

Bibliografía recomendada:

Apuntes de Algoritmos de la Càtedra.
Capítulo 2 de S. Chapra, R. Canale; *Métodos Numéricos para Ingenieros*; Mc Graw Hill ; 1999
J. Mathews, J. Fink (2000). *Métodos Numéricos con Matlab.* Prentice Hall.
S. Nakamura (1997). *Análisis Numérico y Visualización Gráfica con MATLAB.* Prentice Hall
S. Nakamura (1992). *Métodos Numéricos Aplicados con Software.* Prentice Hall

Objetivo:

Se pretende introducir las estructuras algorítmicas básicas mediante ejercicios simples con el uso de la sintaxis de MATLAB.

Estructura Secuencial

Se trata de un conjunto de “ordenes” expresadas una en cada línea y que se ejecutan de a una por vez. Hasta que no se ha completado la ejecución de una orden, no se ejecuta la siguiente.
Se puede expresar mediante un Pseudocódigo, o bien en un Lenguaje determinado como por ejemplo MATLAB.
Se considera el siguiente ejemplo:

Se busca desarrollar un algoritmo para leer la base de un triángulo, la altura y calcular el área; entregando el resultado.

El algoritmo siempre debe tener un nombre, un comienzo y un final. Debe ser un número finitos de ordenes, o líneas.

Pseudocódigo

```
Programa area_triang
    Reales (ba; h; A)

    Escribir “Ingrese la base del triángulo”
    Leer ba
    Escribir “Ingrese la altura del triángulo”
    Leer h

    A=ba*h

    Escribir “El área del triangulo es: ”, A
Fin Programa
```

MATLAB

Se genera el siguiente archivo “area_triang.m” que pasará a ser un “comando” de MATLAB

```
function area_triang
  % Reales (ba; h; A)                    %no necesita de la declaración de variables
  disp('Ingrese base del triangulo:');   %muestra por pantalla
  ba = input('');                        %Ingresa desde teclado la base del triangulo
  disp('Ingrese altura del triangulo:');
  h = input('');

  A = ba*h/2;                            %Calcula el área con datos anteriores
  disp('el Area es:'); disp(A);          %muestra por pantalla el resultado obtenido
end
```

El nuevo comando de MATLAB generado por el archivo “area_triang.m” se puede ejecutar desde la interface del Editor de Texto de MATLAB o desde la línea de comandos del Command Window.

En lugar de ingresar los datos desde el teclado, es posible definirlos en las líneas del programa. Así es posible hacer el siguiente archivo “area_triang_01.m” que pasará a ser un “comando” de MATLAB

```
function area_triang_01
  % Reales (ba; h; A)            %no necesita de la declaración de variables
  ba = 3;                        %Se define la base del triangulo igual a 3
  h = 6; ;                       %Se define la altura del triangulo igual a 6

  A = ba*h/2;                    %Calcula el área con datos anteriores
  disp('el Area es:'); disp(A);  %muestra por pantalla el resultado obtenido
end
```

El inconveniente de esta variante es que cada vez que se pretende cambiar los datos, se debe grabar el archivo “area_triang_01.m”, con el riesgo de alterar en forma incorrecta alguna de sus líneas.

Una alternativa es separar en archivos (comandos) distintos los datos y las líneas que no deben cambiar nunca del algoritmo. Así se deben hacer dos archivos: "area_triang_02.m" y "area_triang_datos.m"
En el archivo "area_triang_02.m" se tiene lo que no debe cambiar nunca

```
function area_triang_02
    area_triang_datos               %es un comando que "hace" todo lo que tenga
                                    %el archivo "area_triang_datos.m"
                                    %en este caso es la asignación de datos
    A = ba*h/2;
    disp('el Area es:'); disp(A);
end
```

En el archivo "area_triang_datos.m" se tienen los datos que cambian para cada caso de interés

```
      % datos de  base y altura
      ba= 12
      h= 100
```

Ejercicios de Estructuras Secuenciales

1. Desarrolle un algoritmo para leer el valor de los catetos de un triángulo rectángulo y calcular la hipotenusa. Imprimir el resultado.
2. Desarrollar un programa para que una vez ingresado el valor de temperatura en grados Celsius, entregue su valor equivalente en grados Farenheit (F=(9/5)*(ºC)+32)
3. Elaborar un algoritmo que lea dos puntos del plano X,Y, con coordenadas (X1,Y1) y (X2,Y2) y calcule el punto medio de ambos. Imprimir el resultado
4. Elaborar un algoritmo que lea dos puntos del plano X,Y, con coordenadas (X1,Y1) y (X2,Y2) y calcule la pendiente de la recta que pasa por dichos puntos. Imprimir el resultado.
5. Elaborar un algoritmo que lea dos puntos del plano X,Y, con coordenadas (X1,Y1) y (X2,Y2) y calcule el valor de la abscisa Xr en la cual la recta que pasa por dichos puntos, intersecta al eje de las abscisas X. Imprimir el resultado Xr y el valor que toma la recta en Xr.

Estructuras de Decisión

Es una estructura que permite bifurcar una línea de proceso, o de pensamiento. Analiza una condición lógica y según sea el resultado del análisis verdadero o falso se realiza uno u otro proceso materializados por secuencias de líneas (comandos u órdenes) distintas.

Se considera el siguiente ejemplo:

Se busca desarrollar un algoritmo para calcular las raíces de la ecuación de segundo grado ax2 +bx +c =0 y las entregue.

Pseudocódigo

Programa raíces_seg
  Reales (a; b; c; Discrim; r1;r2)
  Escribir “Ingrese los coeficientes a, b, c”
  Leer a b c
  Discrim=(b^2-4*a*c)

  If (Discrim >0) **then**
    r1=(-b+ Discrim)^0.5/(2*a)
    r2=(-b- Discrim) ^0.5/(2*a)
    Escribir “Las raices son:”, r1,r2
  Else
    Escribir “Las raices son complejas conjugadas”
  Endif
Fin Programa

MATLAB

Se genera el siguiente archivo “area_triang.m” que pasará a ser un “comando” de MATLAB

```
function raices_seg
  % Reales (a; b; c; Discrim; r1;r2)      %no necesita de la declaración de variables
  disp('Ingrese Coeficiente a:');         %muestra por pantalla
  a = input('');                          %Ingresa desde teclado el Coef a
  disp('Ingrese Coeficiente b:');         %muestra por pantalla
  b = input('');                          %Ingresa desde teclado el Coef b
  disp('Ingrese Coeficiente c:');         %muestra por pantalla
  c = input('');                          %Ingresa desde teclado el Coef c
  Discrim=( b^2-4*a*c)                    %Calcula el Discriminante
  if (Discrim > 0)
        r1=(-b+ Discrim^0.5)/(2*a)
        r2=(-b- Discrim^0.5)/(2*a)
        disp( 'Las raíces son:');disp(r1);disp(r2)
  else
        disp( 'Las raíces son complejas conjugadas:')
  end
end
```

Ejercicios sobre Estructura de Decisión

1. Elaborar un algoritmo que lea dos puntos del plano X,Y, con coordenadas (X1,Y1) y (X2,Y2) y seleccione el punto de mayor ordenada. Imprimir el resultado.
2. Elaborar un algoritmo que lea cuatro datos y determine cual es el menor. Imprima el resultado
3. Las siguientes desigualdades definen un área en el plano x-y. Desarrolle un algoritmo para que determine si un punto de coordenadas dadas (X1, Y1), se encuentra o no dentro del área definida.

$$Y < 3$$
$$X+Y > 1$$
$$Y < 2X + 1$$
$$Y > X$$

4. Elaborar un algoritmo que lea dos puntos del plano X,Y, con coordenadas (X1,Y1) y (X2,Y2) y calcule el valor medio de X1 y X2, sólo si las ordenadas Y1 e Y2 tienen signos opuestos.
5. Elaborar un algoritmo que lea dos puntos del plano X,Y, con coordenadas (X1,Y1) y (X2,Y2) y calcule el valor medio de X1 y X2, sólo si las ordenadas Y1 e Y2 tienen signos opuestos. De lo contrario que imprima el mensaje "ordenadas de igual signo".

Estructura Variar

Es una estructura que permite realizar una serie de comandos un número finito y pre establecido de veces, que se conoce a priori.

Se considera el siguiente ejemplo:

Desarrollar un algoritmo que lea e imprima las componentes de un vector de dimensión 4.

Pseudocódigo

```
Programa vector_leer
    Reales (vec(4))
    Enteros N
    DOFOR j=1 TO 4
        Escribir "Ingrese componente"
        Leer vec(j)
    ENDDO

    DOFOR j=1 TO 4
        Escribir vec(j)
    ENDDO
Fin Programa
```

MATLAB

Se genera el siguiente archivo "vec_leer.m" que pasará a ser un "comando" de MATLAB

```
function vec_leer
% Real (vec(4))                    %No es necesario la declaración
% Integer N
%
disp('Lectura de las componentes')
N=4;
for j=1:N                         % Varia j desde 1 hasta N
    disp('Ingrese la componente')     % escribe
    vec(j)= input(' ');               % lee desde teclado
end                               % fin del bloque variar
%
disp('Escritura de las componentes')
for j=1:N
    disp('Componente ingresada'),disp(vec(j))
end
```

Se considera el siguiente ejemplo:

Una estación meteorológica recibe datos de temperatura en cada hora del día. Desarrollar un algoritmo que reciba dichos datos, y al final del día calcule el promedio e imprima el resultado.

```
function vec_valor_medio
% Lee y calcula el valor medio
% Real (vec(24), sum, vm)              %No es necesario la declaración
% Integer N
disp('Lectura de las componentes')
N=24;                                % Asigna en N las 24 horas de un día
for j=1:N                            % Varia j desde 1 hasta N
    disp('Ingrese la componente')        % escribe
    vec(j)= input(' ');                  % lee desde teclado la temperatura
end                                  % fin del bloque variar
sum=0;                         % Hace la suma de las componentes
for j=1:N                        % Varia j desde 1 hasta N
    sum = sum + vec(j);               % suma
end                              % fin del bloque variar

vm = sum/N;                      % Calcula el promedio
disp ('El promedio es'), vm

end
```

Ejercicios sobre Estructura Variar

1. Una estación meteorológica recibe datos de temperatura en cada hora del día. Desarrollar un algoritmo que reciba dichos datos, y cada vez que los recibe calcule el valor medio e imprima los resultados.
2. Una estación meteorológica recibe datos de temperatura en cada hora del día. Desarrollar un algoritmo que reciba dichos datos, y al final del día encuentre e imprima el valor mínimo y máximo.
3. Una estación meteorológica recibe datos de temperatura en cada hora del día, durante cada mes. Desarrollar un algoritmo que reciba dichos datos, y al final del día calcule el valor medio diario y del mes hasta ese momento e imprima los resultados.

Estructura Iterativa

Es una estructura que permite realizar una serie de comandos un número finito de veces, aunque NO está pre establecido a priori la cantidad de veces que se debe realizar. Se utiliza el comando while (Mientras). Es importante destacar que la estructura while (mientras) tiene las siguientes caraceristicas:

- La expresión entre paréntesis del while debe tener un resultado lógico (verdadero o falso).
- Las variables que participan de ella deben estar asignadas antes de la expresión while.
- Las variables que participan de ella deben modificarse dentro del bloque de comandos del while.

Se considera el siguiente ejemplo:

Escriba un algoritmo que lea dos puntos (X1,Y1) (X2, Y2), calcule el punto medio (XmYm) y repita la operación hasta que el punto medio tenga ordenada Ym nula. Escriba la cantidad de iteraciones realizadas y los puntos cuyo Ym es nula.

```
function iterar_ym_nulo
% Real x1,x2,y1,y2,xm,ym,tol
tol=0.0001;                       % Valor debajo del cual se considera CERO
ym=1;
iter=0;                           % contador de iteraciones
while (abs(ym)> tol)              % Bloque MIENTRAS
  iter = iter + 1;                      % acumula el número de iteraciones
                                        % Bloque ingreso de datos
  disp('Ingrese x1')
  x1=input(' ');
  disp('Ingrese y1')
  y1=input(' ');
  disp('Ingrese x2')
  x2=input(' ');
  disp('Ingrese y2')
  y2=input(' ');
                                       % Bloque de cálculo del Punto Medio
  xm=(x1+x2)/2;
  ym=(y1+y2)/2;
  disp('El valor de ym es'), ym
end                              % FIN del Bloque MIENTRAS

                                 % Entrega de resultados
disp ('El valor de ym es prácticamente nulo')
disp ('El número de iteraciones fue'), iter
disp ('El primer punto es'), x1, y1
disp ('El segundo punto es'),x2, y2
end
```

Ejercicios sobre Estructura Iterativa

1. Desarrolle un algoritmo para leer números reales, sumarlos y detenerse sólo cuando lea un cero. Imprima los resultados.
2. Escriba un algoritmo que mientras la función $f(x)=(3x^2-12)$ sea distinto de cero o el número de iteraciones sea menor que 100, lea un valor de abscisa x0, calcule la función f(x0) en ese valor x0, y sólo entregue un valor de x0 tal que la función f(x) es cero.
3. Los términos de una serie se generan con $(1/2)^k*x^k$, con k>=1. Desarrollo un algoritmo que dados los valores x, k; genere los elementos de la serie, los sume, e imprima los resultados y se detenga cuando la suma haya convergido con cierta tolerancia elegida.

Arreglos

Son variables que tienen un nombre común, pero la posibilidad de multiples asignaciones en las llamadas componentes. Típicamente vectores y matrices. Los primeros tienen la necesidad de un indicador de posición para el que se utiliza una variable entera. Mientras que las matrices necesitan dos indicadores de posición, uno para las filas y otro para las columnas, cuyas variaciones en general son independientes entre sí.

Ejercicios de Arreglos

1. Desarrollar un algoritmo que lea un vector, obtenga su norma cuadrática y la entregue.
2. Desarrollar un algoritmo que lea un vector e imprima el versor unitario que indica la dirección del vector dato.
3. Desarrollar un algoritmo que lea un vector, obtenga la norma infinito (valor absoluto máximo) del vector y la entregue.
4. Desarrollar un algoritmo que lea un vector e imprima otro vector que sea paralelo al vector dato pero con norma infinito uno. Es decir, el vector normalizado según la norma infinito.
5. Desarrollar un algoritmo que lea dos vectores, realice su diferencia e imprima los tres vectores. Asumir que el orden de los vectores permite realizar la diferencia.
6. Desarrollar un algoritmo que lea dos vectores, realice su diferencia almacenando el vector diferencia en el vector "resta" e imprima los resultados. Asumir que el orden de los vectores permite realizar la diferencia.
7. Desarrollar un algoritmo que lea dos vectores de 100 componentes, y obtenga el modulo (la norma cuadrática) del vector diferencia de los vectores leídos. Sólo si dicha norma es no nula, debe escribir el nuevo vector diferencia normalizado con dicha norma, de lo contrario debe escribir que la norma del vector diferencia es nula
8. Desarrollar un algoritmo que lea un vector de 100 componentes, obtenga la norma infinito del vector (valor absoluto máximo de sus componentes). Si dicha norma es no nula, debe escribirla y también escribir un nuevo vector normalizado del vector leído con dicha norma infinito.
9. Desarrollar un algoritmo que lea dos vectores de 100 componentes, y obtenga la norma infinito del vector diferencia de los vectores leídos (valor absoluto máximo de sus componentes). Sólo si dicha norma es no nula, debe escribir el nuevo vector normalizado del vector diferencia con dicha norma infinito, de lo contrario debe escribir que la norma del vector diferencia es nula.
10. Desarrollar un algoritmo que lea e imprima las componentes de una matriz.
11. Desarrollar un algoritmo que lea un vector y una matriz, que realice el producto entre ambos e imprima el nuevo vector. Asumir que el orden del vector y la matriz son tales que es posible el producto.
12. Desarrollar un algoritmo que lea una matriz A(i,j) de orden N y genere otra matriz Triangular Inferior TS(i,j) con la siguiente característica:
    a) cuando j es mayor que i, los elementos TS(i,j) deben ser 0;
    b) cuando i es mayor a j los elementos TS(i,j) son iguales a A(i,j)/A(i,i);
    c) los elementos de la diagonal de T(i,j) son iguales a 1
13. Desarrollar un algoritmo que lea una matriz A(i,j) de orden N y genere otra matriz Triangular Superior TS(i,j) con la siguiente característica: elementos TS(i,j)=0 cuando i es mayor que j; y con TS(i,j)=A(i,j)/A(i,i) si j es mayor o igual a i.

14. Desarrollar un algoritmo que lea una matriz A(i,j) de orden N. Luego, debe almacenar la "norma infinito" de cada fila de la matriz A en un vector denominado NormaFila. Seguidamente, debe calcular la norma infinito del vector NormaFila. Finalmente, sólo si este último resultado es cero, debe entregar el mensaje "Matriz Nula"; de otra forma, debe entregar el resultado no nulo obtenido precedido del mensaje "la norma es".Recordar que la "norma infinito" es la mayor de las componentes tomadas en valor absoluto.
15. Desarrollar un algoritmo que lea una matriz A(i,j) de orden N y un vector X(j) de orden N, y en un nuevo vector Y(i) almacene el resultado del producto de la Matriz A por el vector X. Además debe imprimir el vector resultante.
16. Desarrolle un algoritmo que lea dos matrices y almacene en una tercera la matriz el producto de las matrices datos. Asuma que el orden de las matrices es tal que es posible realizar la operación.

Subprogramas

Es un conjunto de órdenes que constituyen todo un proceso, y que se identifican con un nombre particular. Puede tener o no pasaje de argumentos; y éstos pueden ser uno o más. Los argumentos pueden ser de entrada o de salida.
Según sea el caso será la sintaxis.
Se considera el siguiente ejemplo:

Escriba un algoritmo que lea dos puntos (X1,Y1) (X2, Y2), calcule el punto medio (XmYm) y repita la operación hasta que el punto medio tenga ordenada Ym nula. Escriba la cantidad de iteraciones realizadas y los puntos cuyo Ym es nula.

En sintaxis de **MATLAB** se puede escribir el algoritmo en la siguiente forma:

```
function iterar_ym_nulo_01
% Real x1,x2,y1,y2,xm,ym,tol
tol=0.0001;                      % Valor debajo del cual se considera CERO
ym=1;
iter=0;                          % contador de iteraciones
while (abs(ym)> tol)  % Bloque MIENTRAS
  iter = iter + 1;            % acumula el número de iteraciones
  % Bloque ingreso de datos con un subprograma o function
  [x1, y1, x2, y2]= iterar_ym_nulo_01_datos;
                      % sin pasaje de argumentos de entrada
                      % con multiples argumentos de salida
  % Bloque de calculo del Punto Medio con un subprograma o function
   ym = iterar_ym_nulo_01_medio(x1, y1, x2, y2);
                      % con pasaje de argumentos de entrada
                      % con un único argumento de salida
  % Bloque de salida de resultado parcial
   disp('El valor de ym es'), ym
end                   % FIN del Bloque MIENTRAS
%  Bloque de Entrega de resultados finales
disp ('El valor de ym es practicamente nulo')
disp ('El número de itearciones fue'), iter
disp ('El primer punto es'), x1, y1
disp ('El segundo punto es'),x2, y2
end
```

```
% Este es un subprograma o function
% puede estar en este archivo
% o en otro archivo que debe llamarse "iterar_ym_nulo_01_datos.m"
function [a, b, c, d]=iterar_ym_nulo_01_datos
  disp('Ingrese x1')
  a=input(' ');
  disp('Ingrese y1')
  b=input(' ');
  disp('Ingrese x2')
  c=input(' ');
  disp('Ingrese y2')
  d=input(' ')
end

% Este es un subprograma o function
% puede estar en este archivo
% o en otro archivo que debe llamarse "iterar_ym_nulo_01_medio.m"
function zm = iterar_ym_nulo_01_medio(a, b, c , d)
  xx=(a+c)/2;
  zm=(b+d)/2;
end
```

En forma de **Pseudocódigo** es posible sintetizar el algoritmo en la siguiente forma:

`function iterar_ym_nulo_01`
Definición de datos
Mientras (abs(ym)> tol)
    Contador de iteraciones
    Bloque de ingreso de datos con un subprograma
    Blouqe de cálculo del Punto Medio con un subprograma
    Bloque de salidas de resultados parciales
Fin del Mientras
Salida de resultados finales
Fin de `iterar_ym_nulo_01`

Ejercicios de subprogramas

1. Desarrolle el algoritmo de un subprograma que lea un vector y entregue su norma cuadrática
2. Desarrolle el algoritmo de un subprograma que lea un vector y entregue el versor unitario que indica la dirección del vector dato.
3. Desarrolle un subprograma que dados un vector y una matriz entregue el vector resultante del producto de la matriz por el vector. Asumir que las dimensiones son tales que es posible realizar el producto.
4. Desarrollar un subprograma que reciba una matriz G, un vector X1y un vector C y entregue el vector resultante de la operación X2=G*X1+ C. Asumir que las dimensiones son tales que es posible realizar el producto.
5. Desarrolle un subprograma que dados dos vectores entregue la norma infinito del vector diferencia.

Algoritmos sobre Métodos de Cálculo Numérico

1. Desarrolle un algoritmo que realice el método de Bisección. Si usa subprogramas detalle los mismos.
2. Desarrolle un algoritmo que realice el método de la Secante. Si usa subprogramas detalle los mismos.
3. Desarrolle un algoritmo que realice el método de Jacobi. Si usa subprogramas detalle los mismos.

Ejercicios Integradores

Completar con los "comandos correspondientes" según sea necesario.

1. Desarrolle un algoritmo para leer dos vectores, calcular y entregar su producto escalar.

```
%Producto escalar
for i=1:5
  b(i)=input ('íngrese componente');
end
a=[ 2; 4; 6; 8; 10];
sum=0;
[                 .]
    sum=sum+a(i)*b(i);
end
[                          .]
```

2. Desarrolle un algoritmo para leer un vector, calcular su norma cuadrática; y si es no nula entregar el versor unitario. De lo contrario debe entregar la leyenda "vector nulo"

```
%Versor de un vector dado
N=input ('íngrese dimensión del vector');
for i=1:N
  a(i)=input ('íngrese componente');
end

[                  .]
for i=1:N
    sum=sum+a(i)*a(i);
end
norma2 = sum^0.5

[                  .]
    disp ('el vector es nulo')
[                  .]
    disp ('el versor es:')
    for i=1:N
      [                  .]
      disp (' '), v(i)
    end
[                  .]
```

3. Desarrolle un algoritmo para leer un vector, calcular su norma infinita; y si es no nula entregar el versor unitario. De lo contrario debe entregar la leyenda "vector nulo"

```
%Versor de un vector dado
N=input ('íngrese dimensión del vector');
for i=1:N
  a(i)=input ('íngrese componente');
end

[         ;]
for i=1:N
    if (abs(a(i)) > norinf)
     norinf= abs(a(i));
    end
end
%
```

```
if norinf==0
    disp ('el vector es nulo')
          .
    disp ('el versor es:')
                          .
       v(i)= a(i)/norinf;
       disp (' '), v(i)
                          .
end
```

4. Desarrolle un algoritmo para leer un vector con las notas de los exámenes de un alumno. Cuando lea un una nota negativa, debe detener la lectura, calcular el promedio de las notas leídas, y entregar el promedio y la cantidad de notas leídas.

```
%Promedio Notas
%Declaracion de variables
i=1
a(i)=input ('íngrese componente');
%
           .  (a(i)>= 0)

  i = i + 1;
  a(i)=input ('íngrese componente');

           .
dim=i-1;
sum=0;
for                .
    sum=sum+a(i);
end
%
prom=sum/dim
disp ('El promedio es'), prom
disp ('El numero de notas es:'), dim
```

5. Escriba un algoritmo que mientras la función $f(x)=(3x^2-12)$ sea distinto de cero o el número de iteraciones sea menor que 100, lea un valor de abscisa x0, calcule la función f(x0) en ese valor x0, y sólo entregue un valor de x0 tal que la función f(x) es cero.

```
tol=1e-12;
max=      ;
k=1;
x=input ('íngrese abscisa');
f= 3*x^2-12;
A(1,k)=x;
A(2,k)=f;
%
.        (abs(f) > tol ……. k < max)
 k =k + 1;
 x=input ('íngrese componente');
 f= 3*x^2-12 ;
 A(1,k)=x;
 A(2,k)=f;
        .
disp(' '),A
```

Guía de Estudio para Raíces de Ecuaciones No Lineales.

Contenido

Guía de Estudio para Raíces de Ecuaciones No Lineales. 1
Bibliografía recomendada: 1
Objetivo: 1
Definiciones 1
Ejercicios del Método de Bisección y Regula Falsi 2
Ejercicios del Método de la Secante 4
Ejercicios del Método Newton Raphson 5
Ejercicios del Método Newton Raphson con MATLAB 6
Respuesta ejercicio. Opción 1 7
Respuesta ejercicio. Opción 2 8
Respuesta ejercicio. Opción 3 9
Respuesta ejercicio. Opción 4 11
Ejercicios del Método de Punto Fijo 12
Ejercicios Teóricos 13
Ejercicios de aplicaciones 13
Ejercicios Integradores 15

Bibliografía recomendada:

J. Mathews, J. Fink (2000). *Métodos Numéricos con Matlab.* Prentice Hall.
S. Nakamura (1997). *Análisis Numérico y Visualización Gráfica con MATLAB.* Prentice Hall
S. Nakamura (1992). *Métodos Numéricos Aplicados con Software.* Prentice Hall
S. Chapra, R. Canale (1999). *Métodos Numéricos para Ingenieros*, Mc Graw Hill.
R. Burden, J. Faires (1998). *Análisis Numérico*, International Thomson Editores.

Objetivo:

Se pretende comprender a los Métodos para obtener Raíces de Ecuaciones No lineales, como Métodos Iterativos en general.Se busca poder implementarlos en planillas de cálculo a modo de Tabla de Cálculo y en programas de MATLAB.

Definiciones

La obtención de las ***raíces de ecuaciones no lineales*** se hará mediante ***Métodos Iterativos***. Cualquier método iterativo es un procedimiento repetitivo que origina una sucesión de soluciones aproximadas. En general, tienen los siguientes “elementos” en su procedimiento (algoritmo):

- ***Condiciones de inicialización***, que son los requisitos que deben cumplir los datos iniciales.
- ***Fórmula de recurrencia***, que origina la sucesión de soluciones aproximadas.
- ***Control de detención***, que verifica la “certeza” de la solución aproximada.
- ***Actualización de variables***, que cuando el control de detención “falla” indicando que conviene obtener una nueva solución aproximada, incluye la última aproximación como dato inicial que cumple la condición de inicialización.

Las iteraciones se repiten hasta tanto el control de detención “no falla” o, en otras palabras, hasta que la solución aproximada es tan precisa como se pretende.

Ejercicios del Método de Bisección y Regula Falsi

1) Dada la función $f(x) = x^3 - 27$ y los intervalos de la variable x, $I_1 = [1{,}5;\ 4]$ $I_2 = [4,\ 6]$,
   a) aplicando la condición de inicialización del Método de Bisección, justifique en qué intervalo f(x) tiene una raíz.
   b) Aplicando el algoritmo del Método de Bisección, comprobar que las primeras 10 iteraciones son:

itera	a	b	f(a)	f(b)	r	f(r)	Error Bisección
1	1,5	4	-23,625	37	2,75	-6,203125	6,20E+00
2	2,75	4	-6,203125	37	3,375	11,4433594	1,14E+01
3	2,75	3,375	-6,203125	11,4433594	3,0625	1,72290039	1,72E+00
4	2,75	3,0625	-6,203125	1,72290039	2,90625	-2,45297241	2,45E+00
5	2,90625	3,0625	-2,45297241	1,72290039	2,984375	-0,41968155	4,20E-01
6	2,984375	3,0625	-0,41968155	1,72290039	3,0234375	0,63776922	6,38E-01
7	2,984375	3,0234375	-0,41968155	0,63776922	3,00390625	0,10560614	1,06E-01
8	2,984375	3,00390625	-0,41968155	0,10560614	2,99414063	-0,15789434	1,58E-01
9	2,99414063	3,00390625	-0,15789434	0,10560614	2,99902344	-0,02635861	2,64E-02
10	2,99902344	3,00390625	-0,02635861	0,10560614	3,00146484	0,0395701	3,96E-02

   c) Notar que como Error de Bisección se ha tomado el valor absoluto de f(r), y que según la tolerancia elegida, es la precisión obtenida en la aproximación de la raíz. En esta cantidad de iteraciones no se alcanza el valor exacto, pero si valores suficientemente precisos para muchas aplicaciones en ingeniería.

   d) Identifique en la Tabla anterior cual es la columna en la que se tiene el resultado de la Fórmula de Recurrencia del método.
   e) Identifique en la Tabla anterior cuales son las columnas en las que se tienen los resultados de la actualización de variables, y exprese cuales son las expresiones de dicha actualización.
   f) Identifique cuales son las columnas que se deben modificar en su forma de calcularlas para encontrar la raíz de otra ecuación no lineal
   g) Identifique cuales son las columnas que se deben modificar en su forma de calcularlas para encontrar la raíz mediante el método de Regula Falsi.

   h) Aplicando el algoritmo del Método de Regula Falsi, comprobar que las primeras 10 iteraciones son:

itera	a	b	f(a)	f(b)	r	f(r)	Error Regula Falsi
1	1,5	4	-23,625	37	2,4742268	-11,85328261	1,19E+01
2	2,4742268	4	-11,85328261	37	2,84442549	-3,986446419	3,99E+00
3	2,84442549	4	-3,986446419	37	2,95681961	-1,149170006	1,15E+00
4	2,95681961	4	-1,149170006	37	2,98824341	-0,31618567	3,16E-01
5	2,98824341	4	-0,31618567	37	2,99681617	-0,085872116	8,59E-02
6	2,99681617	4	-0,085872116	37	2,99913904	-0,023239257	2,32E-02

   i) Comprar la evolución del Error de Bisección con el Error de Regula Falsi

2) Realice un análisis para cada una de las siguientes funciones; basándose en dicho análisis, decida si hay o no al menos una raíz en el intervalo dado.
   a) $f(x) = \dfrac{x^2 + 2}{x^2 - 1}$ en el intervalo [0,5; 1,5].

b) $f(x) = x - 2^{-x}$, en los intervalos [0, 1] y [2, 3].
c) $f(x) = e^{-x} - x$ en [1, 3] y [-1, 0,5].
d) $f(x) = x^5 - 32$ en [1,3 ; 2,9].

3) Si es posible resuelva por el método indicado cada ejercicio hasta la cuarta iteración (k = 0, 1, . ..4) y evalúe tres medidas del error; si no es posible, indique por qué:

Ecuación	Método	Intervalo
a) $x - 2^{-x} = 0$	Bisección	[0,1]
c) $x - 2^{-x} = 0$	Bisección	[2, 3]
d) $0 = \dfrac{x^2 + 2}{x^2 - 1}$	Bisección	[0,5; 1,5]
f) $e^{-x} - x=0$	Bisección	[1, 3]
g) $e^{-x} - x=0$	regula falsi	[-1; 0,5]
b) $x - 2^{-x} = 0$	regula falsi	[0,1]
e) $\ln x^2 = 0,7$	regula falsi	[0,5; 2]

4) Dada la función $f(x) = x^3 - 27$ y el intervalo de la variable x, $I_1 = [1,5; 4]$, escriba un programa en MATLAB que encuentre una aproximación de la raíz

El siguiente archivo Bisección_00.m es una forma de resolver el ejercicio

```
function bisección_00
  %BLOQUE DE INICIALIZACION
    disp('Ingrese el primer numero:');
    a = input('');
    disp('Ingrese el segundo numero:');
    b = input('');
    disp('Ingrese el error tolerado:');
    e = input('');

    it = 0;                    % contador de iteraciones
    numsol =0;                 % indicador de solución obtenida
   % BLOQUE ITERATIVO
    while (numsol == 0)
        it = it + 1;           % Se hace una iteración
        r = (a+b)/2;           % Se aplica la FORMULA DE RECURRENCIA

        if ((abs(fc(r)))<e )      % Bloque CONTROL DE DETENCION
            numsol=1;             % Hace uso de la función fc(x)
        end
        if ((it == 1000))
            numsol = 999;
        end

        if (fc(a) * fc(r) > 0) %Bloque ACTUALIZACIÓN DE VARIABLES
           a = r;              % Hace uso de la función fc(x)
        else
           b = r;
        end
   end
    disp(el resultado es r= '); disp(r);
    disp('número de iteraciones:'); disp(it);
end
```

La función fc(r) debe definirse en el mismo archivo bisección_00.m o en otro archivo fc.m, de la siguiente ma:

```
      function fx = fc(x)
            fx=x^3-27;
      end
```

5) Modifique el programa MATLAB anterior para incorporar como criterio de detención que el valor del cambio de las aproximaciones de la raíz entre dos iteraciones consecutivas sea inferior a una tolerancia deseada.
6) Modifique el programa MATLAB anterior para encontrar la raíz de la función $f(x) = x^3 - 27$ en el intervalo de la variable x, $I_1 = [1{,}5; 4]$.

Ejercicios del Método de la Secante

Encontrar la raíz de la función $f(x) = x^3 - 27$.

Resolución. Se deben conocer dos aproximaciones anteriores. Así por ejemplo se tienen los valores obtenidos con el método de bisección x= 2,5 y 2,75. Se puede ordenar el cálculo con la siguiente Tabla.

iteración	*raíz*	*función* $f(x_i)$	*Pendiente* m_i	*Nueva Raíz* $x_{i+1} = x_i - \frac{f(x_i)}{m_i}$
i-1	x_{i-1}	$f(x_{i-1})$		
i	x_i	$f(x_i)$	$m_i = \frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}}$	$x_{i+1} = x_i - \frac{f(x_i)}{m_i}$
i+1				

que para los datos resulta

iter	raíz	función	Pendiente	Nueva Raíz
0	2,5	-11,375		
1	2,75	-6,203125	20,6875	3,04984894
2	3,04984894	1,36840957	25,2511632	2,995657
3	2,995657	-0,11709132	27,4118408	2,99992856
4	2,99992856	-0,00192883	26,9602892	3,0000001
5	3,0000001	2,7951E-06		

Se puede observar que en cuatro iteraciones se consigue una precisión muy alta.

7) Realice cuatro iteraciones con el Método de la Secante para las siguientes ecuaciones y aproximaciones dadas

a)	$e^{-x} - x=0$	-1	0,5
b)	$x - 2^{-x} = 0$	0	1
c)	$\ln x^2 = 0{,}7$	0,5	2]

Ejercicios del Método Newton Raphson

Ejercicio modelo con aplicación de planilla de cálculo.

Halle el valor de $+\sqrt{4}$ con un error inferior a 0,01, utilizando el **método de Newton-Raphson**, con valor inicial $x_0 = 1{,}3$.

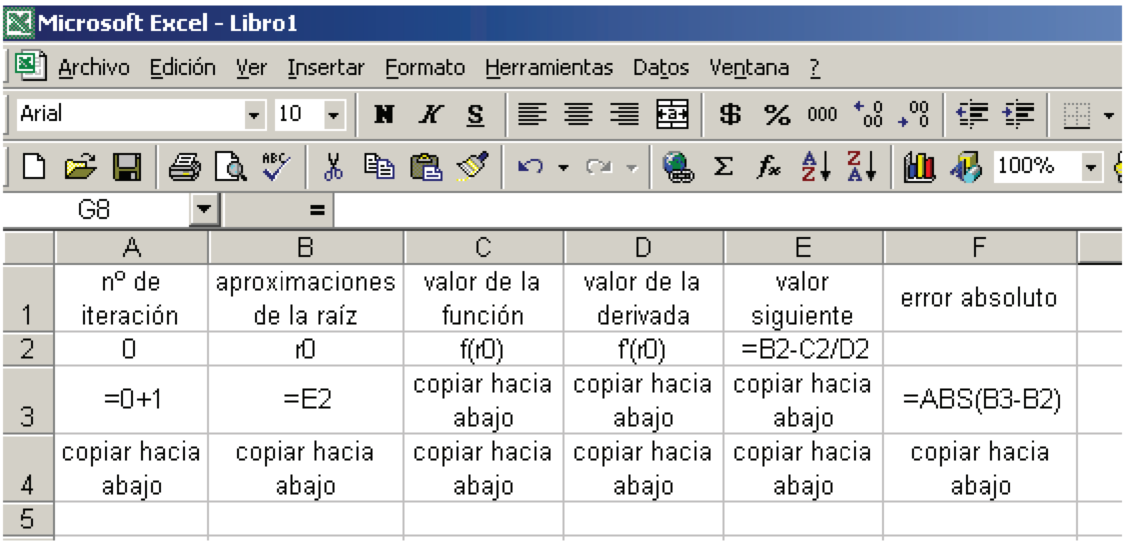
Resolución. Se debe obtener la raíz positiva de $f(x) = x^2 - 4$, que es el valor deseado.

Para ello se crea una tabla como la siguiente:

número de iteración	valor de abscisa	Función $f(x_i)$	Derivada $f'(x_i)$	valor siguiente $x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$	error %	¿Se cumple la condición?
i	x_i	$f(x_i) = x_i^2 - 4$	$f'(x_i) = 2x_i$		$\left\lvert \frac{x_{i+1} - x_i}{x_{i+1}} \right\rvert$	¿es el error% menor que 0,01?
0	1,3	-2,31	2,6	2,188461538	0,405975395	No
1	2,188461538	0,789363905	4,376923077	2,008114776	0,089808991	No
2	2,008114776	0,032524955	4,016229553	2,000016396	0,004049157	Sí
3	2,000016396	6,55838E-05	4,000032792	2	8,1979E-06	Sí
4	2	2,68821E-10	4	2	3,36027E-11	Sí
5	2	0	4	2	0	Sí
6	2	0	4	2	0	Sí

que permite hallar una aproximación del valor de la raíz buscada, x = 2,008 con un error menor que 0,01 en tres iteraciones. Se han agregado 4 filas más a la tabla para mostrar cómo podría haber continuado el proceso con unas cuantas iteraciones más.

Se incluye a efectos ilustrativos su implementación en Excel:

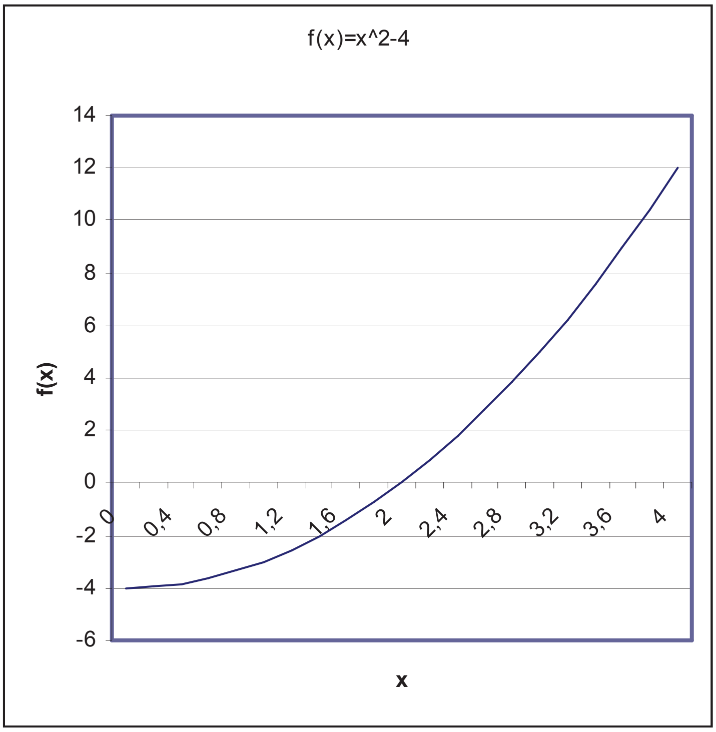


	A	B	C	D	E	F
1	nº de iteración	aproximaciones de la raíz	valor de la función	valor de la derivada	valor siguiente	error absoluto
2	0	r0	f(r0)	f'(r0)	=B2-C2/D2	
3	=0+1	=E2	copiar hacia abajo	copiar hacia abajo	copiar hacia abajo	=ABS(B3-B2)
4	copiar hacia abajo	copiar hacia abajo	copiar hacia abajo	copiar hacia abajo	copiar hacia abajo	copiar hacia abajo
5						

Además conviene completar esta información con un gráfico de la función con la que se trabaja, que se puede realizar por ejemplo de la siguiente manera:

x	f(x)
0	=A2^2-4
=A2+0,2	copiar hacia abajo
copiar hacia abajo	

x	f(x)
0	-4
0,2	-3,96
0,4	-3,84
0,6	-3,64
0,8	-3,36
1	-3
1,2	-2,56
1,4	-2,04
1,6	-1,44
1,8	-0,76
2	0
2,2	0,84
2,4	1,76
2,6	2,76
2,8	3,84
3	5
3,2	6,24
3,4	7,56
3,6	8,96
3,8	10,44
4	12



8) Resuelva utilizando una planilla de cálculo la ecuación x + 0,5 + 2 cos π x = 0 por el método de Newton-Raphson con valor inicial x_0 = 1,4. Considere un error de 10^{-2}.

9) Si es posible resuelva cada ejercicio hasta la cuarta iteración (k = 0, 1, ..., 4) y evalúe tres medidas del error; si no es posible, indique por qué:

b) Resuelva 4 cos x = e^x por el método de Newton-Raphson con valor inicial $x_0 = 2$.
c) Resuelva 4 cos x = e^x por el método de la secante en el intervalo [π/4; π/2].
d) Halle la raíz de f(x) = x^2 por el método de Newton-Raphson con valor inicial $x_0 = 1$.
e) Calcule una aproximación de $\sqrt{3}$ por el método de Newton-Raphson con valor inicial $x_0 = 2$.

Ejercicios del Método Newton Raphson con MATLAB

Se pretende encontrar las raíces de la siguiente función no lineal:

$$f(x) = x^2 + 6 \cdot x - 27.$$

Se presentan 4 soluciones en MATLAB.

Respuesta ejercicio. Opción 1

Se hace todo el algoritmo usando un solo file de Matlab. Ese file se llama f1_nr_00_while.m .

Cada línea del file es un COMANDO, SENTENCIA u ORDEN DE EJECUCIÓN para MATLAB.

Cada línea que empiece con un % es un COMENTARIO para MATLAB y no un COMANDO.

Todo texto precedido por % es un COMENTARIO, desde el % hasta el final de la línea. Se puede usar para poner comentarios en líneas de comandos.

File f1_nr_00_while.m

```
% f1_nr_00_while.m
%EL método de Newton Raphson con while
%TODO EN UN SOLO ARCHIVO para f(x)=x^2+6x-27
%Sin funciones usadas. Inconveniente:
%    Cada vez que se cambia la f(x) hay que reprogramar las líneas
%    donde se calcula f; y; y df. Están destacadas en ROJO.
%
%DATOS
p0= 89                          % punto inicial
epsilonfun=1.e-6                % tolerancia en cero de la función
delta=1.e-3                     % tolerancia para cambio en Raíz
epsi100=0.01                    % tolerancia porcentual de cambio
                                % de Raíz
kmax1=100                       % número máximo de iteraciones
%
%CICLO ITERATIVO
flag=true;
k=0;
while(flag)
   f = p0^2+6*p0-27;
   df= 2*p0+6;
   p1=p0-f/df;                  % Recurrencia
   k=k+1;
   %
   y = p1^2+6*p1-27;            % cálculo de la f en nueva aproximación
   err=abs(p1-p0);              % cálculo del cambio en la aproximación
   err100=abs((p1-p0)/p1);      % cálculo del cambio en la aproximación
   % CONTROLES DE DETENCIÓN
    if ((abs(y)<epsilonfun))
       flag=false;
    end
    if (err<delta)
       flag=false;
    end
    if ((err100 < epsi100))
       flag=false;
    end
    if (k > kmax1)
       flag=false;
    end
    % ACTUALIZACIÓN
    p0=p1;
end
```

```
% DEVOLUCION DE RESULTADOS
 p0
 y
 k
 err
 err100
```

Respuesta ejercicio. Opción 2

f2_nr_00_while.m

Se programaron el cálculo de la función f(x) cuya raíz se pretende obtener en un file; y el cálculo de la derivada de f(x) en otro file aparte. Los datos iniciales se mantuvieron en el archivo Principal, que sólo se cambia en los datos iniciales pero NO el resto de las líneas. Esta última es la ventaja respecto de la Opción 1.

```
% f2_nr_00_while.m
%EL método de Newton Raphson con while
%TODO EN UN SOLO ARCHIVO para f(x)=x^2+6x-27
%CON funciones para calcular f y df. Definidas en f2_mif.m y f2_dmif.m
%Inconveniente:
%              Cada vez que se cambia la f(x)
%              hay que reprogramar f2_mif.m y f2_dmif.m
%DATOS INICIALES
p0= 65
delta=1.e-3
epsilon=1.e-6
max1=100
%CICLO ITERATIVO
flag=true;
k=0;
while(flag)
   f = f2_mif(p0);      % f y df son funciones definidas externamente.
   df= f2_dmif(p0);
   p1=p0-f/df;            % RECURRENCIA
   k=k+1;
   y = f2_mif(p1);          % cálculo de la f en nueva aproximación
   err=abs(p1-p0);          % cálculo del cambio en la aproximación
   err100=abs((p1-p0)/p1);  % cálculo del cambio en la aproximación
   % CONTROLES DE DETENCIÓN
    if ((abs(y)<epsilonfun))
       flag=false;
    end
    if (err<delta)
       flag=false;
    end
    if ((err100 < epsi100))
       flag=false;
    end
    if (k > kmax1)
       flag=false;
    end
    % ACTUALIZACIÓN
    p0=p1;
end
```

```
% DEVOLUCIÓN DE RESULTADOS
 p0
 y
 k
 err
 err100
```

Se debe armar los files f2_mif.m y f2_dmif.m para la función cuyos ceros se pretende encontrar. Estos files DEBEN mantener su nombre (f2_mif.m y f2_dmif.m), pero en su interior se DEBE escribir la función cuyos ceros se pretende encontrar.
Para el caso de la función `f(x)=x^2+6x-27,` los files son los siguientes:
File f2_mif.m

```
%definición de funciones
function mif = f2_mif(x)
  mif= x^2+6*x-27;
```

File f2_dmif.m

```
%definición de funciones
function dmif = f2_dmif(x)
  dmif= 2*x+6;
```

Se debe destacar que en cada uno de estos files se define una función de una variable x. Dicha función se evaluará con la variable x. Al momento de usar estas funciones en el programa principal que las invoque (que las llame), en lugar de la variable x podrá ir cualquier otra variable del programa principal que tenga un valor compatible con la variable x.

El resultado de la operación algebraica que se realiza en f2_mif.m se almacena en la variable del programa principal a la que esté igualada f2_mif. Así por ejemplo:

Si en el programa principal se escribe la línea

```
f = f2_mif(p0);
```

en la variable f del programa principal se tendrá el resultado de la operación algebraica que esté definida en el file f2_mif.m, para cuando la variable x toma el valor p0 del programa principal.

Si en el programa principal se escribe la línea

```
df= f2_dmif(p0);
```

en la variable df del programa principal se tendrá el resultado de la operación algebraica que esté definida en el file f2_dmif.m, para cuando la variable x toma el valor p0 del programa principal.

Respuesta ejercicio. Opción 3
f3_nr_00_while.m
Se programaron el cálculo de la función f(x) y de la derivada de la función cuya raíz se pretende obtener en un file aparte; en un único file aparte. Los datos iniciales se mantuvieron en el archivo Principal, que sólo se cambia en los datos iniciales pero NO el resto de las líneas.
La novedad es el hecho de calcular la función y su derivada en un mismo file, que es el único a modificar en el caso de cambiar la función cuya raíz se desea calcular.

```
% f3_nr_00_while.m
%EL metodo de Newton Raphson con while
% para la función f(x)=x^2+6x-27
% CON función f3_mif.m
% que calcula para un x dado el valor de la función y de su derivada
```

```
%      Cada vez que se cambia la f(x)
%%     hay que reprogramar f3_mif.m
%DATOS INICIALES
p0= 65
delta=1.e-3
epsilon=1.e-6
max1=100
%
%CICLO ITERATIVO
flag=true;
k=0;
while(flag)
    [f, df] = f3_mif(p0);    % cálculo de f y df con una sola función
                             % externa
   p1=p0-f/df;               % RECURRENCA
   k=k+1;
   %
   [f, df] = f3_mif(p1);      % cálculo de la f en nueva aproximación
   y = f;
   err=abs(p1-p0);            % cálculo del cambio en la aproximación
   err100=abs((p1-p0)/p1);    % cálculo del cambio en la aproximación
   % CONTROLES DE DETENCIÓN
    if ((abs(y)<epsilonfun))
       flag=false;
    end
    if (err<delta)
       flag=false;
    end
    if ((err100 < epsi100))
       flag=false;
    end
    if (k > kmax1)
       flag=false;
    end
    % ACTUALIZACiÖN
    p0=p1;
end
% DEVOLUCION DE RESULTADOS
 p0
 y
 k
 err
 err100
```

La función f3_mif.m es una función de una entrada y dos salidas. La entrada es la variable x y las salidas son el valor de la función que se almacena en mif; y de la derivada que se almacena en dmif.

```
%definicion de funciones
function [mif, dmif] = f3_mif(x)
  mif= x^2+6*x-27;
  dmif=2*x+6;
```

Al llamar desde el programa principal a la función f3_mif se le debe dar el valor de x para el cual se pretende calcular la función y su derivada. Este valor de x será el de alguna variable compatible del

programa principal. Los valores calculados por la f3_mif serán asignados en variables compatibles del programa principal.
Así por ejemplo
```
[y, dy] = f3_mif(p0)
```
Indica que en y se almacenará el valor de la función evaluada en p0; y en dy el de su derivada también evaluada en p0.

Respuesta ejercicio. Opción 4
f4_nr_00_while.m
Se programaron el cálculo de la función f(x) y de la derivada de la función cuya raiz se pretende obtener en un único file (f4_mif.m) aparte del programa principal (f4_nr_00_while.m). Los datos iniciales se escriben en un file (f4_input.m) aparte del archivo Principal. De esta manera el programa principal nunca se modifica.

```
%f4_nr_00_while.m
% EL método de Newton Raphson con while
% para una función definida en f4_mif.m
% que calcula para un x dado el valor de la función y de su derivada
%     Cada vez que se cambia la f(x)
%    hay que reprogramar f4_mif.m
% Los datos se deben escribir en el f4_input.m
% Este programa principal no se modifica nunca.
% DATOS INICIALES
f4_input;
%CICLO ITERATIVO
flag=true;
k=0;
while(flag)
   [f, df] = f4_mif(p0);    % cálculo de f y df con una sola función
                            % externa
   p1=p0-f/df;              % RECURRENCA
   k=k+1;
   [f, df] = f4_mif(p1);       % cálculo de la f en nueva aproximación
   y = f;
   err=abs(p1-p0);             % cálculo del cambio en la aproximación
   err100=abs((p1-p0)/p1);     % cálculo del cambio en la aproximación
   % CONTROLES DE DETENCIÓN
    if ((abs(y)<epsilonfun))
       flag=false;
    end
    if (err<delta)
       flag=false;
    end
    if ((err100 < epsi100))
       flag=false;
    end
    if (k > kmax1)
       flag=false;
    end
    % ACTUALIZACiÖN
    p0=p1;
end
```

```
% DEVOLUCION DE RESULTADOS
 p0
 y
 k
 err
 err100
```

El file de ingreso de datos resulta

```
% (f4_input.m)
%DATOS INICIALES
p0= 65
delta=1.e-3
epsilon=1.e-6
max1=100
```

El file de cálculo de la función y de su derivada es

```
% (f4_mif.m)
%definición de funciones
function [mif, dmif] = f4_mif(x)
  mif= x^2+6*x-27;
  dmif=2*x+6;
```

Ejercicios del Método de Punto Fijo

10) Si es posible resuelva cada ejercicio hasta la cuarta iteración (k = 0, 1, ..., 4) y evalúe tres medidas del error; si no es posible, indique por qué:

a) Halle la raíz de $f(x) = e^{-x} - x$ por el método del punto fijo considere $g(x) = e^{-x}$ y x_0=0.

f) Calcule una aproximación de $\sqrt{3}$ por el método del punto fijo con $g(x) = -\frac{1}{3}x^2 + x + 1$ y x_0=2.

11) La ecuación $x^3 + 4x^2 - 10 = 0$ tiene una sola raíz en [1, 2]. Existen muchas maneras de cambiar la ecuación a la forma x = g (x), efectuando manipulaciones algebraicas simples. Algunas de las expresiones que se puede obtener son:

Este ejercicio está basado en un ejemplo presentado en el libro Análisis Numérico de Burden y Faires, ed. 1985, Pág. 49.

$$x = g_1(x) = x - x^3 - 4x^2 + 10 \qquad x = g_2(x) = \left(\frac{10}{x} - 4x\right)^{\frac{1}{2}}$$

$$x = g_3(x) = \frac{1}{2}\left(10 - x^3\right)^{\frac{1}{2}} \qquad x = g_4(x) = \left(\frac{10}{4 + x}\right)^{\frac{1}{2}}$$

$$x = g_5(x) = x - \frac{x^3 + 4x^2 - 10}{3x^2 + 8x}$$

a) Con r_0 = 1,5 aplique el método de punto fijo a cada una de las cinco alternativas para g y compare los resultados.
b) ¿Para cuáles de las funciones g_i converge el método?
c) Para las funciones g_i que permiten la convergencia del método, verifique que el punto fijo hallado de cada una de ellas es una solución de la ecuación original.

12) ¿Cuál es la condición de convergencia para el método del punto fijo? Basándose en ello, indique cuáles de las siguientes manipulaciones de la ecuación dada permiten una exitosa aplicación de dicho método para hallar la solución de la ecuación en el intervalo [1, 3] con valor inicial $r_0 = 1{,}6$:

Ecuación: $f(x) = x^2 - 3 = 0$

Despeje:

$$x = g_1(x) = -\frac{1}{3}x^2 + x + 1$$

$$x = g_2(x) = \frac{1}{3}x^2 + x - 1$$

$$x = g_3(x) = \frac{3}{x}$$

Ejercicios Teóricos

13) Escriba expresiones algebraicas que puedan usarse como "control de detención" y representen:
   a) El cero aproximado de la función en la solución aproximada. Interprete gráficamente.
   b) El cambio absoluto en la solución aproximada entre dos iteraciones consecutivas.
   c) El cambio relativo en la solución aproximada entre dos iteraciones consecutivas.

14) Para los métodos de Bisección, Regula Falsi, de la Secante, de Newton-Raphson y de Punto Fijo:
   a) Escriba las condiciones de inicialización para aplicar el algoritmo correspondiente.
   b) Demuestre la Fórmula de Recurrencia.
   c) ¿Cuál o cuáles son los posibles Criterios de Detención?
   d) Escriba la Actualización de Variables para cada método.
   e) Indique para cada método sus ventajas y desventajas.
   f) ¿En qué casos se puede asegurar que cada método converge?

15) Demuestre la condición de convergencia genérica de los métodos de Punto Fijo.

16) Acote el error para el método de bisección.

17) Compare e indique cuáles son las diferencias entre los métodos de la falsa posición y de la secante.

18) Describa e interprete geométricamente el método de Newton-Raphson para determinar raíces de ecuaciones no lineales.

19) Derive la fórmula de Newton-Raphson usando serie de Taylor y halle el orden del error (véase Chapra y Canale).

20) Demuestre que el método de Newton-Raphson es un caso particular del método de punto fijo. A partir de esta idea, halle una condición de convergencia para el método de Newton-Raphson.

Ejercicios de aplicaciones

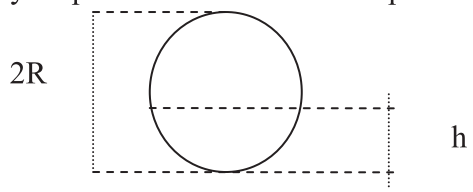
1) Obtener la raíz aproximada de la siguiente ecuación no lineal:

$$f(x) = x - \cos(x) = 0$$

2) Las frecuencias de vibración libre de una viga empotrada libre son las raíces de la siguiente ecuación trascendente:

$$f(x) = \cosh(x)\cos(x) + 1 = 0$$

3) El volumen *V* de un sector esférico de radio *R* y de profundidad *h* está dado por:



$$\frac{\pi}{3}\left(3\cdot R\cdot h^{2}-h^{3}\right)=V$$

Para R=6000 mm; V=200000e6 mm, determine la profundidad *h* utilizando alguno de los métodos de solución de ecuaciones no lineales desarrollados anteriormente.

4) La ecuación de un sistema fuerza-resorte la condición de equilibrio estático está dada por:

$$k\cdot u-p=0\,.$$

Cuando la *k* del resorte es un valor constante dato, para cada valor de *p* es fácil obtener qué valor de *u* le corresponde. Así la "trayectoria de equilibrio" dada por los pares ordenados (*u,p*) es simple de obtener. Pero cuando la *k* del resorte es una función de los desplazamientos *u*, la ecuación de equilibrio es no lineal. Para cada valor de *p*, se debe resolver una ecuación no lineal para obtener el *u* solución de la ecuación de equilibrio. Sea por ejemplo:

$$k=k_{0}\sin\left(\frac{\pi}{2}\frac{u}{u_{y}}\right)\,,$$

con los valores k_0 y u_y dados como datos. Así, la ecuación de equilibrio no lineal es:

$$k_{0}\sin\left(\frac{\pi}{2}\frac{u}{u_{y}}\right)\cdot u-p=0\,.$$

Utilizando el Método de Newton Raphson y comenzando desde la solución trivial (0,0) para distintos valores incrementales de *p* obtenga los correspondientes valores de *u*. Graficar *p* en función de *u*.

Para valores de k_0=1000 y u_y=4, encontrar el valor aproximado de *p* tal que la "trayectoria de equilibrio" dada por los pares ordenados (*u,p*) tiene pendiente horizontal. Grafique *p* en función de *u*, para varios valores de *p*, hasta "cerca de la pendiente horizontal".

Discuta en esa condición qué ocurre con el método de Newton Raphson.

5) Mediante la implementación de programas de MATLAB adaptados a cada método, resuelva cada ejercicio aplicando el método pedido. Además, realice una representación gráfica de la función en un entorno de la raíz.

a) Resuelva $x^4 - 2x^3 - 4x^2 + 4x + 4 = 0$ por el método de bisección en los intervalos [-2,0], [0,2] y [1,2]. Considere un error de 10^{-2}.
b) Resuelva $e^{x} + 2^{-x} + 2\cos x - 6 = 0$ por el método de la secante en el intervalo [1,2]. Considere un error de 10^{-5}.
c) Resuelva $e^{x} - x^2 + 3x - 2 = 0$ por el método de regula falsi en el intervalo [0,1]. Considere un error de 10^{-5}.
d) Resuelva la ecuación $\ln(x^2)=0{,}7$ por el método del punto fijo con valor inicial $x_0 = 0{,}6$. Considere $g(x) = -(\ln(x^2) - 0{,}7) + x$ y trabaje con un error de 10^{-4}.
e) Halle la raíz real positiva de $e^{-x} - x = 5$ por el método del punto fijo. Para ello elija el valor inicial x_0 y la función g(x) que considere convenientes. Trabaje con un error de 10^{-4}.

Ejercicios Integradores

6) Dado el siguiente programa de MATLAB para calcular la raíz de 3*x^2-12=0 mediante el método de Newton Raphson, completar con los comandos necesarios y responder.

```
tol=1e-12;
max=30;
k=1;
x=input ('íngrese abscisa ');
f= 3*x^2-12;
df= 6*x;
A(1,k)=x;
A(2,k)=f;
%
[         .] ((abs(f) > tol) [.  ] (k < max))
      k =k + 1;
      x=x-f/df;
      f= 3*x^2-12 ;
      df=6*x;
      A(1,k)=x;
      A(2,k)=f;
[         .]
format long
disp(' '),A
```

- ¿Qué realiza la variable “tol”?
- ¿Qué almacena la matriz A?

Otra alternativa a este problema es el siguiente programa:

```
tol=1e-12;
tolx=1e-12;
max=30;
k=1;
x=input ('íngrese abscisa ');
f= 3*x^2-12;
df= 6*x;
A(1,k)=x;
A(2,k)=f;

nohaysol=true;
%
[       .]( nohaysol )
    k =k + 1;
    x=x-f/df;
    f= 3*x^2-12 ;
    df=6*x;
    A(1,k)=x;
    A(2,k)=f;
    [    .](abs(f) < tol )
       nohaysol=false;
    [     .]
    if ([k >       ] )
      nohaysol=false;
    end
    if (abs(A(1,k)-A(1,k-1)) < tolx )
      nohaysol=false;
    end
[     .]
disp(' '),A
```

- ¿Qué realiza la variable “tolx”?
- ¿Qué mide el último if `abs(A(1,k)-A(1,k-1)) < tolx` ?

7) Dado el siguiente programa de MATLAB para calcular la raíz de 3*x^2-12=0 mediante el método de la Secante, completar con los comandos necesarios y responder.

```
tol=1e-12;
tolx=1e-12;
max=30;
k=1;
x1=input ('íngrese aproximación 1 ');
x2=input ('íngrese aproximación 2 ');

A(1,k)=x2;
A(2,k)=3*x2^2-12;
%
nohaysol=true;
%
while (               . )
    k =k + 1;
    f1= 3*x1^2-12;
    f2= 3*x2^2-12;
    df= (f2-f1)/(x2-x1);
    x=                .;
    f= 3*x^2-12 ;
    A(1,k)=x;
    A(2,k)=f;
         (abs(f) < tol )
       nohaysol=false;
          .
    if (     >     )
      nohaysol=false;
    end
    if (abs(A(1,k)-A(1,k-1)) < tolx )
      nohaysol=false;
    end
    x1=x2;
    x2=x;
end
format long
disp(' '),A
```

Sin almacenar la historia de las aproximaciones, se puede sintetizar el algoritmo de la siguiente forma

```
tol=1e-12;
tolx=1e-12;
max=30;
k=.       ;
x1=input ('íngrese aproximación 1 ');
x2=input ('íngrese aproximación 2 ');
%
nohaysol=true;
%
.          ( nohaysol )
    k =k + 1;
    f1= 3*x1^2-12;
    f2= 3*x2^2-12;
    df= (f2-f1)/(x2-x1);
    x=x2-f2/df;
    f= 3*x^2-12 ;

    if (abs(f) < tol )
       …                  ;
    end
```

```
    ._________(k > max )
      nohaysol=false;
    end

    if (abs(x2-x) < tolx )
      _________________.;
    end

    x1=x2;
    x2=x;
________.
x
k
```

8) Dado el siguiente programa de MATLAB para calcular la raíz de 3*x^2-12=0 mediante el método de Regula Falsi, completar con los comandos necesarios y responder.

```
tol=1e-12;
tolx=1e-12;
max=30;
k=0;
a=input ('íngrese extremo a ');
b=input ('íngrese extremo b ');
fa= 3*a^2-12;
fb= 3*b^2-12;
if (fa*fb < 0)
   nohaysol=true;
else
   nohaysol=false;
   disp ('NO cumple con inicializacion de Reg. Falsi')
   break
end
%
___________. ( nohaysol )
    k =k + 1;
    fa= 3*a^2-12;
    fb= 3*b^2-12;
    df= (fb-fa)/(b-a);
    r=b-fb/df;
    fr= 3*r^2-12 ;
    _________.(abs(fr) < tol )
       nohaysol=false;
    _________.
    _________.(          )              %número máximo de iteraciones
      nohaysol=false;
    _________.
    if (abs(b-a) < tolx )
      nohaysol=false;
    end
    if (fa*fr < 0)
        ___=___;
        ___=___;
    end
    if (fb*fr < 0)
        ___=___;
        ___=___;
    end
_________.
r
k
```

9) Dado el siguiente programa de MATLAB para calcular la raíz de 3*x^2-12=0 mediante el método de Bisección, completar con los comandos necesarios y responder. Comparar con el Ejercicio anterior de Regula Falsi

```
tol=1e-12;
tolx=1e-12;
max=30;
k=0;
a=input ('ingrese extremo a ');
b=input ('ingrese extremo b ');
fa= 3*a^2-12;
fb= 3*b^2-12;
if (fa*fb < 0)
   nohaysol=true;
else
   nohaysol=false;
   disp ('NO cumple con inicializacion de Reg. Falsi')
   break
end
%
while ( nohaysol )
    k =k + 1;

    r=            ;       % Recuerrencia del método de Bisección

    fr= 3*r^2-12 ;
    if (abs(fr) < tol )
       nohaysol=false;
    end
    if (k > max )
      nohaysol=false;
    end
    if (abs(b-a) < tolx )
      nohaysol=false;
    end
    if (fa*fr < 0)
        a=a;
        b=r;
    end
    if (fb*fr < 0)
        a=r;
        b=b;
    end
end
r
k
```

10) Dado el siguiente programa de MATLAB para calcular la raíz de x^2-3=0 mediante el método de Punto Fijo, completar con los comandos necesarios y responder.

Comprobar que x=1+x-(x^2)/3 es equivalente a x^2-3=0 y cumple con la condición de convergencia del método de punto fijo en el el intervalo [1, 3] con valor inicial $r_0 = 1{,}6$

```
tol=1e-12;
tolx=1e-12;
max=30;
k=0;

r=input ('ingrese aprox. inicial ');

fr= r^2-3;
```

```
if (abs(fr) > tol)
   nohaysol=true;
else
   nohaysol=false;
   disp ('tenes mucha suerte!!')
   break
end
%
while (                  . )
    k =k + 1;
    rn= -((r^2)/3)+r+1;       %recurrencia de punto fijo
    fr= rn^2-3;

   .      (abs(fr) < tol )
       nohaysol=false;
        .

    if (k > max )
      nohaysol=false;
    end

    .     (abs(rn - r) < tolx )
      nohaysol=false;
          .

    r=      ;                % Actualización de variables

end
r
fr
k
```

Otra variante puede ser

```
tol=1e-12;
tolx=1e-12;
max=30;
k=0;
r=input ('íngrese aprox. inicial no nula ');
fr= r^2-3;

%
while ( (abs(fr) > tol ) & (k < max )  )
    k =k + 1;
    rn= -((r^2)/3)+r+1;       %recurrencia de punto fijo
    fr= rn^2-3;
    r=rn;
end
r
fr
k
```

Comparar con el ejemplo anterior los controles de detención.

Guía de Estudio para Sistemas de Ecuaciones Lineales.

Contenido

Guía de Estudio para Sistemas de Ecuaciones Lineales. 1
Bibliografía recomendada: 1
Objetivo: 1
Definiciones: 1
Ejercicios de Sistemas de Ecuaciones Lineales 2
Ejercicios de Sistemas de Ecuaciones Lineales con MATLAB 4

Bibliografía recomendada:

Apuntes de la Càtedra.
J. Mathews, J. Fink (2000). *Métodos Numéricos con Matlab.* Prentice Hall.
S. Nakamura (1997). *Análisis Numérico y Visualización Gráfica con MATLAB.* Prentice Hall
S. Nakamura (1992). *Métodos Numéricos Aplicados con Software.* Prentice Hall
S. Chapra, R. Canale (1999). *Métodos Numéricos para Ingenieros*, Mc Graw Hill.
R. Burden, J. Faires (1998). *Análisis Numérico*, International Thomson Editores.

Objetivo:

Se busca resolver sistemas de ecuaciones lineales mediante los métodos iterativos de Jacobi y de Gauss Seidel, implementándolos en planillas de cálculo y en programas de MATLAB

Definiciones:

La obtención de la solución de un ***sistema de ecuaciones lineales*** (SEL) se hará mediante ***Métodos Iterativos:*** procedimientos repetitivos que originan una sucesión de soluciones aproximadas. En general, tienen los siguientes “elementos” en su procedimiento (algoritmo):

- ***Condiciones de inicialización***, que son los requisitos que deben cumplir los datos iniciales.
- ***Fórmula de recurrencia***, que origina la sucesión de soluciones aproximadas.
- ***Control de detención***, que verifica la “certeza” de la solución aproximada.
- ***Actualización de variables***, que cuando el control de detención “falla” indicando que conviene obtener una nueva solución aproximada, incluye la última aproximación como dato inicial que cumple la condición de inicialización.

Las iteraciones se repiten hasta tanto el control de detención “no falla” o, en otras palabras, hasta que la solución aproximada es tan precisa como se pretende.

Ejercicios de Sistemas de Ecuaciones Lineales

1) Considerar un sistema de ecuaciones (dado en forma matricial) **A·x = b**. Para las siguientes matrices **A** indicar cuáles cumplen la condición de convergencia para el método Jacobi o de Gauss-Seidel.

$$A_1 = \begin{bmatrix} 0 & 3 \\ -4 & 2 \end{bmatrix} \quad A_2 = \begin{bmatrix} -2 & 1 \\ 3 & 4 \end{bmatrix} \quad \mathrm{A}_3 = \begin{bmatrix} -3 & 1 & -2 \\ 4 & -5 & 0 \\ 1 & -3 & 6 \end{bmatrix} \quad \mathrm{A}_4 = \begin{bmatrix} 0 & -2 & 6.8 & -11 \\ 1.2 & 4 & 8.9 & 0.4 \\ 2 & -9 & 1 & -5 \\ -5 & 1.6 & 1 & -2.3 \end{bmatrix} \quad \mathrm{A}_5 = \begin{bmatrix} 0 & 23 & 8 \\ 5 & -9 & 16 \\ 17 & -11 & -4 \end{bmatrix}$$

2) Resolver los siguientes sistemas **A·x = b** por el método de Jacobi. Realizar cuatro iteraciones y evaluar un criterio de detención en cada paso, y la norma del "vector residuo". Se define vector residuo a **r = A·x - b**

$$\begin{bmatrix} -2 & 1 \\ 3 & 4 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} = \begin{bmatrix} 2 \\ -1 \end{bmatrix}$$

$$\begin{bmatrix} -3 & 1 & -2 \\ 4 & -5 & 0 \\ 1 & -3 & 6 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{bmatrix} -2 \\ 5 \\ 6 \end{bmatrix}$$

3) Escribir expresiones algebraicas que puedan usarse como "control de detención" y representen:
   a) El cambio absoluto en la solución aproximada entre dos iteraciones consecutivas.
   b) El cambio relativo en la solución aproximada entre dos iteraciones consecutivas.

4) Describir la diferencia entre los métodos de Jacobi y de Gauss-Seidel.

5) Resolver los siguientes sistemas **A·x = b** por el método de Gauss Seidel. Realice cuatro iteraciones y evalúe un criterio de detención en cada paso, y la norma del "vector residuo". Se define vector residuo a **r = A·x - b**

$$\begin{bmatrix} -2 & 1 \\ 3 & 4 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} = \begin{bmatrix} 2 \\ -1 \end{bmatrix}$$

$$\begin{bmatrix} -3 & 1 & -2 \\ 4 & -5 & 0 \\ 1 & -3 & 6 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{bmatrix} -2 \\ 5 \\ 6 \end{bmatrix}$$

6) Interpretar geométricamente la aplicación del método de Gauss-Seidel, para el sistema

$$\begin{cases} 2\mathrm{x} + \mathrm{y} = 2 \\ \mathrm{x} - 2\mathrm{y} = -2 \end{cases}.$$

7) Implementar una planilla de cálculo que resuelva sistemas de ecuaciones de orden 4 por el método de Gauss-Seidel, con un error absoluto máximo aceptable de 0.01 y un error relativo máximo aceptable de 0.01. Asegurar que se realicen al menos 50 iteraciones y se distingan cuántas iteraciones hacen falta para cumplir las condiciones impuestas sobre cada error. Ejemplo:

	A	B	C	D	E	F	G	H	I
1	Matriz A:	a11	a12	a13	a14	Vector b	b1		
2		a21	a22	a23	a24		b2		
3		a31	a32	a33	a34		b3		
4		a41	a42	a43	a44		b4		
5	vector x transpuesto				Cálculos previos del error absoluto				ERROR ABSOLUTO
6	1	1	1	1	-	-	-	-	-
7	= 1/B1 (G1-C1·B6-D1·C6-E1·D6)	= 1/C2 (G2-B2·A7-D2·C6-E2·D6)	= 1/D3 (G3-B3·A7-C3·B7-E3·D6)	etc.	= ABS(A7-A6)	= ABS(B7-B6)	= ABS(C7-C6)	= ABS(D7-D6)	=MAX(E7:H7)
8									

Utilizar la planilla desarrollada para resolver los sistemas $\begin{bmatrix} 0 & -2 & 6.8 & -11 \\ 1.2 & 4 & 8.9 & 0.4 \\ 2 & -9 & 1 & -5 \\ -5 & 1.6 & 1 & -2.3 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{Bmatrix} = \begin{Bmatrix} 2 \\ 0 \\ 4 \\ -8 \end{Bmatrix}$

$$\begin{bmatrix} -3 & 1 & -2 \\ 4 & -5 & 0 \\ 1 & -3 & 6 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} -2 \\ 5 \\ 6 \end{Bmatrix}$$

8) Muestrar que el método de Jacobi es una aplicación matricial del método de punto fijo.
Para el siguiente sistema de ecuaciones $\mathbf{A_{3*}x}=\mathbf{b_3}$, encuentre la matriz **T** y el vector **c**, si se considera el mètodo de Jacobi como $\mathbf{x}_{(k)} = \mathbf{T}\ \mathbf{x}_{(k-1)} + \mathbf{c}$

$$A_3 = \begin{bmatrix} -3 & 1 & -2 \\ 4 & -5 & 0 \\ 1 & -3 & 6 \end{bmatrix} \quad b_3 = \begin{bmatrix} -2 \\ 5 \\ 6 \end{bmatrix}$$

9) Escribir para el método de Jacobi: las condiciones de inicialización; la fórmula de recurrencia; los posibles controles de detención y la actualización de variables.

10) Escribir para el método de Gauss-Seidel: las condiciones de inicialización; la fórmula de recurrencia; los posibles controles de detención y la actualización de variables.

Ejercicios de Sistemas de Ecuaciones Lineales con MATLAB

Se busca poder implementar los algoritmos iterativos de Jacobi y Gauss Seidel en MATLAB. Para ello son de utilidad algunos algoritmos que permiten el uso de vectores y matrices.

11) Escribir un programa en MATLAB que permita realizar el producto de una matiz por un vector, que deben ser previamente leídos.

```
% prod_mat_vec.m es un programa que multiplica una matriz por un vector y entrega el resultado
clear
clc
a=input('Ingrese n° filas de la matriz A: ');
b=input('Ingrese n° columnas de la matriz A: ');
c=input('Ingrese n° filas del vector B: ');
if (b~=c)
    error('el n° de columnas de A debe coincidir con el n° de filas de B');
end
for i=1:a
    for j=1:b
        A(i,j)=input('Ingrese componente de la matriz A: ');
    end
end

for j=1:b
        B(j)=input('Ingrese componente del vector B: ');
end

for i=1:a
    C(i)=0;
    for j=1:b
        C(i)=C(i)+A(i,j)*B(j);
    end
end
disp(‘ la matriz dato es:’), A
disp(‘ el vector dato es:’), B'
disp(‘ el vector respuesta es:’), C'
```

Para el programa anterior considere:

- ¿Cuál es la función del primer if…end if?
- La matriz A ¿se ingresa por filas o por columnas?
- En los últimos comandos disp. ¿Por qué se pone la “comilla” en B’ y C’?

12) Escribir un programa que en una variable guarde el valor absoluto de la máxima diferencia entre las componentes de dos vectores datos (norma infinito de la diferencia de dos vectores).

13) Escribir un ´programa que a partir de una matiz cuadrada A(i,j) dada, genere una nueva matriz T(i,j) del mismo orden que la anterior con las siguientes propiedades: T(i,j) es nulo si i es igual a j; y de lo contrario, T(i,j) es igual a (–A(i,j)/A(i,i)).

14) Escribir un programa en MATLAB que permita resolver un Sistema de Ecuaciones Lineales con el método de Jacobi.

Una posibilidad es la siguiente

```
%SEL_Jacobi_00.m
clc
clear
N=4
tol=2e-5
A=[ 50   -25    0     0
   -25    50   -25    0
    0    -25    50    -25
    0     0    -25    50]
b= [10 ; 20; 20; 10]
% INICIALIZACION DE VARIABLES
x=zeros(N,1);       % inicializa x con una matriz Nx1 llena de ceros
er=1000;            % inicializa error absoluto
it=0;               % inicializa el contador de iteraciones
while (er>tol)
    it = it +1;
    for i=1:N
        sum=0;          % inicializa acumulador de suma
        for j=1:N
            if (j~=i)
                sum = sum + A(i,j)*x(j);
            end
        end
        xn(i)=(b(i)-sum)/A(i,i);
        e(i)=abs(xn(i)-x(i));
    end
    for i=1:N           % Actualiza el vector aproximación de la solución
       x(i)=xn(i);
    end
    er=max(e);          % calcula la norma infinita del vector e
end

disp('El vector x es'), x
disp('El error es'), er
disp('La iteraciones son:'), it
```

- Identifique las líneas de código que constituyen el bloque "iterativo" propiamente dicho, y su criterio de detención
- Identifique las líneas de código que constituyen la "recurrencia" (generación de soluciones aproximadas)
- Identifique las líneas de código donde se evalúa el error. ¿Podría estar en orto lugar?. Proponga una modificación asociada.
- Justifique porque el `if (j~=i)`
- En el programa anterior, además de la condición de detención existente, incluya que el número máximo de iteraciones sea 1000 iteraciones.
- En el programa anterior reemplace `er=max(e)` por el cálculo de la norma infinito programado en detalle sin hacer uso del comando de MATLAB max()
- En el programa anterior reemplace la matriz y el vector datos por un ingreso de las mismas desde teclado.

15) Escribir un programa en MATLAB que permita resolver un Sistema de Ecuaciones Lineales con el método de Gauss Seidel.

Una posibilidad es la siguiente

```
% SEL_GaussSeidel_01.m
clc
clear
N=4
tol=2e-5
itmax=200
A=[ 50  -25   0    0
   -25   50  -25   0
    0   -25   50   -25
    0    0   -25   50]
b= [10 ; 20; 20; 10]
% INICIALIZACION DE VARIABLES
x=zeros(N,1);       % inicializa x con una matriz Nx1 llena de ceros
er=1000;            % inicializa error absoluto
it=0;               % inicializa el contador de iteraciones
while (er>tol & it<itmax)
    it = it +1;
    for i=1:N
        sum=0;          % inicializa acumulador de suma
        for j=1:N
            if (j~=i)
                if (j<i)
                sum = sum + A(i,j)*xn(j);
                end
                if(j>i)
                sum = sum + A(i,j)*x(j);
                end
            end
        end
        xn(i)=(b(i)-sum)/A(i,i);
        e(i)=abs(xn(i)-x(i));
    end
    for i=1:N           % Actualiza el vetor aproximacion de la solucion
       x(i)=xn(i);
    end
    er=max(e);          % calcula la norma infinita del vector e
end

disp('El vector x es'), x
disp('El error es'), er
disp('La iteraciones son:'), it
```

- Indique cuales son las líneas correspondientes al bloque iteartivo porpiamente dicho, y su criterio de detención.
- Analice y justifique los (if…end) dentro del bloque (for j=1:N….end). ¿Qué sucede si j=i?

El siguiente código es una versión "optimizada" del anterior donde se ha modificado el bloque iterativo propiamente dicho. Particularmente se ha modificado el producto de la matriz A por el vector x y la actualización de éste.

Analice en bloque iterativo propiamente dicho:

- ¿Qué ocurre cando j=i?
- ¿Dónde está ubicada la actualización de la x(i) por xn(i)?
- ¿Podría ubicarse dicha actualización antes de evaluar el error e(i)?. Justifique
- Identifique la posición del acumulador de la sumatoria "sum".

```
%SEL_Gauss_Seidel_01_optimizado.m
clc
clear
N=4
tol=2e-5
itmax=200
A=[ 50  -25   0    0
   -25   50  -25   0
    0   -25   50   -25
    0    0   -25   50]
b= [10 ; 20; 20; 10]
% INICIALIZACION DE VARIABLES
x=zeros(N,1);       % inicializa x con una matriz Nx1 llena de ceros
er=1000;            % inicializa error absoluto
it=0;               % inicializa el contador de iteraciones
while (er>tol & it<itmax)
    it = it +1;
    for i=1:N
        sum=0;          % inicializa acumulador de suma
        for j=1:N
            if (j~=i)
              sum = sum + A(i,j)*x(j);
            end
        end
        xn(i)=(b(i)-sum)/A(i,i);
        e(i)=abs(xn(i)-x(i));
        x(i)=xn(i);    % Actualiza el vector aproximacion de la solucion
    end
    er=max(e);         % calcula la norma infinita del vector e
end

disp('El vector x es'), x
disp('El error es'), er
disp('La iteraciones son:'), it
```

16) En el siguiente programa que implementa el método de Gauss Seidel se utiliza un vector s(j) como "acumulador" para la sumatoria en el producto de matrices.

Responder

- ¿ es necesario que sea un vector?
- ¿está correctamente ubicado?. Busque el justificativo de porque no funciona correctamente
- Ubique correctamente el acumulador

```
%%%%%%%%%%%sel_gseidel_00_con_error.m%%%%%%%%%%%%%%%%%%%%%%%%%%
clc
clear
for i=1:3
    for j=1:3
        a(i,j)=input ('componente de la matriz de coeficientes A: ');
    end
end
for j=1:3
    b(j)=input ('componente del vector b: ');
end
x=zeros(3,1)            % crea una matriz 3x1 llena de ceros: así inicializo x
s=zeros(3,1)            % inicializa acumulador
er=1000;                % inicializa error absoluto

while (er>.005)
    for i=1:3
        for j=1:3
            if (j~=i)
                s(i)=s(i)+a(i,j)*x(j)
            end
        end
        xn(i)=(b(i)-s(i))/a(i,i)
        e(i)=abs(xn(i)-x(i))
        x(i)=xn(i);                 % con esto es gauss-seidel y no jacobi
    end
    er=max(e)                       % calcula la norma infinita del vector e
end

disp('El vector x es'), x
disp('El error es'), er
```

Para el programa anterior considere:

- La matriz A ¿se ingresa por filas o por columnas?
- Justifique porque se considera `if (j~=i)`
- Justifique porque la línea `x(i)=xn(i)` ubicada en ese lugar cambia el algoritmo de Jacobi a Gauss Seidel
- En los últimos comandos disp. ¿se debe poner x' y er'?

17) Escribir un programa en MATLAB que permita resolver un Sistema de Ecuaciones Lineales con el método de Jacobi, utilizando la forma del algoritmo a modo de método de punto fijo, cuya fórmula de recurrencia se puede expresar en la forma $\mathbf{x}_{(k)} = \mathbf{T}\,\mathbf{x}_{(k-1)} + \mathbf{c}$

18) Modificar el programa en MATLAB anterior para que permita resolver un Sistema de Ecuaciones Lineales con el método de Gauss Seidel, expresado con fórmula de recurrencia como el método de punto fijo.

Guía de Estudio para Autovalores y Autovectores.

Contenido

Guía de Estudio para Autovalores y Autovectores. 1
Bibliografía recomendada: 1
Objetivo: 1
Definiciones 1
Ejercicios del Método de la Potencia y Potencia Inversa 2
Ejercicios teóricos 3
Ejercicios integradores 4

Bibliografía recomendada:

Notas de la Cátedra: Valores y Vectores Propios.
J. Mathews, J. Fink (2000). *Métodos Numéricos con Matlab.* Prentice Hall.
S. Nakamura (1997). *Análisis Numérico y Visualización Gráfica con MATLAB.* Prentice Hall
S. Nakamura (1992). *Métodos Numéricos Aplicados con Software.* Prentice Hall
S. Chapra, R. Canale (1999). *Métodos Numéricos para Ingenieros*, Mc Graw Hill.
R. Burden, J. Faires (1998). *Análisis Numérico*, International Thomson Editores.

Objetivo:

Obtener los valores propios y los vectores propios asociados, utilizando el método de la Potencia, también denominada método de iteración matricial.
Se busca poder implementarlos en planillas de cálculo a modo de Tabla de Cálculo y en programas de MATLAB.

Definiciones

Dada una matriz **A** se denominan autovalores λ y autovectores $\underline{x}$, a los números y vectores no nulos respectivamente que son solución del siguiente sistema de ecuaciones lineales

$$(\mathbf{A} - \lambda \cdot \mathbf{I})\underline{x} = \underline{0}$$

Alternativamente, se tiene el denominado problema generalizado de autovalores y autovectores

$$(\mathbf{K} - \lambda \cdot \mathbf{M})\underline{x} = \underline{0}$$

Los autovectores $\underline{x}$ definen direcciones invariantes de la transformación lineal asociada a la matriz A.

El método de la Potencia es un algoritmo iterativo que encuentra dicha dirección invariante, asociada al autovalor dominante.

Ejercicios del Método de la Potencia y Potencia Inversa

1) Dada la matriz $\mathbf{A} \in \mathrm{R}^{2x2}$, se busca determinar sus autovectores y sus autovalores.

$$\mathbf{A} = \begin{bmatrix} -10 & 4 \\ -4 & 0 \end{bmatrix}$$

a) Encuentre el valor propio dominante y su correspondiente vector propio empleando el Método de las Potencias. Utilice el algoritmo sin escalamiento y considere $\boldsymbol{\varepsilon} = 0{,}1$ como factor de tolerancia.
b) Encuentre el valor propio dominante y su correspondiente vector propio empleando el Método de las Potencias. Utilice el algoritmo con escalamiento y considere $\boldsymbol{\varepsilon} = 0{,}1$ como factor de tolerancia.
c) Encuentre el mínimo valor propio y su correspondiente vector propio, empleando el Método de las Potencias Inversas. Utilice el algoritmo con escalamiento y considere $\boldsymbol{\varepsilon} = 0{,}1$ como factor de tolerancia.
d) Represente en forma gráfica la curva de convergencia.

2) Dada la matriz $\mathbf{A} \in \mathrm{R}^{2x2}$, se busca determinar sus autovectores y sus autovalores.

$$\mathbf{A} = \begin{bmatrix} 200 & -1 \\ -1 & 2 \end{bmatrix}$$

b) Encuentre el valor propio dominante y su correspondiente vector propio empleando el Método de las Potencias. Utilice el algoritmo sin escalamiento y considere $\boldsymbol{\varepsilon} = 0{,}1$ como factor de tolerancia.
c) Encuentre el valor propio dominante y su correspondiente vector propio empleando el Método de las Potencias. Utilice el algoritmo con escalamiento y considere $\boldsymbol{\varepsilon} = 0{,}1$ como factor de tolerancia
d) Describa la ventaja de utilizar el algoritmo con escalamiento.

3) Dada la matriz $\mathbf{A} \in \mathrm{R}^{3x3}$, se busca determinar sus autovectores y sus autovalores.

$$\mathrm{A} = \begin{bmatrix} 4 & -1 & 1 \\ -1 & 3 & -2 \\ 1 & -2 & 3 \end{bmatrix}$$

a) Encuentre el valor propio dominante y el vector propio asociado aplicando el Método de las Potencias.
b) Encuentre el mínimo valor propio y el vector propio asociado aplicando el Método de la Potencia Inversa.
c) Proponga un algoritmo alternativo para resolver el inciso anterior que evite el cálculo de la matriz inversa.

4) Dada la matriz $\mathbf{A} \in \mathrm{R}^{3x3}$, se busca determinar sus autovectores y sus autovalores, usando el Método de la Potencia y Potencia Inversa.

$$\mathrm{A} = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 2 \end{bmatrix}$$

5) Dado el problema generalizado de valores propios, $(\mathbf{K}-\omega^2\mathbf{M})\ \underline{x} = \underline{0}$ hallar el mayor y el menor autovalor ω^2 y los autovectores $\underline{x}$ asociados. Comparar con el ejercicio anterior.

$$\left\{ \begin{bmatrix} 2 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 2 \end{bmatrix} - \omega^2 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 6 \end{bmatrix} \right\} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \end{Bmatrix}$$

Ejercicios teóricos

6) Demostrar que el algoritmo del Método de la Potencia converge siempre al valor propio dominante. Realizar la demostración con y sin escalamiento.

7) Si en el algoritmo de método de la Potencia, para una matriz **A** cuyos autovectores $\underline{v}_1, \underline{v}_2, ... \underline{v}_N$, y autovalores $\lambda_1, \lambda_2, ...\lambda_N$, resulta que para la iteración k se tiene

$$\underline{y}_k = \lambda_1^k \left[a_1 \underline{v}_1 + \left(\frac{\lambda_2}{\lambda_1}\right)^k a_2 \underline{v}_2 + \left(\frac{\lambda_3}{\lambda_1}\right)^k a_3 \underline{v}_3 + ... + \left(\frac{\lambda_N}{\lambda_1}\right)^k a_N \underline{v}_N \right]$$

Justificar que

$\lambda_1 = \lim\limits_{k \to \infty} \dfrac{\underline{y}_{k+1}(j)}{\underline{y}_k(j)}$ y $\lambda_1 = \lim\limits_{k \to \infty} \dfrac{\underline{y}_{k+1}(j)}{\underline{x}_k(j)}$ siendo j el índice que indica componente del vector

$\lambda_1 = \lim\limits_{k \to \infty} \dfrac{\left\langle \underline{y}_{k+1}, \underline{y}_{k+1} \right\rangle}{\left\langle \underline{x}_k, \underline{y}_{k+1} \right\rangle}$ $\quad \left\langle a, b \right\rangle$ indica el producto escalar entre los vectores *a* y *b*

8) Suponga que la matriz $\mathbf{A} \in \mathbb{R}^{3x3}$, tiene 3 autovalores distintos ($|\lambda_1| > |\lambda_2| > |\lambda_3|$) asociados a 3 autovectores linealmente independientes, v_1, v_2 y v_3, que usted no conoce. Explique qué sucedería si eligiera (sin saberlo) el vector v_1 como vector de arranque para aplicar el método de las potencias. ¿Y si eligiera (sin saberlo) el autovector v_2?

9) Comprobar que el sistema de EDO

$$\dot{\underline{\mathbf{y}}}(t) = \mathbf{A} \cdot \underline{\mathbf{y}}(t) \qquad \text{con} \qquad \mathbf{y}(0) = \begin{Bmatrix} 5 \\ 3 \end{Bmatrix}, \qquad \mathbf{A} = \begin{pmatrix} -10 & 4 \\ -4 & 0 \end{pmatrix}$$

admite como solución $\underline{\mathbf{y}(t)} = \underline{\mathbf{v}} \cdot e^{\lambda \cdot t}$ siendo $\underline{\mathbf{v}}$, λ la solución del siguiente sistema de autovalores y autovectores

$$\left(\begin{bmatrix} -10 & 4 \\ -4 & 0 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) \begin{Bmatrix} v_1 \\ v_2 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix}$$

La solución general será la combinación lineal $\underline{\mathbf{y}(t)} = \underline{\mathbf{v}_1} \cdot e^{\lambda_1 \cdot t} + \underline{\mathbf{v}_2} \cdot e^{\lambda_2 \cdot t}$, que cumple con las condiciones iniciales. Es decir, $\underline{\mathbf{y}(t)} = \dfrac{1}{3} \begin{Bmatrix} 1 \\ 2 \end{Bmatrix} \cdot e^{-2 \cdot t} + \dfrac{14}{3} \begin{Bmatrix} 1 \\ 0.5 \end{Bmatrix} \cdot e^{-8 \cdot t}$

10) El siguiente sistema de EDO homogéneo

$$16 \frac{d^2 \theta_1(t)}{dt^2} + 4(\theta_2 + \theta_1) = 0$$

$$32 \frac{d^2 \theta_2(t)}{dt^2} + 9\theta_1 = 0$$

Tiene como solución $\begin{Bmatrix} v_1 \\ v_1 \end{Bmatrix} e^{\lambda \cdot t} = \begin{Bmatrix} \theta_1(t) \\ \theta_1(t) \end{Bmatrix}$ siendo λ v_1 y v_2 soluciones del problema de valores propios asociado. Comprobar que la solución general es la combinación lineal $\underline{\theta(t)} = \underline{\mathbf{v}_1} \cdot e^{\lambda_1 \cdot t} + \underline{\mathbf{v}_2} \cdot e^{\lambda_2 \cdot t}$.
Con λ_1=0,418 y $\underline{\mathbf{v}_1}$ ={1; 0,6723} ; con λ_2= -1/5,95 y $\underline{\mathbf{v}_2}$ ={1; 1,6723}.

Ejercicios integradores

11) Con auxilio de la planilla de cálculo o un programa MATLAB verifique las siguientes propiedades.

    a) Si λ es autovalor de **A**, entonces λ^n es autovalor de $\mathbf{A}^n$, siendo n un entero positivo.
    b) Si λ es autovalor de **A**, entonces $\beta\,\lambda$ es autovalor de β**A**, siendo β un número real.

12) Cuando en un sistema de ecuaciones diferenciales ordinarios de la forma $\dot{\underline{\mathbf{y}}}(t) = \mathbf{A}\cdot\underline{\mathbf{y}}(t)$, se aplica el Método de Euler explícito se obtiene una forma algorítmica dada por:

$$\underline{\mathbf{y}}_{m+1} = \mathbf{G}_E \cdot \underline{\mathbf{y}}_m$$

$$\mathbf{G}_E = (\mathbf{I} + \Delta t\cdot\mathbf{A}) = \left(\begin{bmatrix} 1+\Delta t\cdot a & \Delta t\cdot b \\ \Delta t\cdot c & 1+\Delta t\cdot d \end{bmatrix}\right)$$

$$\mathbf{A} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

Proponer valores de Δt que hagan que el radio espectral (mayor de los autovalores tomados en valor absoluto) de la matriz $\mathbf{G_E}$ sea menor que uno. Considerar los coeficientes de la Matriz A como: a=-10; b=4; c=-4; d=0.

13) Considerar la siguiente ecuación diferencial ordinaria con valores en la frontera o condiciones de borde. Usando una aproximación central de la derivada segunda, y cinco puntos interiores, hallar el menor λ y una aproximación en forma discreta del u(x) asociado.

$$\frac{d^2u(x)}{dx^2} + \lambda\cdot u(x) = 0 \qquad si\ x\varepsilon\Omega = \{x\varepsilon R : 0 \le x \le 1\},$$
$$u(0) = 0, \qquad\qquad u(1) = 0.$$

Comparar con la solución exacta que es: $\lambda = \pi^2$ con *u(x)=sin(π.x)*.

14) La posición u(x,t) de una cuerda que está fija a los puntos x=0 y x=L y que está sometida a una fuerza T en sus extremos, se puede determinar resolviendo la siguiente ecuación diferencial a derivadas parciales

$$T\frac{\partial^2 u(x,t)}{\partial x^2} = \rho(x)\frac{\partial^2 u(x,t)}{\partial t^2} \qquad si\ x\varepsilon\Omega = \{x\varepsilon R : 0 \le x \le L\}$$
$$con\ \ u(0,t) = u(L,t) = 0 \quad \forall t$$

Demostrar que con la solución $u(x,t) = v(x)\cdot e^{I\omega t}$ siendo I la unidad imaginaria, y ω una constante a determinar, la ecuación a resolver resulta:

$$T\frac{d^2v(x)}{dx^2} + \omega^2\cdot\rho\cdot v(x) = 0 \qquad si\ x\varepsilon\Omega = \{x\varepsilon R : 0 \le x \le L\}$$
$$con\ \ v(0) = v(L) = 0$$

Usando una aproximación central de la derivada segunda, y cinco puntos interiores, hallar el menor ω^2 y una aproximación de *v(x)* en forma discreta, con L=12; T=500; ρ(x)=10. Asociar con *u(x,t)*.

15) La posición *u(x,t)* de una cuerda que está fija a los puntos x=0 y x=L y que está sometida a una fuerza T en sus extremos, se puede determinar resolviendo la siguiente ecuación diferencial a derivadas parciales

$$T\frac{\partial^2 u(x,t)}{\partial x^2}=\rho(x)\frac{\partial^2 u(x,t)}{\partial t^2} \qquad si\ x\varepsilon\Omega=\{x\varepsilon R: 0\le x\le L\}$$
$$con\ \ u(0,t)=u(L,t)=0 \quad \forall t$$

Plantear una solución usando derivación numérica y al menos cinco puntos interiores en el dominio **Ω**. Encontrar el sistema de ecuaciones lineales y resolver para L=12; T=500; ρ(x)=10.

16) Repetir para la siguiente función ρ(x):

X	0	2,0	4,0	6,0	8,0	10	12
ρ(x)	10	20	40	80	40	20	10

Comparar con el ejercicio anterior.

17) Para la siguiente ecuación diferencial,

$$-EA\frac{\partial^2 u(x,t)}{\partial x^2}+\rho A\frac{\partial^2 u(x,t)}{\partial t^2}=0 \qquad si\ x\varepsilon\Omega=\{x\varepsilon R: 0\le x\le L\}$$
$$u(0)=0$$
$$EA\frac{\partial u}{\partial x}(L)=-M\,\frac{\partial^2 u(L,t)}{\partial t^2}$$

Usando una aproximación central de la derivada segunda, y cinco puntos interiores, comprobar que se obtiene el siguiente sistema de EDO

$$\mathbf{M}\cdot\underline{\ddot{\mathbf{u}}}(t)+\mathbf{K}\cdot\underline{\mathbf{u}}(t)=\mathbf{0}$$

Comprobar que la solución $\underline{\mathbf{u}}(t)=\underline{\mathbf{v}}\cdot e^{I\omega t}$ siendo I la unidad imaginaria, y ω una constante a determinar, es solución del sistema EDO. Hallar el menor ω^2 y una aproximación de $\underline{\mathbf{v}}$ en forma discreta, con EA=210000; L=50000; ρA=7,85 y M=200.

18) La ecuación diferencial en coordenadas polares que determina la frecuencia natural de vibración libre ω de una membrana circular de radio interno ra y radio externo rb, sometida a una tracción uniforme T, densidad por unidad de área ρ, está dada por:

$$T\frac{d^2u(r)}{dr^2}+\frac{T}{r}\frac{du}{dr}+\omega^2\rho\cdot u(r)=0 \qquad si\ r\varepsilon\Omega=\{r\varepsilon R: ra\le r\le rb\}$$
$$con\ \ u(ra)=u(rb)=0$$

Plantear una solución usando derivación numérica y al menos cinco puntos interiores en el dominio **Ω**. Encontrar el sistema de ecuaciones lineales que debe resolverse.

19) La ecuación diferencial en coordenadas polares que determina la frecuencia natural de vibración libre ω de una membrana circular de radio externo rb, sometida a una tracción uniforme T, densidad por unidad de área ρ, está dada por:

$$T\frac{d^2u(r)}{dr^2}+\frac{T}{r}\frac{du}{dr}+\omega^2\rho\cdot u(r)=0 \qquad si\ r\varepsilon\Omega=\{r\varepsilon R: 0\le r\le rb\}$$
$$con\ \left.\frac{du}{dr}\right|_{r=0}=0 \qquad y\ \ u(rb)=0$$

Plantear una solución usando derivación numérica y al seis puntos interiores en el dominio **Ω**. Considerar derivada segunda y derivada primera con error del orden de h^2. Encontrar el sistema de ecuaciones lineales que debe resolverse. Obtener la menor frecuencia natural y compararla con l valor exacto dado por $\omega=(2{,}404/rb)\sqrt{T/\rho}$ que se puede consultar en Timoshenko, S., Young D.H. Weaver, W. (1988) "Vibration Problems in Engineering".

Guía de Estudio para Interpolación y Aproximación.

Contenido

Guía de Estudio para Interpolación y Aproximación.. 1
Bibliografía recomendada:.. 1
Objetivo:.. 1
Definiciones: 1
Ejercicios de Interpolación.. 2
Ejercicios de Interpolación con MATLAB 3
Ejercicios de Aplicaciones de Interpolación 10
Ejercicios de Aproximación por Mínimos Cuadrados 10
Ejercicios Teóricos... 15

Bibliografía recomendada:

Apuntes de la Càtedra.
J. Mathews, J. Fink (2000). *Métodos Numéricos con Matlab.* Prentice Hall.
S. Nakamura (1997). *Análisis Numérico y Visualización Gráfica con MATLAB.* Prentice Hall
S. Nakamura (1992). *Métodos Numéricos Aplicados con Software.* Prentice Hall
S. Chapra, R. Canale (1999). *Métodos Numéricos para Ingenieros*, Mc Graw Hill.
R. Burden, J. Faires (1998). *Análisis Numérico*, International Thomson Editores.

Objetivo:

Obtener polinomios de interpolación usando como base polinomios elementales, de Lagrange y de Newton.

Comprender la idea de Error de interpolación para funciones polinómicas y no polinómicas.

Obtener aproximaciones de Mínimos Cuadrados con bases polinómicas, trigonométricas y exponenciales.

Comprender la diferencia entre interpolación y aproximación por mínimos cuadrados

Implementar los métodos en planillas de cálculo y en programas de MATLAB

Definiciones:

Un conjunto de funciones base es un conjunto de funciones linealmente independientes; es decir, que no se obtiene una de ellas como combinación lineal de las demás.

Los residuos son las diferencias entre los valores de la función discreta dada y la función de interpolación o aproximación medida en las abscisas datos.

Un polinomio de interpolación, de colocación o polinomio interpolante es el obtenido al asegurar que los residuos en los puntos datos son nulos.

Una función de aproximación por mínimo cuadrados asegura que la norma cuadrática de los residuos es mínima, para esos datos en la base utilizada.

Ejercicios de Interpolación

1) Para la función discreta y=f(x) definida con los siguientes dos puntos

x	3	7
y	5	-1

- Hallar el polinomio interpolante con polinomios elementales
- Hallar el polinomio interpolante con polinomios de Lagrange
- Hallar el polinomio interpolante con polinomios de Newton
- Evalúe los polinomios interpolantes en x=3,1416 y x=9, y compare los resultados
- Discutir y justificar las diferencias (si es que hay) entre los valores obtenidos y los polinomios obtenidos.

2) Para la función discreta y=f(x) definida con los siguientes dos puntos

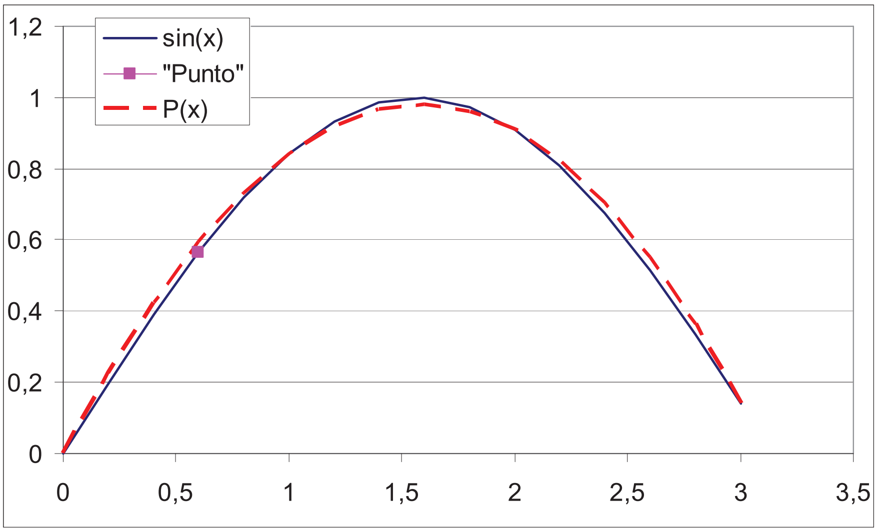
x	3	7	9
y	5	-1	2

- Hallar el polinomio interpolante con polinomios elementales
- Hallar el polinomio interpolante con polinomios de Lagrange
- Hallar el polinomio interpolante con polinomios de Newton
- Evalúe los polinomios interpolantes en x=3,1416 y x=9, y compare los resultados
- Discutir y justificar las diferencias (si es que hay) entre los valores obtenidos y los polinomios obtenidos.
- Comparar con los cálculos realizados en el ejercicio anterior y discutir ¿Qué método conviene elegir y por qué? para realizar la menor cantidad de cálculos ya que sólo se agrego un punto nuevo.

3) Buscar el polinomio interpolante que se obtiene de considerar una versión discreta de la función seno de x.

X (rad)	0	1	2	3
Sen(x)	Sen(0)	Sen(1)	Sen(2)	Sen(3)

a) Obtener los polinomios de Lagrange asociados a cada x dato
b) Obtener el polinomio interpolante.
c) Evaluar el polinomio interpolante en x=0,6 y evaluar el error absoluto (e=P(0,6)-Sen(0,6))
d) Identificar los puntos en que el error es cero.
e) Verificar que la grafica de ambas funciones es la siguiente.



f) ¿Qué ocurrirá con el error en x=0,6 si se usa un polinomio interpolante pero con valores de x=0; 0,5; 1,0 y 1,5?.

g) ¿Qué ocurrirá con el error en x=0,6 si se usa un polinomio interpolante pero con valores de x=0; 0,25; 0,5 y 1,0?.

Ejercicios de Interpolación con MATLAB

Se busca poder implementar los algoritmos de Interpolación en MATLAB.

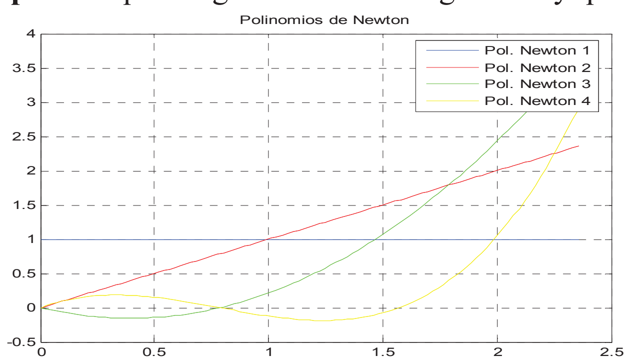
4) Escribir un programa en MATLAB que dadas las cuatro abscisas que están en las componentes del vector x=[0 1 2 3], grafique los polinomios de Newton asociados a dichas abscisas.

Una forma de resolver es con el siguiente código

%Pol_Newton_grafica

```
function pol_new
% Generación de los Polinomios de Newton
% usando 4 puntos;
clc,clear
% ABSCISAS y Función discreta DATOS
x=[0 pi/4 pi/2 3*pi/4]          % se ponen los puntos datos en un vector x
% GRAFICACIÓN
NP=100;                %Asignción del número de puntos
DX=(x(4)-x(1))/NP;     %Cálculo del incremento en x para graficar
xg=x(1):DX:x(4);       %asignación de las abscisas a graficar
% CALCULO DEl POLINOMIO DE INTERPOLACION para Graficar
N=length(xg);                       %Obtiene la dimensión del vector xg
 for j=1:N
   n1(j)=1;                         %Obtinen el polin de Newton n1
   n2(j)=n1(j)*(xg(j)-x(1));        %Obtinen el polin de Newton n2
   n3(j)=n2(j)*(xg(j)-x(2));        %Obtinen el polin de Newton n3
   n4(j)=n3(j)*(xg(j)-x(3));        %Obtinen el polin de Newton n4
 end
%GRAFICA DE las funciones
figure (1)         % se grafica cada función con un comando distinto
plot(xg,n1,'b')    % gráfica del Polin. de Newton 1
hold on
plot(xg,n2,'r')    % gráfica del Polin. de Newton 2
hold on
plot(xg,n3,'g')    % gráfica del Polin. de Newton 3
hold on
plot(xg,n4,'y')    % gráfica del Polin. de Newton 4
hold on
title ('Polinomios de Newton')
legend('Pol. Newton 1', 'Pol. Newton 2', 'Pol. Newton 3', 'Pol. Newton 4')
grid on
hold on
%
figure (2)        % se grafican con un sólo comando
plot(xg, n1,'b', xg,n2,'r', xg,n3,'g', xg,n4,'y')
title ('Polinomios de Newton')
legend('Pol. Newton 1', 'Pol. Newton 2', 'Pol. Newton 3', 'Pol. Newton 4')
grid on
end
```

Comprobar que las gráficas son las siguientes y que hay dos formas alternativas de generarlas.



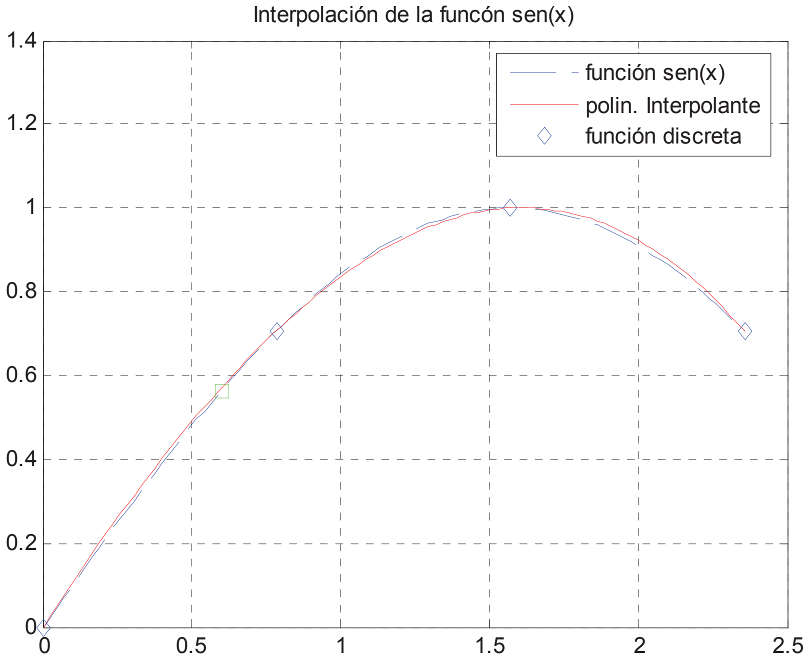
5) Escribir un programa en MATLAB que dadas los cuatro puntos cuyas abscisas están en las componentes del vector x=[0 pi/4 pi/2 3*pi/4]
y las ordenadas en el vector f=[sin(0) sin(pi/4) sin(pi/2 sin(3*pi/4)] permita:
- calcular la Tabla de Diferencias Divididas de Newton
- calcular y graficar el polinomio interpolante en el rango de abscisas datos

En el siguiente archivo se presentan una forma posible que resuelven el ejercicio.
%Pol_Newton_interpola_00.m

```
function interpola
% Generación de los Polinomios de Lagrange
% usando 4 puntos;
clc,clear
% ABSCISAS y Función discreta DATOS
x=[0 pi/4 pi/2 3*pi/4]    % se ponen los puntos datos en un vector x
f=sin(x)                  % función discreta obtenida de sin(x)
xm=0.6                    % Abscisa donde se mide el error
% CALCULO DE LAS DIFERENCIAS DIVIDIDAS DE NEWTON
% que se asignan en una Matriz Df(i,j)
% donde la columna j tiene las Dif Div de orden j
Df=zeros(4,3);
% Diferencias Divididas de Primer orden
Df(2,1)=(f(2)-f(1))/(x(2)-x(1));
Df(3,1)=(f(3)-f(2))/(x(3)-x(2));
Df(4,1)=(f(4)-f(3))/(x(4)-x(3));
% Diferencias Divididas de Segundo orden
Df(3,2)=(Df(3,1)-Df(2,1))/(x(3)-x(1));
Df(4,2)=(Df(4,1)-Df(3,1))/(x(4)-x(2));
% Diferencias Divididas de Tercer orden
Df(4,3)=(Df(4,2)-Df(3,2))/(x(4)-x(1));
% Coeficientes del Polinomio de Interpolación con Newton
a(1)=f(1);
a(2)=Df(2,1);
a(3)=Df(3,2);
a(4)=Df(4,3);
%CALCULO DEL POLINOMIO DE INTERPOLACION en Punto a Medir Error
 ne1=1;                   %Obtine el polin de Newton n1
 ne2=ne1*(xm-x(1));       %Obtine el polin de Newton n2
 ne3=ne2*(xm-x(2));       %Obtine el polin de Newton n3
 ne4=ne3*(xm-x(3));       %Obtine el polin de Newton n4
 % Obtiene el Polinomio de Interpolación con polin de Newton
 Pm=a(1)*ne1+a(2)*ne2+a(3)*ne3+a(4)*ne4;
 % Obtiene el sen(x)en el punto a medir el error
 ym=sin(xm);
 Error=abs(sin(xm)-Pm)
  ym
  Pm
% GRAFICACIÓN
NP=100;                %Asigngción del número de puntos
DX=(x(4)-x(1))/NP;     %Cálculo del incremento en x para graficar
xg=x(1):DX:x(4);       %asignación de las abscisas a graficar
% CALCULO DE LA FUNCIÓN CONTINUA A GRAFICAR
yg=sin(xg);
% CALCULO DEl POLINOMIO DE INTERPOLACION para Graficar
N=length(xg);                      %Obtiene la dimensión del vector xg
 for j=1:N
   n1(j)=1;                        %Obtinen el polin de Newton n1
   n2(j)=n1(j)*(xg(j)-x(1));       %Obtinen el polin de Newton n2
   n3(j)=n2(j)*(xg(j)-x(2));       %Obtinen el polin de Newton n3
   n4(j)=n3(j)*(xg(j)-x(3));       %Obtinen el polin de Newton n4
   % Obtiene el Polinomio de Interpolación con Polin de Newton
   P(j)=a(1)*n1(j)+a(2)*n2(j)+a(3)*n3(j)+a(4)*n4(j);
```

```
 end

figure (1)        % se grafican con un sólo comando
plot(xg, n1,'b', xg,n2,'r', xg,n3,'g', xg,n4,'y')
title ('Polinomios de Newton)
legend('Pol. Newton 1', 'Pol. Newton 2', 'Pol. Newton 3', 'Pol. Newton 4')
grid on
%
figure (2)
plot(xg,yg,'--b', xg,P,'r', x,f,'db', xm,ym,'sg')
title('Interpolación de la funcón sen(x)')
legend('función sen(x)', 'polin. Interpolante', 'función discreta')
grid on
end
```

Comprobar que la gráfica que se genera es la siguiente



Analizar el error para valores de xm 0.5; 3 y 4. Calcular el error relativo al valor exacto.
Notar que el polinomio interpolante con polinomios de Newton requiere las mismas líneas de código cada vez que se lo calcula en una abscisa, ya sea ésta xg(j) o xm. Es por ello que resulta conveniente definir una function que realice esa operación con los argumentos apropiados. Una posibilidad es la siguiente function.

```
function y=new(x,abscisa,k)
  y=1;
  if (k>1)
      for i=2:k
         y=y*(abscisa-x(i-1));
      end
  end
end
```

Notar que el cálculo del polinomio de interpolación usando polinomios de Newton en una abscisa xm cualquiera se puede realizar con

```
Pm=0;
 for k=1:4;
   Pm=Pm+a(k)*new(x,xm,k);
 end
```

sabiendo que en a(k) están los valores de las coeficientes obtenidos con Diferencias Divididas.
Modificar el Pol_Newton_interpola_00.m para usar la function anterior.

Discutir la posibilidad de hacer las diferencias divididas de primer orden para NP puntos con

```
for k=2:NP
  Df(k,1)=(f(k)-f(k-1))/(x(k)-x(k-1))
end
```

Discutir la posibilidad de hacer las diferencias divididas superiores al primer orden para NP puntos con

```
for m=2:NP-1
   for k=m+1:NP
      Df(k,m)=(Df(k,m-1)-Df(k-1,m-1))/(x(k)-x(k-m))
   End
end
```

6) Interpretar que tarea realiza el siguiente código MATLAB

```
function interpola
  clc,clear
  NP= 4
  x=[0 pi/4 pi/2 3*pi/4];
  f=sin(x)
  xm=0.6
  Df=zeros(NP,NP-1);
  for k=2:NP
   Df(k,1)=(f(k)-f(k-1))/(x(k)-x(k-1));
  end
  for m=2:NP-1
    for k=m+1:NP
         Df(k,m)=(Df(k,m-1)-Df(k-1,m-1))/(x(k)-x(k-m));
    end
  end
  a(1)=f(1);
  for k=2:NP
    a(k)=Df(k,k-1);
  end
  Pm=0;
  for k=1:4;
    Pm=Pm+a(k)*new(x,xm,k);
  end
  ym=sin(xm);
  Error=abs(sin(xm)-Pm)
  ym
  Pm
  NP=100;
  DX=(x(4)-x(1))/NP;
  xg=x(1):DX:x(4);
  yg=sin(xg);
  N=length(xg);
  for j=1:N
     P(j)=0;
     for k=1:4
         P(j)=P(j)+a(k)*new(x,xg(j),k);
     end
  end
  figure (1)
  plot(xg,yg,'--b', xg,P,'r', x,f,'db', xm,ym,'sg')
end
function y=new(x,abscisa,k)
  y=1;
  if (k>1)
      for i=2:k
         y=y*(abscisa-x(i-1));
      end
  end
end
```

7) Escribir un programa en MATLAB que dadas cuatro abscisas permita graficar los polinomios de Lagrange asociados y graficarlos.
   En MATLAB existe el comando "**plot(x,y)**" que realiza la gráfica del contenido de **los vectores x** e **y** en un sistema cartesiano. De modo que para graficar una función conocida se la debe evaluar en un conjunto de abscisas dadas que son las componentes de un vector xg; y el valor obtenido asignarlo en las componentes de otro vector yg. Luego usar el comando Plot con xg e yg.

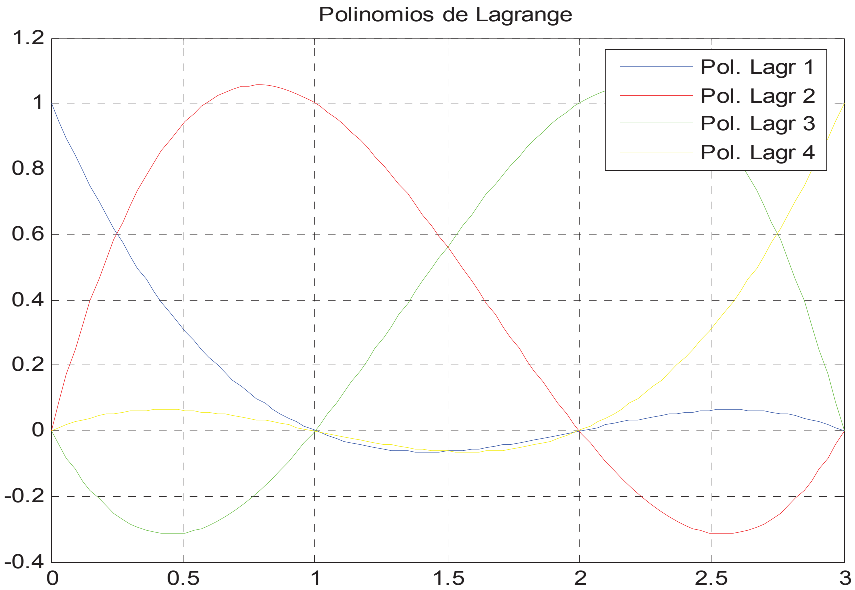
En el siguiente archivo se presentan 5 function (interpola, la_1, la_2, la_3, y la_4) que resuelven el ejercicio.

%Pol_Lagrange_grafica.m

```
function interpola
% Generación de los Polinomios de Lagrange
% usando 4 puntos;
  clc,clear
% ABSCISAS DATO
  x=[0 1 2 3]          %se ponen los puntos datos en un vector x
  NP=100               %Asigancíon del número de puntos
  DX=(x(4)-x(1))/NP;   %Cálculo del incremento en x para graficar
  xg=x(1):DX:x(4);     %asiganación de las abscisas a graficar
%
%% CALCULO DEl POLINOMIO DE INTERPOLACION para Graficar
  N=length(xg);               %Obtiene la dimensión del vector xg
% para cada componente del vector xg se calculan las componentes
% de cuatro vectores l1, l2, l3, l4 asociado a cada Pol. De Lagrange
  for j=1:N
     % Polinomio de Lagrange l1
      l1(j)=la_1(x,xg(j));      %Uso de function la_1
     % Polinomio de Lagrange l2
      l2(j)=la_2(x,xg(j));      %Uso de function la_2
     % Polinomio de Lagrange l3
      l3(j)=la_3(x,xg(j));      %Uso de function la_3
     % Polinomio de Lagrange l4
      l4(j)=la_4(x,xg(j));      %Uso de function la_4
  end
%
%GRAFICA DE las funciones
  figure (1)         % se grafica cada función con un comando distinto
  plot(xg,l1,'b')    % gráfica del Polin. de Lagrange 1 en Blue
  hold on
  plot(xg,l2,'r')    % gráfica del Polin. de Lagrange 2 en Red
  hold on
  plot(xg,l3,'g')    % gráfica del Polin. de Lagrange 3 en Green
  hold on
  plot(xg,l4,'y')    % gráfica del Polin. de Lagrange 4 en Yelow
  hold on
  title ('Polinomios de Lagrange')
  legend('Pol. Lagr 1', 'Pol. Lagr 2', 'Pol. Lagr 3', 'Pol. Lagr 4')
  grid on
  hold on
%
  figure (2)        % se grafican con un sólo comando
  plot(xg, l1,'b', xg,l2,'r', xg,l3,'g', xg,l4,'y')
  title ('Polinomios de Lagrange')
  legend('Pol. Lagr 1', 'Pol. Lagr 2', 'Pol. Lagr 3', 'Pol. Lagr 4')
  grid on
end
function y=la_1(x,abscisa)
  y=1;
  y=y*(abscisa-x(2))/(x(1)-x(2));
  y=y*(abscisa-x(3))/(x(1)-x(3));
  y=y*(abscisa-x(4))/(x(1)-x(4));
end
```

```
function y=la_2(x,abscisa)
    y=1;
    y=y*(abscisa-x(1))/(x(2)-x(1));
    y=y*(abscisa-x(3))/(x(2)-x(3));
    y=y*(abscisa-x(4))/(x(2)-x(4));
end
function y=la_3(x,abscisa)
    y=1;
    y=y*(abscisa-x(1))/(x(3)-x(1));
    y=y*(abscisa-x(2))/(x(3)-x(2));
    y=y*(abscisa-x(4))/(x(3)-x(4));
end
function y=la_4(x,abscisa)
    y=1;
    y=y*(abscisa-x(1))/(x(4)-x(1));
    y=y*(abscisa-x(2))/(x(4)-x(2));
    y=y*(abscisa-x(3))/(x(4)-x(3));
end
```

Las graficas obtenidas son:

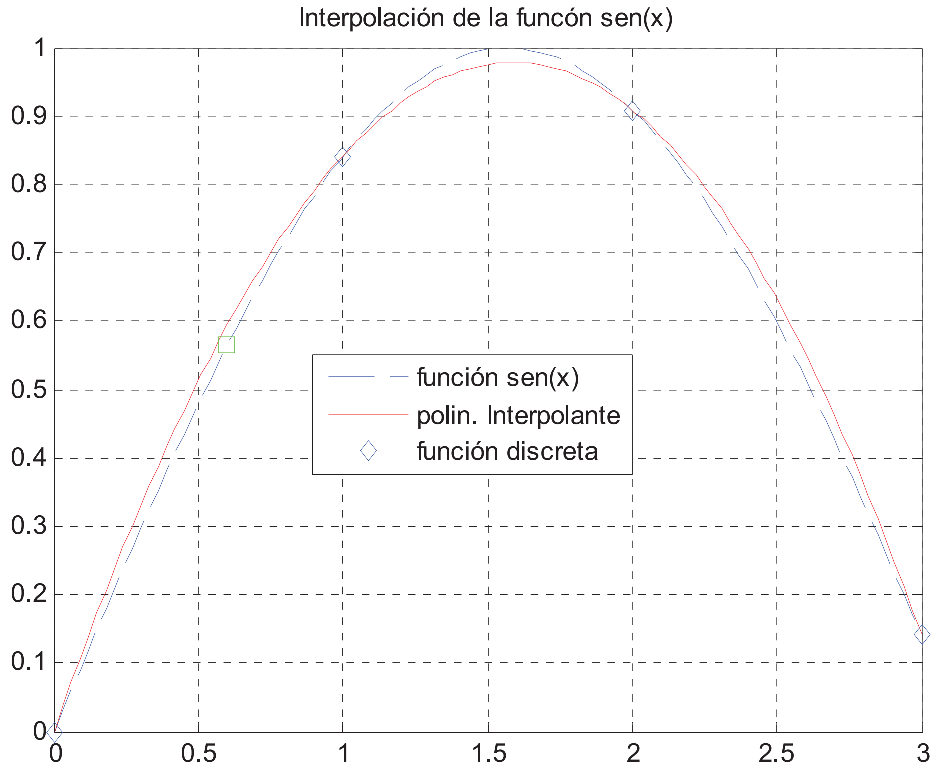


Para el programa anterior (%Pol_Lagrange_grafica.m) considere:

- Cambie las abscisas dato y vea el efecto que produce
- Cambie el número de puntos para graficar NP y vea el efecto que produce
- Cada polinomio de Lagrange está definido mediante una function con "variables genéricas" (abscisa e y). Al momento de usar las functions se asigna en abscisa alguna componente de xg; y el valor calculado con y se asigna en las componentes del vector que luego se grafica.
- Eliminar algún "hold on" al graficar la Figure (1) y analice su efecto.
- Notar que la Figure(1) y la Figure(2) son iguales, aunque con códigos distintos
- Cambiar la function la_4 de modo de usar la estructura algorítmica for-end

Modificar el programa anterior (%Pol_Lagrange_grafica.m) para que dadas 4 abscisas en un **vector x**,

- Calcule en un **vector f** el valor que toma en las abscisas dadas la función sen(x)
- Grafique la función sen(x) en el rango de las abscisas dadas
- Calcule el Polinomio interpolante para el conjunto de puntos definidos por los **vectores x** y **f**
- Grafique en una misma figura la función sen(x) y el polinomio interpolante
- Para los puntos dados por x=[0 1 2 3] y f=[sin(0) sin(1) sin(2) sin(3)] comprobar que el error de interpolación en la abscisa 0.6 es (sen(0.6)-Pol Interpola(0.6)=0.0296 y que la grafica del polinomio de Interpolación es la siguiente.



Ejercicios de Aplicaciones de Interpolación

8) Estime la población de una ciudad en el año 1983 a partir de censos realizados según la siguiente tabla:

Año	1960	1966	1971	1980	1990
Población (millones)	12	15	18	23	26

9) Interpole mediante la fórmula de Lagrange la función ln (x) a través de los puntos dados en la siguiente tabla:

i	0	1	2	3
x_i	1	2	3	4
y_i	0	0,6931472	1,0986123	1,3862944

a) Dé la expresión del polinomio $P_3(x)$.
b) Evalúe dicha expresión para x=2,5.
c) Coteje el valor obtenido con respecto al valor exacto ln (2,5) = 0,91629. Haga ln(2,5) - P_3(2,5).
d) Acote el error teórico en el intervalo mediante la expresión:

$$e(x) = (x - x_0)(x - x_1)(x - x_2)(x - x_3)\frac{\ln^{(iv)}(\varphi)}{4!}$$, donde φ es un punto interior al intervalo.

10) La función ln(cos x) resulta incómoda de manejar en ciertos procesos matemáticos, por lo que un investigador ha resuelto sustituirla por una expresión polinomial de tercer grado que sea suficientemente precisa en el intervalo abierto $0 < x < \pi/2$. Expresando x en grados ha resulto evaluar el ln (cos x) para 5º, 25º, 65º, 85º, obteniendo:

x	5º	25º	65º	85º
ln (cos x)	$-3.8126\cdot10^{-3}$	-0.09837	-0.86128	-2.440058

a) ¿Qué error comete en x = 30º y en x = 60º con respecto a los valores exactos?
b) Obtenga el polinomio $P_3(x)$ y calcule:
   b1) $e_{30º}$ = ln (cos 30º)- P_3(30).
   b2) $e_{60º}$ = ln (cos 60º)- P_3(60).

Ejercicios de Aproximación por Mínimos Cuadrados

11) Aproxime según el criterio de mínimos cuadrados, utilizando aproximación lineal, los siguientes puntos, cuyas coordenadas son:

x	0	1	2	3
y	1	1	2	4

Resolución:

Usando $\Phi^T\Phi\bar{a} = \Phi^T\bar{y}$ resulta:

$$\begin{bmatrix}1 & 1 & 1 & 1\\0 & 1 & 2 & 3\end{bmatrix}\begin{bmatrix}1 & 0\\1 & 1\\1 & 2\\1 & 3\end{bmatrix}\begin{bmatrix}a\\b\end{bmatrix} = \begin{bmatrix}1 & 1 & 1 & 1\\0 & 1 & 2 & 3\end{bmatrix}\begin{bmatrix}1\\1\\2\\4\end{bmatrix}$$

Que también en este caso particular se puede calcular como:

$$\begin{bmatrix} N & \sum_{i=1}^{4} x_i \\ \sum_{i=1}^{4} x_i & \sum_{i=1}^{4} x_i^2 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{4} y_i \\ \sum_{i=1}^{4} y_i x_i \end{bmatrix}$$

Que resulta

$$\begin{bmatrix} 4 & 6 \\ 6 & 14 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 8 \\ 17 \end{bmatrix}$$

y obtenemos **a = ½** y **b = 1** y así

y= ½ + x.

12) Encuentre, por el **método de los mínimos cuadrados**, la recta que aproxima la función dada por los siguientes puntos:

x	0	1	2	3	4	5	6
y	4	7	9	10	9	12	15

Resolución: Para resolver $\Phi^T \Phi \overline{a} = \Phi^T \overline{y}$ es posible plantear en forma auxiliar la siguiente tabla:

i	x	y	x^2	x^3	x·y
0	0	4	0	0	0
1	1	7	1	1	7
2	2	9	4	8	18
3	3	10	9	27	30
4	4	9	16	64	36
5	5	12	25	125	60
6	6	15	36	216	90
Sumat	21	66	91	441	241

Con los datos de la última fila de la tabla armamos el sistema de ecuaciones:

$$\Phi^T \Phi \overline{a} = \Phi^T \overline{y}$$

$$\begin{bmatrix} N & \sum_{i=0}^{6} x_i \\ \sum_{i=0}^{6} x_i & \sum_{i=0}^{6} x_i^2 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^{6} y_i \\ \sum_{i=0}^{6} y_i x_i \end{bmatrix}$$

$$\begin{bmatrix} 7 & 21 \\ 21 & 91 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 66 \\ 241 \end{bmatrix}$$

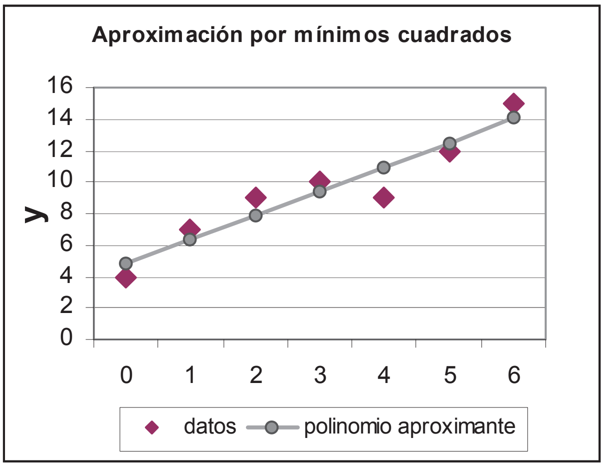
Resolviendo el sistema de ecuaciones (por ejemplo invirtiendo la primera matriz) se obtiene el siguiente resultado:

a = 4,821
b = 1,536

La ecuación de la recta será:

y = 4,821 + 1,536 x

Podemos verificar la calidad de la aproximación graficando las dos funciones:



13) Ubique un polinomio de segundo grado (parábola) de segundo grado entre los puntos siguientes, según el criterio de mínimos cuadrados.

x	0	1	2	2	3	3	4	4
y	1	1	1	2	2	3	4	5

Resolución:

Para este caso de una función polinómica de segundo grado (parábola), la ecuación será:

$$\mathbf{y = a + b \cdot x + c \cdot x^2}$$

con N = 8 y m = 2.

El sistema de ecuaciones será:

$$\begin{vmatrix} N & \sum x_i & \sum x_i^2 \\ \sum x_i & \sum x_i^2 & \sum x_i^3 \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^4 \end{vmatrix} \begin{vmatrix} a \\ b \\ c \end{vmatrix} = \begin{vmatrix} \sum y_i \\ \sum y_i \cdot x_i \\ \sum y_i \cdot x_i^2 \end{vmatrix}$$

y reemplazando queda:

$$\begin{vmatrix} 8 & 19 & 59 \\ 19 & 59 & 199 \\ 59 & 199 & 707 \end{vmatrix} \begin{vmatrix} a \\ b \\ c \end{vmatrix} = \begin{vmatrix} 19 \\ 58 \\ 202 \end{vmatrix}.$$

Resolviendo este sistema se obtiene:

a=1.0588 **b=-0.4657** **c=0.3284**

y la parábola es $\mathbf{y = 1.0588 - 0.4657\, x + 0.3284\, x^2}$.

14) Aproximar por regresión lineal los siguientes datos:

x	0	1	2	3	4
y	0	2	2	1	3

15) Aproximar por regresión polinómica de segundo grado los siguientes datos:

x	0	0	1	2	3
y	1	2	2	3	3

16) Considerando como funciones bases a sen(x) y cos(x), halle la curva de la forma f(x) = a cos(x) + b sen(x) que mejor se ajusta al siguiente conjunto de puntos datos en el sentido de los mínimos cuadrados.

x	-3.0	-1.5	0.0	1.5	3.0
y	-0.1385	-2.1587	0.8330	2.2774	-0.5110

17) Considere las funciones f, g y h dadas en forma discreta en la siguiente tabla:

x	0	$\pi/4$	$\pi/2$	$3\pi/4$	π	$5\pi/4$	$3\pi/2$	$7\pi/4$	2π
f(x)	10	10	10	10	10	10	10	10	10
g(x)	10	0	0	0	10	10	10	0	10
h(x)	3	2.5355	2	1.7071	-1	-4.5355	-4	0.2929	3

Aplique el método de mínimos cuadrados para hallar funciones que aproximen a cada una de ellas, tomando como funciones bases

$\phi_0 = \text{sen}(x)$, $\phi_1 = \text{sen}(2x)$, $\phi_2 = \cos(x)$, $\phi_3 = \cos(2x)$.

18) Para la siguiente tabla de datos (x,y):

x	1.0	1.25	1.50	1.75	2.0
y	5.1	5.79	6.53	7.45	8.46

Encuentre los coeficientes a y b de la aproximación $y = b\,e^{a\,x}$, usando el método de mínimos cuadrados.

19) La expresión $P = a\,V^{\,b}$ relaciona la presión y el volumen de una máquina de vapor. Para los valores datos de la siguiente tabla, halle las constantes a y b, usando el método de mínimos cuadrados.

V	53.90	26.40	14.00	7.00	4.27	2.74	1.85
P	6.87	14.70	28.80	60.40	101.90	163.30	250.30

20) a) Halle el polinomio que ajusta los siguientes puntos por un método de interpolación a elección:

x	1	2	3
y	2	1	1.5

b) Para los mismos datos encuentre una aproximación lineal por mínimos cuadrados.

c) Evalúe los polinomios hallados en los incisos a) y b) en x = 10. Compare los resultados.

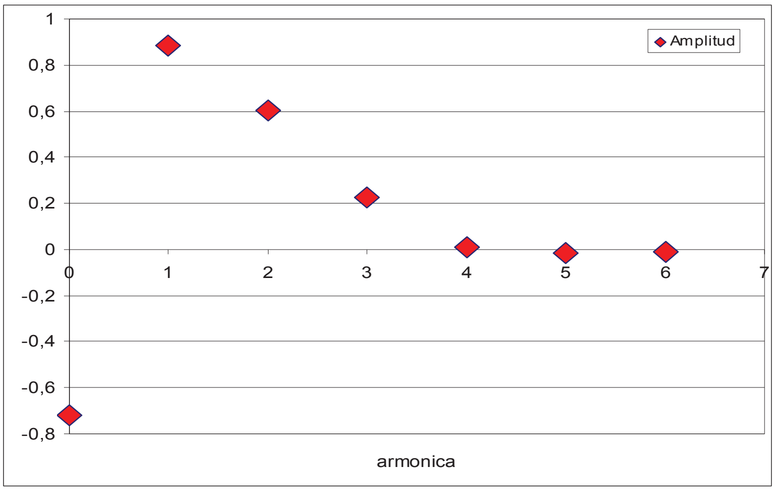
21) La presión del viento sobre una superficie cilíndrica está dada en forma de una función discreta por el Reglamento CIRSOC 102, que se reproduce en la siguiente Tabla.

Ang (grados)	presión
0	1,00
10	0,90
20	0,70
30	0,35
40	0,00
50	-0,50
60	-0,80
70	-1,10
80	-1,23
90	-1,30
100	-1,30
110	-1,28
120	-1,25
130	-1,20
140	-1,20
150	-1,20

Si la base considerada es:

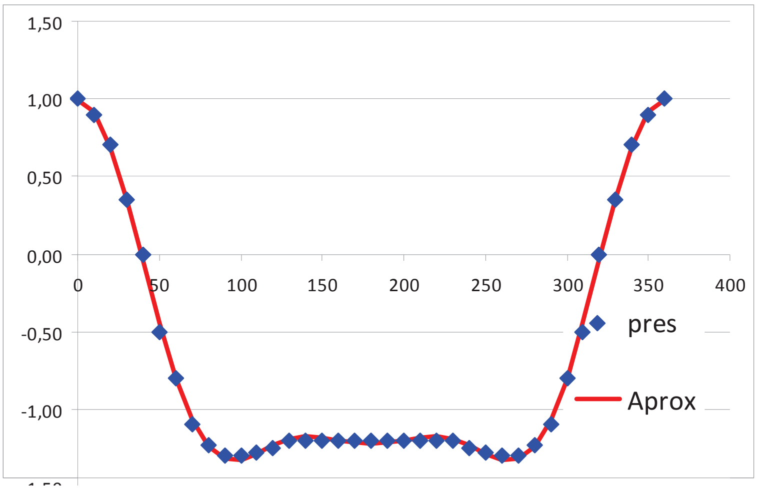
$\{\cos(0x); \cos(x); \cos(2x); \cos(3x); \cos(4x); \cos(5x); \cos(6x)\}$,

La aproximación de mínimos cuadrados tiene por coeficientes solución (amplitudes de la combinación lineal de las funciones base), los siguientes coeficientes:



160	-1,20
170	-1,20
180	-1,20
190	-1,20
200	-1,20
210	-1,20
220	-1,20
230	-1,20
240	-1,25
250	-1,28
260	-1,30
270	-1,30
280	-1,23
290	-1,10
300	-0,80
310	-0,50
320	0,00
330	0,35
340	0,70
350	0,90

Es evidente que los coeficientes asociados con las armónicas 4, 5 y 6 son despreciables frente a los demás.
Cuando se compara la función discreta con la Aproximación considerando sólo las amplitudes de las armónicas 0, 1, 2 y 3 se obtiene la siguiente gráfica.



Comprobar que Aproximación resulta cuando sólo se consideran las amplitudes de las armónicas 0, 1 y 2.

En vez de trabajar con la Tabla dada es posible trabajar con la aproximación dada por una base de cosenos {cos(k x)} con k=0,1,2,3 y Amplitudes Ak. Es decir por la siguiente Tabla

Armónica k	0	1	2	3
Amplitud Ak	-0,717222222	0,884863921	0,602500243	0,224616777

Ejercicios Teóricos

22) Considere dado un sistema de coordenadas cartesianas.
   a) Dados dos puntos (x_0, y_0) y (x_1, y_1), ¿cuántas rectas pasan por ambos? ¿Y cuántos polinomios de segundo grado (parábolas)?
   b) ¿Cuántos polinomios de segundo grado (parábolas) pasan por tres puntos no alineados dados?
   c) Dados tres o más puntos alineados, ¿es posible que pertenezcan todos a un polinomio de segundo grado (parábola)? ¿Y pueden pertenecer todos al gráfico de otra función polinómica?
   d) ¿Cuántos polinomios de grado n o menos pasan por n+1 puntos dados? ¿Y cuántos polinomios de grado mayor que n pasan por n+1 puntos dados?
   e) De acuerdo a lo anterior, ¿es posible obtener distintos polinomios al interpolar un conjunto de puntos dados si se trabaja con diferentes métodos (como el método directo, de Lagrange o de Newton)?

23) Obtener los tres primeros coeficientes del Polinomio de Interpolación usando como base los polinomios de Newton.

24) Para los siguientes conjuntos de puntos,
   a) decida si es o no posible ajustar un polinomio por algún método de interpolación. En caso de no ser posible, indique por qué y diga si se puede o no evitar ese inconveniente de alguna manera. En caso de ser posible, hállelo por algún método.

k	0	1	2
x_k	1	2	2
y_k	2	1	3

k	0	1	2
x_k	1	2	2
y_k	2	1	1

k	0	1	2	3
x_k	1	2	7	10
y_k	2	1	14	2

   b) ¿Podría hallar la recta que mejor aproxima esos puntos datos aplicando el método de mínimos cuadrados? ¿Presenta alguno de los conjuntos de puntos algún inconveniente?

Guía de Estudio para Integración Numérica.

Contenido

Guía de Estudio para Integración Numérica. 1
Bibliografía recomendada: 1
Objetivo: 1
Definiciones 1
Ejercicios de los Métodos de Simpson simple y de Simpson compuesto. 4
Ejercicios de Regla de extrapolación de Richardson y Método de Romberg. 4
Ejercicios de cuadratura de Gauss Legendre 7
Ejercicios con MATLAB 7
Ejercicios Teóricos 7
Ejercicios de aplicaciones 8
Ejercicios Integradores 12

Bibliografía recomendada:

Apuntes de la Càtedra.
J. Mathews, J. Fink (2000). *Métodos Numéricos con Matlab.* Prentice Hall.
S. Nakamura (1997). *Análisis Numérico y Visualización Gráfica con MATLAB.* Prentice Hall
S. Nakamura (1992). *Métodos Numéricos Aplicados con Software.* Prentice Hall
S. Chapra, R. Canale (1999). *Métodos Numéricos para Ingenieros*, Mc Graw Hill.
R. Burden, J. Faires (1998). *Análisis Numérico*, International Thomson Editores.

Objetivo:

Se pretende comprender a los Métodos para resolver Integrales Numéricas o CUADRATURAS y conocer la aplicación del procedimiento de extrapolación de Richardson.

Se busca poder implementarlos en planillas de cálculo y en programas de MATLAB.

Definiciones

Se entiende por *Regla de Integración Numérica o Cuadratura* a la combinación lineal de las ordenadas de la función discreta, que aproxima la integral definida de dicha función en un intervalo.

Una regla de integración numérica es la obtenida a partir a partir de la integración del polinomio de interpolación de una función discreta con un número determinado de puntos.

Cada regla de integración numérica tiene un error de integración asociado a su definición, que permite el control de la confianza que se tiene en la aproximación obtenida.

El *error de integración* siempre depende de una potencia de la separación de los datos y del valor de la derivada de la función, a la que se aplica la regla de integración, de orden mayor o igual a la cantidad de puntos usados en la definición de la regla de integración.

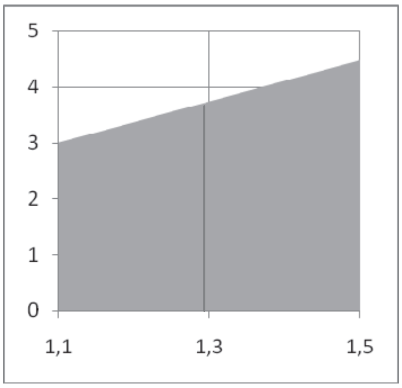
El *orden de la regla de integración* está dado el grado del mayor polinomio que integra en forma exacta, y ello corresponde a un grado menos que la máxima derivada que aparece en el error de integración.

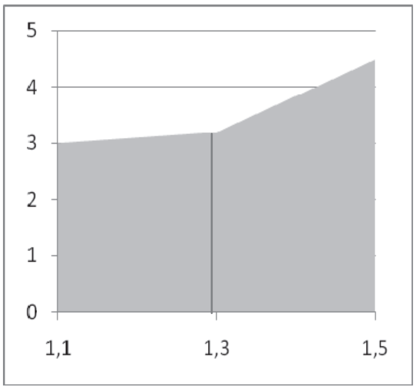
El *orden del error de la regla de integración* está dado por la potencia de la separación de los datos que figura en la expresión del error.

Ejercicios de los Métodos de Trapecios simple y Trapecios compuesto.

1) Dada la función f : **R**→ **R** en forma discreta, se busca determinar I = $\int_{1.1}^{1.5} f(x)\,dx$.

x	1.1	1.3	1.5
f(x)	3.0042	3.2	4.4817





a) ¿Cuál de los gráficos anteriores corresponde a una aplicación del método de trapecios simple y cuál a una aplicación del método de trapecios compuesto?
b) A continuación se han calculado las aproximaciones de la integral buscada por los métodos de trapecios simple y compuesto, obteniendo los siguientes valores:

xi	Yi
1.1	3.0042
1.5	4.4817

xi	yi
1.1	3.0042
1.3	3.2
1.5	4.4817

$$\begin{aligned} I_{TS} &= \frac{h_{TS}}{2}(y_0 + y_1) \\ &= \frac{0.4}{2}(3.0042 + 4.4817) \\ &= 1.49718 \end{aligned}$$

$$\begin{aligned} I_{TC} &= \frac{h_{TC}}{2}(y_0 + 2y_1 + y2) \\ &= \frac{0.2}{2}(3.0042 + 2 \cdot 3.2 + 4.4817) \\ &= 1.38859 \end{aligned}$$

c) ¿Cuál es el paso utilizado en cada método?
d) ¿A qué se debe el coeficiente 2 que aparece en el cálculo de la integral por el método de trapecios compuesto, I_{TC}?
e) La aplicación del método de trapecios simple anterior se puede pensar como el producto escalar de los vectores columna **y**=(3.0042, 4.4817)^T (la ( )^T indica transpuesto, ya que **y** es un vector columna) formado por las ordenadas de los puntos datos, y el vector $c = \left(\frac{h}{2}, \frac{h}{2}\right)^T$ de coeficientes.

   Indique cuáles son los vectores **y** y *c* para expresar la aproximación de la integral por el método de trapecios compuesto.
f) Justifique cuál de las dos aproximaciones anteriores elegiría usted.

2) Dada la función f en forma discreta en el intervalo de la variable x [0; 0.4], por la tabla siguiente,

x	0	0.1	0.2	0.3	0.4
f(x)	1	7	4	3	5

a) Escriba un programa en MATLAB que encuentre aproximaciones de la integral de f(x) en ese intervalo por los métodos de trapecios simple y compuesto.

b) Halle una aproximación de la integral de la función dada,

i) usando la regla de trapecios simple entre 0 y 0.4.
ii) Usando la regla de trapecios compuesta entre 0 y 0.4.
iii) ¿Cuál es el paso h utilizado en cada caso?

El archivo “integraltrapecios_0.m” presentado a continuación es una forma de resolver el ejercicio:

```
% Integral por trapecios simple o compuesto.
% En este programa el usuario ingresa los valores de h y de y manualmente.
% Para hacer trapecios simple, usar N=1 y trabajar sólo con dos puntos.
clc
clear
N=input('ingrese el n° de datos con los que va a trabajar: ');

%leer datos
h=input('ingrese el paso h: ');
y=zeros(N,1);       % inicializo el vector de ordenadas y

for i=1:N
    %CUIDADO: NO SE USA y0 SINO y1 COMO PRIMER ELEMENTO DEL VECTOR Y.
    y(i)=input(['ingrese la ordenada y(',num2str(i),'): ']);
    % num2str(i) convierte el valor numérico de variable i en un string
    % (un string es una cadena de caracteres, no un número)
    % "input" admite un vector de strings como argumento
end

% generación del vector de coeficientes
c=zeros(N,1);
c(1)=h*1/2;
c(N)=h*1/2;
for i=2:N-1
    c(i)=h*1;
end

% producto escalar
int=0;                  % inicializo la variable acumulador
for i=1:N
    int=int+c(i)*y(i);
end

disp('El vector de coeficientes es '),disp(c)
disp('El valor de la integral es '), disp(int)
```

c) En el programa anterior incorpore los comandos necesarios para graficar la función f(x).

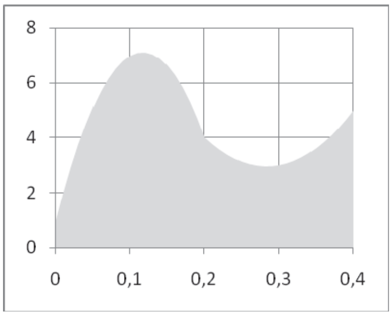
d) En el programa anterior unifique en un mismo for-end el ingreso de las ordenadas de la función y el cálculo del producto escalar con el que se calcula la integral.

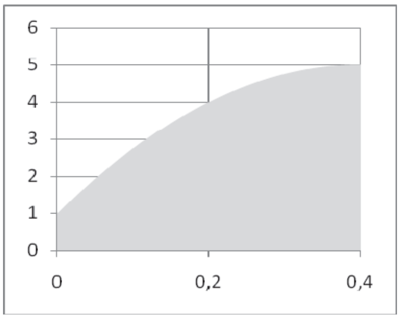
Ejercicios de los Métodos de Simpson simple y de Simpson compuesto.

3) Considere la función dada por los siguientes datos,

x	0	0.1	0.2	0.3	0.4
f(x)	1	7	4	3	5

a) ¿Cuál de los siguientes gráficos corresponde a una aplicación del método de Simpson simple y cuál a una aplicación del método de Simpson compuesto para aproximar $I = \int_0^{0.4} f(x)dx$?





b) Halle una aproximación de la integral de la función dada,
   i) Usando la regla de Simpson simple entre x=0 y x=0.4.
   ii) Usando la regla de Simpson compuesta entre x=0 y x=0.4.
   iii) ¿Cuál es el paso h utilizado en cada caso?
   iv) Justifique cuál de las aproximaciones anteriores es más precisa.

4) Modifique el programa integraltrapecios_0.m presentado anteriormente para que aproxime el valor de la integral por los métodos de Simpson simple y compuesto.

5) Dada la función f : **R**→ **R** en forma discreta,

x	0	0.2	0.4	0.6	0.8
f(x)	1	1.04	1.16	1.36	1.64

a) calcule $I = \int_0^{0.8} f(x)\,dx$ usando la regla de trapecios compuesta.

b) Calcule $I = \int_0^{0.8} f(x)\,dx$ usando la regla de Simpson compuesta.

c) Justifique cuál de las aproximaciones anteriores es más precisa.

Ejercicios de Regla de extrapolación de Richardson y Método de Romberg.

6) Dada la siguiente función y=f(x) en forma discreta, aplicando la regla de trapecios compuesta halle una aproximación de la integral de f(x) entre x_0 y x_4.

x_n=	0	1/4	1/2	3/4	1
y_n=	1	3	3	1.5	0

a) usando un paso h_1=0.5 (el intervalo se divide en 2 subintervalos).

b) Usando un paso h_2=0.25 (el intervalo se divide en 4 subintervalos).

c) Mejore la aproximación lograda en los incisos anteriores usando la Regla de Richardson que, a partir de dos valores aproximados con error de orden h^n, mejora la aproximación haciendo:

$$I(h_1);I(h_2) \text{ tienen orden de } O(h^n)$$

$$I(h^{n+2}) = \frac{\beta \cdot I(h_2) - I(h_1)}{\beta - 1} \quad con \quad \beta = \left(\frac{h_1}{h_2}\right)^n$$

7) Con los siguientes datos,

x	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
f(x)	1	1.3	1.5	1.55	1.61	1.62	1.65	1.68	1.69

a) halle un valor aproximado de la integral $I = \int_0^{0.8} f(x)\,dx$, usando la regla de Simpson compuesta, con un paso h de 0.1.
b) Halle un valor aproximado de la misma integral I, usando la regla de Simpson compuesta, con un paso h de 0.2.
c) Aplique la regla de Richardson a los dos valores obtenidos anteriormente, para conseguir un valor mejorado de I.
d) ¿Cuál de las tres aproximaciones de I es más precisa? Justifique su respuesta.

8) Con los datos siguientes,

x	0	0.1	0.2	0.3	0.4
f(x)	1	7	4	3	5

- Calcular la integral con Trapecios simple y un paso h=0.4
- Calcular la integral con Trapecios compuesto y un paso h=0.2
- Hallar una nueva aproximación de la integral aplicando la regla de Richardson. Cuidado: ¿qué exponente debe usarse para calcular beta?
- Calcular la integral con Trapecios compuesto y un paso h=0.1

En base a los cálculos realizados, se tienen tres aproximaciones de la integral que son las siguientes,

Paso h_i	Aproximación I_i
0.4	1.2
0.2	1.4
0.1	1.7

- Hallar una nueva aproximación de la misma integral, aplicando extrapolación de Richardson en forma reiterada (Romberg) e indicar el orden del error de la nueva aproximación.
  Para resolver este ejercicio puede ser conveniente completar una tabla como la siguiente:

h_i	$I_{1,j}$	Error	$I_{2,j}$ β=0.25	Error	$I_{3,j}$ β=0.0625	Error
h_1=0.4	1.2	$O(h_1^3)$	1.4666	$O(h_1^4)$	1.8222	$O(h_1^6)$
h_2=0.2	1.4	$O(h_2^2)$	1.8	$O(h_2^4)$		
h_3=0.1	1.7	$O(h_3^2)$				

Verificar los resultados presentados en la Tabla.

9) Para la función dada en la siguiente tabla,

x	0	0.1	0.2	0.3	0.4
f(x)	1	7	4	3	5

halle un valor aproximado de la integral utilizando el método de Romberg. Indique el orden del error para el resultado hallado.

10) Escriba un programa en MATLAB que halle dos aproximaciones de una integral, I1 e I2, por un mismo método (trapecios compuesto o Simpson compuesto), con pasos uno el doble del otro, y que conociendo el orden de error del método usado para hallar I1 e I2, encuentre una nueva aproximación de la integral de f en ese intervalo por medio de la regla de Richardson. Al final debe entregar el valor mejorado de la aproximación de la integral y el orden de error de esta aproximación.

Una forma de resolverlo es la siguiente:

```
% Este programa "Richardson.m" calcula dos Integrales por trapecios compuesto.
% En este programa el usuario ingresa los valores de h y de y manualmente.
% N es el número total de puntos que se va a usar con el paso h1 menor.
% Se calcula la integral usando h1 y 2h1 como pasos.
```

```
clc
clear
N=input('ingrese el n° total de datos con los que va a trabajar, que debe ser impar y
mayor o igual a 3: ');

%leer datos
h1=input('ingrese el menor paso h: ');
h2=h1*2;
y=zeros(N,1);       % inicializo el vector de ordenadas y

for i=1:N
    y(i)=input(['ingrese la ordenada y(',num2str(i),'): ']);
end

% genero los vectores de coeficientes para hallar I1 y I2
c1=zeros(N,1);
c1(1)=h1*1/2;
c1(N)=h1*1/2;
for i=2:N-1
    c1(i)=h1*1;
end
c2=zeros(N,1);
c2(1)=h2*1/2;
c2(N)=h2*1/2;
for i=3:2:N-2   %formo un vector que tiene ceros en las componentes pares
    c2(i)=h2*1;
end

% productos escalares
I1=0;
for i=1:N
    I1=I1+c1(i)*y(i);
end
I2=0;
for i=1:2:N             %sólo sumo las componentes pares en el prod. escalar
    I2=I2+c2(i)*y(i); %podría sumar para i=1:N y sólo agregaría ceros,
end                     %dando el mismo resultado

beta=(h1/h2)^2;
I3=(beta*I2-I1)/(beta-1);

disp(['La aproximación de la integral calculada con paso h1=',num2str(h1),
       ' es ',num2str(I1),' y el error es de orden h1^2']);
disp(['La aproximación de la integral calculada con paso h2=',num2str(h2),
       ' es ',num2str(I2),' y el error es de orden h2^2']);
disp(['La nueva aproximación de la integral hallada por Richardson es ',num2str(I3),
       ' y el error es de orden h2^4']);
```

11) Modifique el archivo anterior para que aplique la Regla de Richardson a dos aproximaciones de una misma integral halladas por el método de Simpson con dos pasos h1 y h2.

12) Considere las integrales

$$I_1 = \int_0^{2\pi} \operatorname{sen}^2(x)\,dx \quad \text{e} \quad I_2 = \int_0^{2\pi} \operatorname{sen}(x)\cos(x)\,dx\,.$$

Los resultados exactos son $I_1 = \pi$ e $I_2 = 0$. Verifique numéricamente dichos resultados aplicando la regla de trapecios compuesta y realizando integración de Romberg. Elaborar o modificar el programa MATLAB para calcular I_1 e I_2 con distinta cantidad de intervalos, y así analizar la solución aproximada.

Ejercicios de cuadratura de Gauss Legendre

13) Utilizando las reglas de cuadratura de Gauss-Legendre, para dos y tres puntos calcular las integrales indicadas, y comparar los resultados obtenidos con las soluciones analíticas. Para realizar este ejercicio debe plantear un mapeo de modo de llevar las integrales al dominio unitario donde se conocen los puntos llamados Puntos de Integración de Gauss:

Puntos de Gauss	Abscisa	Coeficiente	Orden del Error de Truncamiento
2	$\pm\sqrt{3}/3$	1	4
3	±0.774596669 0.000000000	0.5555556 0.8888889	6
4	±0.861136312 ±0.339981044	0.3478548 0.6521452	8

Volumen Esfera $V = \int_{-R}^{R} \pi \cdot \left(\sqrt{\left(R^2 - x^2\right)}\right)^2 dx$

$$Int = \int_{0}^{3\pi/2} \sin(2x)\, dx$$

$$Int = \int_{2\pi}^{4\pi} \left[6 - 2\cos(x)\right] dx$$

$$Int = \int_{-\pi}^{\pi} \sin(2x) \cdot sen(x)\, dx$$

$$Int = \int_{\pi}^{3\pi} \cos(x) \cdot sen(x)\, dx$$

$$Int = \int_{3\pi}^{5\pi} \sin(x) \cdot sen(x)\, dx$$

Ejercicios con MATLAB

14) Modifique el programa “integraltrapecios_0.m” para que el vector **y** no sea ingresado manualmente sino que sea generado a través de la evaluación de una función en otro vector, el de las abscisas de puntos datos, **x**.

15) Escriba un programa.m de MATLAB que, dados un intervalo y un número de subintervalos, N, halle la integral de una función dada por el método de Romberg.

16) Escriba un programa.m de MATLAB que, dado un intervalo y un número de puntos de Gauss, N, halle la integral de una función dada por el método de Gauss-Legendre.

Ejercicios Teóricos

17) Los métodos vistos de integración numérica se basan en la siguiente idea:

$$I = I_n + E_n.$$

Además, se calcula la integral aproximada I_n según la fórmula

$$I_n = \sum_{j=0}^{n} \omega_j f(x_j)\,.$$

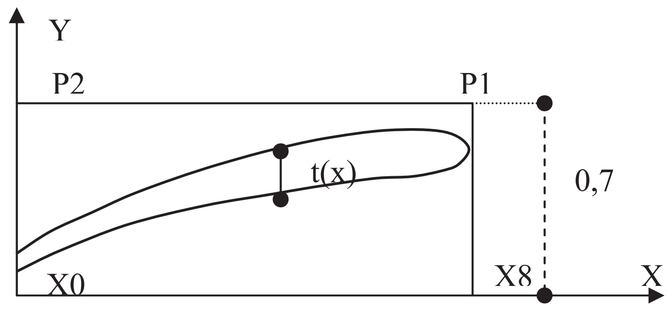
a) Para las reglas de cuadratura de Newton-Cotes, ¿de qué manera se calculan los coeficientes ω_j?
b) Para las reglas de cuadratura de Newton-Cotes, ¿qué puede decir de los valores x_j?
c) Para las reglas de cuadratura de Gauss-Legendre, ¿qué puede decir de los valores x_j?

18) Dada la integral $I = \int_{x_i}^{x_{i+1}} f(x)\,dx$ y conociendo en forma discreta la función y=f(x) sólo en los puntos (x_i, y_i) y (x_{i+1}, y_{i+1}), usando polinomios de Lagrange, plantee una interpolación que pueda integrarse para aproximar el valor de la integral.
   a) ¿Qué regla de integración ha obtenido?
   b) ¿Hasta qué grado de polinomio integra en forma exacta esta regla?
19) Demostrar el error de las Reglas de Trapecios Simple y de Simpson simple.
20) Habiendo obtenido dos aproximaciones de una misma integral, aplicando las reglas compuesta de trapecios y de Simpson, con el mismo paso, indique:
   a) ¿Cuál es el orden del error de la regla compuesta de trapecios?
   b) ¿Cuál es el orden del error de la regla compuesta de Simpson?
   c) Justifique cuál de las dos aproximaciones anteriores es más precisa.
21) a) ¿Cuál es el máximo grado de los polinomios que la regla de trapecios integra en forma exacta? (Este es el *orden de la regla* de integración de trapecios).
   b) En la regla de trapecios, ¿cuál es el *orden del error*? (en términos de h). Justifique mediante algún desarrollo.
   c) ¿Qué puede decir del orden del error en la regla de trapecios múltiple? Justifique su respuesta.
   d) ¿Cuál es el *orden de la regla* de integración de Simpson? (Esto es, ¿cuál es el máximo grado de los polinomios que esta regla integra en forma exacta?)
   e) En la regla de Simpson, ¿cuál es el *orden del error*?
   f) Si el error de la regla de Simpson simple es del orden de h^5, justifique que el orden del error de la regla de Simpson compuesta es del orden de h^4. Analice los valores obtenidos en alguno de los ejercicios anteriores en que haya usado las reglas de Simpson simple y compuesta.
22) ¿Hasta qué grado polinómico integra en forma exacta la cuadratura de Gauss-Legendre con tres puntos de Gauss? Justifique su respuesta, generalizando para n puntos.

Ejercicios de aplicaciones

23) Dada la siguiente función y=t(x): R→R, en forma discreta, se busca una aproximación del área A=A1-*A2*, que se deberá eliminar por mecanizado para lograr el alabe de la figura, a partir de una chapa plana. Siendo el A1 el área del polígono X0, X8, P1, P2 de la Figura; y A2 el resultado de integrar la función Y=t(x) discreta entre X0 y X8.
   a) Determinar el área $A2 = \int_{X0}^{X8} t(x)dx$, utilizando la Regla de Simpson compuesta con el menor paso h posible.
   b) Determinar el área *A2*, con la misma regla de integración, pero el doble del paso h, utilizados en el inciso anterior.
   c) Mejorar el valor obtenido utilizando extrapolación de Richardson.
   d) Calcular el área A

X	Y=t(x)
X0= 0	1/1500
X1= 1/8	80/1500
X2= 2/8	120/1500
X3= 3/8	150/1500
X4= 4/8	110/1500
X5=5/8	70/1500
X6=6/8	40/1500
X7=7/8	20/1500
X8=1	10/1500

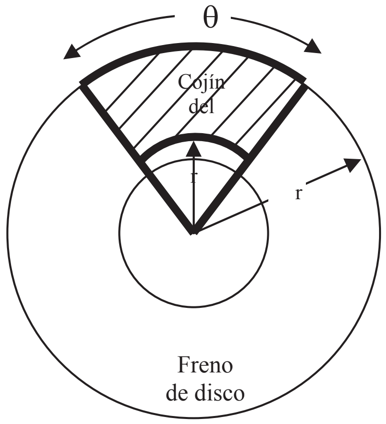


24) * Para simular las características térmicas de los frenos de disco (véase la figura), Secrist y Hornbeck tuvieron que aproximar numéricamente la "temperatura exterior promediada del área", T, en el cojín del freno, basándose para ello en la ecuación

$$T = \frac{\int_{r_e}^{r_0} T(r)\, r\, \theta_p \, dr}{\int_{r_e}^{r_0} r\, \theta_p \, dr}$$

, donde r_e representa el radio donde comienza el contacto entre cojín y disco, r_0 representa el radio exterior de dicho contacto, θ_p representa el ángulo subtendido por los cojines del freno del sector y T(r) es la temperatura en cada punto del cojín, la cual se obtuvo numéricamente al analizar la ecuación del calor. Si r_e=0,308 pies, r_0=0,478 pies, θ_p=0,7051 radianes y si las temperaturas dadas en la tabla siguiente se calcularon en varios puntos del disco, obtenga una aproximación de T.

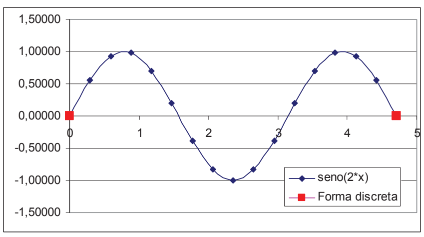
r (pies)	T(r) (°F)	r (pies)	T(r) (°F)	r (pies)	T(r) (°F)
0,308	640	0,376	1034	0,444	1204
0,325	794	0,393	1064	0,461	1222
0,342	885	0,410	1114	0,478	1239
0,359	943	0,427	1152		



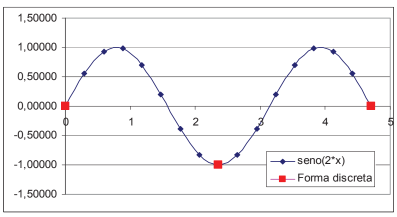
25) Se busca determinar la siguiente integral en forma numérica $Int = \int_{x0}^{xn} \sin(2x)\,dx = \left.\frac{-1}{2}\cos(2x)\right|_{x0}^{xn}$. Con x_0=0 y x_n=3π/2= 4,712, **<u>el valor exacto es 1</u>**. Si **n** es el número de intervalos; **h= (x_n- x_0)/n** es el paso de integración, la regla de Trapecios aproxima la integral mediante $Int = \frac{h}{2}\left(y_0 + \sum_{k=1}^{k=n-1} 2y_k + y_n \right)$, siendo y_k=sin(2*X_k).

Según la cantidad de intervalos en que se discretiza la función se tiene una aproximación distinta.

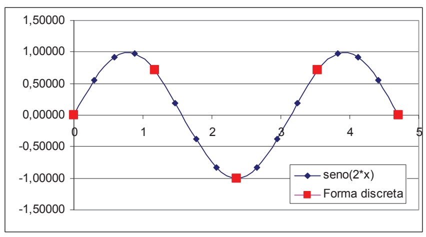
Para n=1, es decir un solo intervalo, se tiene



Para n=2, es decir dos intervalos, se tiene



Para n=4, es decir cuatro intervalos, se tiene



* Este ejercicio está tomado del libro "Análisis Numérico, 7°Ed.", Burden y Faires, International Thomson Editores, 2002, pág. 205-206.

En la siguiente Tabla se calculan las integrales para las distintas discretizaciones realizadas.

n	**16**	**n**	**8**	**n**	**4**	**n**	**2**	**n**	**1**
h	0,294524311	h	0,58905	h	1,1781	h	2,35619	h	4,7124
w	w* Sin(2x)	w	w* Sin(2x)	w	w* Sin(2x)	w	w* Sin(2x)	w	w* Sin(2x)
1	0	1	0	1	0	1	0	1	0
2	1,111140466								
2	1,847759065	2	1,84776						
2	1,961570561								
2	1,414213562	2	1,41421	2	1,4142				
2	0,390180644								
2	-0,765366865	2	-0,7654						
2	-1,662939225								
2	-2	2	-2	2	-2	2	-2		
2	-1,662939225								
2	-0,765366865	2	-0,7654						
2	0,390180644								
2	1,414213562	2	1,41421	2	1,4142				
2	1,961570561								
2	1,847759065	2	1,84776						
2	1,111140466								
1	2,1439E-15	1	2,1E-15	1	2E-15	1	2,1E-15	1	2E-15
Int	**0,970916536**		**0,88157**		**0,488**		**-2,3562**		**5E-15**
h	0,294524311		0,58905		1,1781		2,35619		4,7124

Y la extrapolación de Richardson sucesiva (Método de Romberg) resulta:

Orden	h^2	h^4	h^6	h^8	
Paso h	I(h^2)	I(h^4)	I(h^6)	I(h^8)	I(h^10)
0,29452	0,97092				
0,58905	0,88157	1,00069753			
1,1781	0,48798	1,01277013	0,999892687		
2,35619	-2,3562	1,43604331	0,98455192	1,000136191	
4,71239	5,1E-15	-3,14159265	1,741219036	0,972541331	1,00024441

26) Un volumen de revolución puede generarse al girar una curva y=f(x) dada, alrededor del eje x. Así es posible calcular el volumen de la forma

$$V = \int_{x0}^{xn} \pi \cdot [f(x)]^2 dx ,$$

siendo x0 y xn los límites inferior y superior del volumen.
La superficie lateral de dicho volumen se puede calcular mediante

$$S = \int_{x0}^{xn} 2 \cdot \pi \cdot f(x) \cdot \sqrt{[1 + \left(\frac{d(f(x))}{dx}\right)^2]}\, dx.$$

Elabore una planilla de cálculo que permita obtener una solución aproximada de dichas integrales para las funciones f(x) listadas más abajo. Para elaborar dicha planilla de cálculo se sugiere aplicar el método de Romberg.

Esfera para x entre -R y R	Elipsoide para x entre -a y a
$y = g(x) = \left(R^2 - x^2\right)^{1/2}$ $\frac{dg}{dx} = -x \cdot \left(R^2 - x^2\right)^{-1/2}$	$y = g(x) = \frac{b}{a}\left(a^2 - x^2\right)^{1/2}$ $\frac{dg}{dx} = -x \cdot \frac{b}{a}\left(a^2 - x^2\right)^{-1/2}$
Hiperboloide para x entre -a y a	Parábola Cuadrática para x entre x0 y xn
$y = g(x) = \frac{b}{a}\left(a^2 + x^2\right)^{1/2}$ $\frac{dg}{dx} = +x \cdot \frac{b}{a}\left(a^2 + x^2\right)^{-1/2}$	$y = g(x) = \frac{1}{a^2}\left(a^2 + x^2\right)$ $\frac{dg}{dx} = +x \cdot \frac{2}{a^2}$

Para el caso del <u>volumen de una esfera</u> resulta:

R	**1**									
x0	**-1**									
xn	**1**		**n**	**16**	**n**	**8**	**n**	**4**	**n**	**2**
			h	**0.125**	**h**	**0.25**	**h**	**0.5**	**h**	**1**
x	f(x)= (R^2-x^2)^0.5	PI()*f(x)^2	**w**	**Vh**	**w**	**Vh**	**w**	**Vh**	**w**	**Vh**
-1	0.00000	0.00000	1	0	1	0	1	0	1	0
-0.875	0.48412	0.73631	2	1.472621556						
-0.75	0.66144	1.37445	2	2.748893572	2	2.74889				
-0.625	0.78062	1.91441	2	3.828816047						
-0.5	0.86603	2.35619	2	4.71238898	2	4.71239	2	4.7124		
-0.375	0.92702	2.69981	2	5.399612373						
-0.25	0.96825	2.94524	2	5.890486225	2	5.89049				
-0.125	0.99216	3.09251	2	6.185010537						
0	1.00000	3.14159	2	6.283185307	2	6.28319	2	6.2832	2	6.28319
0.125	0.99216	3.09251	2	6.185010537						
0.25	0.96825	2.94524	2	5.890486225	2	5.89049				
0.375	0.92702	2.69981	2	5.399612373						
0.5	0.86603	2.35619	2	4.71238898	2	4.71239	2	4.7124		
0.625	0.78062	1.91441	2	3.828816047						
0.75	0.66144	1.37445	2	2.748893572	2	2.74889				
0.875	0.48412	0.73631	2	1.472621556						
1	0.00000	0.00000	1	0	1	0	1	0	1	0
			V	**4.172427743**		**4.12334**		**3.927**		**3.14159**
			h	**0.125**		**0.25**		**0.5**		**1**

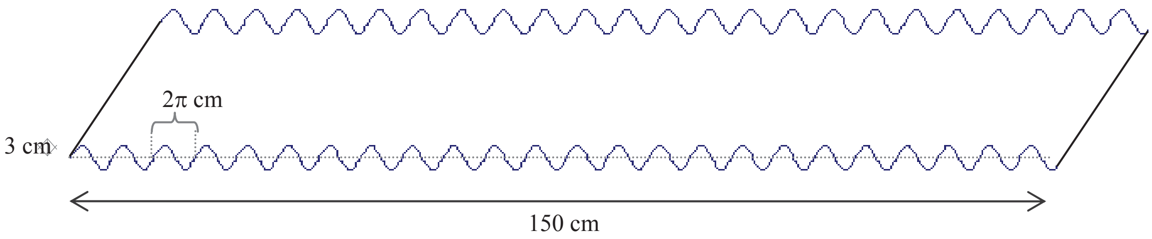
Y la extrapolación de Richardson sucesiva (Método de Romberg)

Orden	h^2	h^4	h^6	h^8
Paso h	v(h^2)	V(h^4)	V(h^6)	V(h^8)
0.125	4.1724			
0.25	4.1233	4.18879		
0.5	3.927	4.18879	4.18879	
1	3.1416	4.18879	4.18879	4.18879

27) Se tiene que construir una hoja corrugada usando una máquina que comprime una hoja plana de aluminio, logrando que su sección transversal tenga la forma de una onda de la función seno. Se necesita una hoja corrugada de un metro y medio de longitud, en la que cada onda tenga una amplitud de 3 cm (desde la línea central) y un periodo de 2π cm.

a) Escriba una fórmula que represente a la función senoidal correspondiente a la sección de la hoja ya comprimida.

b) Recuerde que la longitud de arco del gráfico de la función f entre x=a y x=b se halla calculando la integral $\int_a^b \sqrt{1+[f'(x)]^2}\,dx$. Halle la longitud que debe tener la hoja plana inicialmente, planteando y resolviendo la integral correspondiente. (Se recomienda tomar varios puntos, implementando la regla en computadora).



Ejercicios Integradores

28) Dado el siguiente programa de MATLAB para calcular una aproximación de la integral de una función dada en forma discreta mediante el método de Simpson compuesto, completar con los comandos necesarios y responder.

```
% integral por Simpson
N=input('ingrese el nº de puntos datos, impar y mayor o igual que 3: ');
h=input('ingrese el paso h entre abscisas: ');
y=zeros(N,1);
x=zeros(N,1);
x(1)=input('ingrese el valor de la abscisa del primer punto: ');
for i=1:N
    y(i)=input(['ingrese la ordenada y(',num2str(i),'): ']);
end
for i=2:N
    x(i)=________
end
c=zeros(N,1);
c(1)=1/3;
c(N)=1/3;
for i=2:2:N-1
    c(i)=________
end
____ i=3:2:N-2
    c(i)=________
____

S=0;
____ i=1:N
    S=S+__________
____
int=S*h;
disp('El vector de coeficientes es '),disp(c)
disp('El valor de la integral es '), disp(int)
plot(x,y)
```

- ¿Qué realiza la variable S?
- ¿Qué almacena el vector c?
- ¿Qué realiza el comando plot?. ¿Qué tiene asignado las variables x e y?

29) Calcular mediante un programa MATLAB la siguiente integral haciendo uso de derivación e integración numérica.

$$K_{ij} = \int_0^L \left\{ \frac{d^2\phi_i(x)}{dx^2} \, k \, \frac{d^2\phi_j(x)}{dx^2} \right\} dx \quad i, j = 1,2,3$$

siendo:

$$\phi_k(x) = Cosh(\beta_k L \cdot \frac{x}{L}) - Cos(\beta_k L \cdot \frac{x}{L}) - \sigma_k (Senh(\beta_k L \cdot \frac{x}{L}) - Sen(\beta_k L \cdot \frac{x}{L}))$$

$$con \qquad \sigma_k = \frac{Senh(\beta_k L) - Sen(\beta_k L)}{Cosh(\beta_k L) + Cos(\beta_k L)} \, ; \;\; \beta_1 L = 1{,}875 \, ; \quad \beta_2 L = 4{,}694 \;\; ; \quad \beta_3 L = 7{,}8547$$

y verificar que para L=100 y k=500 las integrales dan los coeficientes de la siguiente matriz:

$$K = \left[\frac{k}{L^3} \begin{pmatrix} (1{,}8751)^4 & 0 & 0 \\ 0 & (4{,}694)^4 & 0 \\ 0 & 0 & (7{,}8547)^4 \end{pmatrix} \right]$$

Guía de Estudio para Derivación Numérica.

Contenido

Guía de Estudio para Derivación Numérica. 1
Bibliografía recomendada: 1
Objetivo: 1
Definiciones 1
Ejercicios de reglas de derivación. 2
Ejercicios con MATLAB 3
Ejercicios Teóricos 5
Ejercicio Integrador 6

Bibliografía recomendada:

Apuntes de la Càtedra.
J. Mathews, J. Fink (2000). *Métodos Numéricos con Matlab.* Prentice Hall.
S. Nakamura (1997). *Análisis Numérico y Visualización Gráfica con MATLAB.* Prentice Hall
S. Nakamura (1992). *Métodos Numéricos Aplicados con Software.* Prentice Hall
S. Chapra, R. Canale (1999). *Métodos Numéricos para Ingenieros*, Mc Graw Hill.
R. Burden, J. Faires (1998). *Análisis Numérico*, International Thomson Editores.

Objetivo:

Se pretende comprender a las aproximaciones para hallar derivadas en forma numérica y aplicar el procedimiento de extrapolación de Richardson.

Se busca poder implementarlos en planillas de cálculo y en programas de MATLAB.

Definiciones

Se entiende por Regla de Derivación Numérica a la combinación lineal de las ordenadas de la función discreta, que aproxima una derivada de dicha función en uno de los puntos de la misma.

Una Regla de Derivación Numérica es la obtenida a partir a partir del Polinomio de Interpolación de una función discreta con un número determinado de puntos. Alternativamente, mediante el empleo de Serie de Taylor.

Cada Regla de Derivación Numérica tiene un error de truncamiento local asociado a su definición, que permite el control de la confianza que se tiene en la aproximación obtenida.

El error de truncamiento local siempre depende de la derivada de algún orden de la función a la que se aplica la regla de derivación, y de una potencia de la separación de los datos.

El orden del error de la regla de derivación está dado por la potencia de la separación de los datos en su error de truncamiento local.

El orden de la regla de derivación es el mayor grado polinómico para el cual no se comete error; y corresponde a un grado menos que la máxima derivada que aparece en su error de truncamiento local.

Ejercicios de reglas de derivación.

1) Dada la función f : **R**→ **R** en forma discreta, se busca determinar algunas derivadas en puntos x_i. Sea la función:

x	0.8	0.9	1	1.1	1.2
f(x)	2.5537129	2.78927897	3	3.19062036	3.36464311

   a) Calcule una aproximación de la derivada primera de la f(x) en x=0.8. Elija una fórmula de derivación según los datos que tiene. Use el paso h_1 que sea el menor posible.

   b) Calcule una aproximación de la derivada primera de la f(x) en x=0.8, con la misma fórmula de derivación utilizada en el inciso anterior, pero usando un paso $h_2 = 2h_1$. Se identifica a h_1 como el paso del inciso anterior.

   c) Indique cuál es el orden del error de truncamiento local de la fórmula de derivación utilizada en los incisos anteriores.

   d) Realice extrapolación de Richardson* a partir de los resultados de los incisos anteriores. Debe justificar el β que utiliza en la fórmula de Richardson.

   e) Repita los incisos a), b), c) y d) pero con otra fórmula de derivación para derivada primera.

   f) Repita los incisos a), b), c), d) y e) pero para el punto x=1.2.

   g) Calcular la función discreta f ’(x) derivada primera de la función dada asegurando un error del orden $(h_1)^2$. Exprese mediante un operador matricial.

   h) Calcule la función discreta f ”(x) derivada segunda de la función dada asegurando un error del orden $(h_1)^2$. Exprese mediante un operador matricial.

   i) Calcule las derivadas tercera y cuarta de la función en el punto x=1.

2) Dada la función f : **R**→ **R** en forma discreta, se busca determinar algunas derivadas en puntos x_i. Sea la función:

x	0.1	0.35	0.6	0.85	1.1
f(x)	-1.60517019	0.90035575	1.97834875	2.67496214	3.19062036

   a) Calcule una aproximación de la derivada primera de la f(x) en x=0.1. Elija una fórmula de derivación según los datos que tiene. Use un paso h_1 tal que pueda tener datos para un paso $h_2 = 2h_1$.

   b) Calcule una aproximación de la derivada primera de la f(x) en x=0.1, con la misma fórmula de derivación utilizada en el inciso anterior. Use un paso $h_2 = 2h_1$, siendo h_1 el paso del inciso anterior.

   c) Realice extrapolación de Richardson a partir de los resultados de los incisos anteriores. Debe justificar el β que utiliza en la fórmula de Richardson.

---

* Extrapolación de Richardson consiste en tener dos aproximaciones de la misma derivada para pasos h_1 y h_2; con un error de aproximación del orden de h^n, y lograr una mejor aproximación con error del orden h^{n+2} .

$$D(h_1); D(h_2)\ tiene\ orden\ de\ O(h^n)$$

$$\beta = \left(\frac{h_1}{h_2}\right)^n$$

$$D(h^{n+2}) = \frac{\beta \cdot D(h_2) - D(h_1)}{\beta - 1}$$

d) Repita los incisos a), b), c) y d) pero para el punto x=1.1.

e) Calcular la función discreta f '(x) derivada primera de la función dada asegurando un error del orden $(h_1)^2$. Exprese mediante un operador matricial.

f) Calcule la función discreta f "(x) derivada segunda de la función dada asegurando un error del orden $(h_1)^2$. Exprese mediante un operador matricial.

g) Calcule las derivadas terceras y cuarta de la función en el punto x=0.6 utilizando fórmulas centrales.

3) Considere las siguientes funciones dadas analíticamente

$$f_1(x)= \text{sen}(x) \qquad y \qquad f_2(x)= x^6-x^5+x^4-x^3+x^2-x+6.$$

a) Elija una discretización del dominio $[x_0;x_n]$, es decir una serie de (n+1) puntos entre x_0 y x_n. Es conveniente que exista un punto en el centro del intervalo, esto es que el número de intervalos sea par.

b) Calcular la función discreta f '(x) derivada primera de la función dada asegurando un error del orden $(h_1)^2$. Exprese mediante un operador matricial. Compare sus resultados con los valores exactos.

c) Calcule la función discreta f "(x) derivada segunda de la función dada asegurando un error del orden $(h_1)^2$. Exprese mediante un operador matricial. Compare sus resultados con los valores exactos.

d) Elija otra discretización de modo que pueda evaluar las derivadas con otro paso y aplicar extrapolación de Richardson. Compare sus resultados con los valores exactos.

Ejercicios con MATLAB

4) Escriba un programa de MATLAB que halle una aproximación de la derivada primera de una función en un punto x, conocidos los valores de la función en dicho punto x y en otro, x+h. Plantee la derivada primera en x como un producto escalar entre el vector de ordenadas y, y un vector de coeficientes. Considerar una función discreta dada por los puntos (x1;y1) (x2;y2).

Una posibilidad es el siguiente código para MATLAB

```
% este programa calcula  las derivadas primeras hacia delante para dos puntos
clc, clear
%leer datos
y=zeros(2,1);          % genera y dimensiona el vector columna para las ordenadas
x=zeros(2,1);          % genera y dimensiona el vector columna para las abscisas
x(1)=input('ingrese el valor de la abscisa del primer punto: ');
h=input('ingrese el paso h: ');
x(2)= x(1)+h;
y(1)=input(['ingrese el valor de la ordenada del punto ',num2str(1),': ']);
y(2)=input(['ingrese el valor de la ordenada del punto ',num2str(2),': ']);
%
% generación de coeficientes para derivada primera adelante
cad= [-1    1]/(h);
% calculamos la derivada primera adelante dy
dy=cad*y;

% Entrega de resultados
% Los datos son
matriz=[x,y];
disp(['       x   ','       y  ']);
disp(matriz);
disp(' El valor de la derivada primera adelante es dy='),dy
```

```
%graficamos la función y la derivada
% armamos la función discreta derivada primera para graficar
dyg=[dy;dy];
hold all            % para que superponga curvas en un mismo gráfico
plot(x,y,'--b')     % grafica de la función discreta
plot(x,dyg,'--r')   % grafica de la derivada primera
title('Derivadas Primeras')
legend('función f(x) discreta',  'función discreta derivada primera')
grid on
```

5) Modifique el programa anterior eliminando todas las líneas y comentarios superfluos. Dejar sólo lo necesario para calcular y entregar el valor de la derivada primera. Eliminar las gráficas.

6) Modifique el programa anterior para que halle una aproximación de la derivada primera de una función en un punto x, conocidos los valores de la función en dicho punto x y en otro, x–h.

7) Unifique los programas anteriores incorporando el ingreso de una opción que haga derivada primera adelante o atrás de la función discreta definida por dos puntos.

8) Modifique el programa anterior para que halle una aproximación de la derivada primera de una función en un punto x, conocidos los valores de la función en los puntos x, x+h y x–h.

9) Modifique el programa anterior para que halle una aproximación de la derivada primera de una función en un punto x, conocidos los valores de la función en dicho punto x, en x+h y en x+2h.

10) Modifique el programa anterior para que halle una aproximación de la derivada primera de una función en un punto x, conocidos los valores de la función en dicho punto x, en x–h y en x–2h.

11) Unifique los programas anteriores incorporando el ingreso de una opción que haga derivada primera adelante o atrás o central de la función discreta definida por tres puntos, y aclare en que punto se calcula la derivada.

12) Escriba un programa de MATLAB, basándose en los programas anteriores, que dada una función en forma discreta en un dominio [x0, xn], halle las derivadas primeras aproximadas de la función en todos los puntos del dominio.

    Una forma de hacerlo es la siguiente:

```
% este programa calcula el vector con las derivadas primeras aproximadas en cada x,
% usando derivada primera asimétrica hacia adelante en el primer punto, asimétrica
% hacia atrás en el último y central en los interiores.
clc
clear

N=input('ingrese el n° de datos con los que va a trabajar, mayor o igual que 3: ');

%leer datos
h=input('ingrese el paso h: ');
y=zeros(N,1);
x=zeros(N,1);
x(1)=input('ingrese el valor de la abscisa del primer punto: ');

%genero el vector de abscisas
for i=2:N
    x(i)=x(i-1)+h;
end

%genero o ingreso el vector de ordenadas
for i=1:N
    y(i)=input(['ingrese el valor de la ordenada del punto ',num2str(i),': ']);
end
```

```
% genero la matriz de coeficientes
c=zeros(N,N);
% armamos la derivada primera asimétrica hacia adelante para el primer punto
c(1,1)=-3/(2*h);
c(1,2)=4/(2*h);
c(1,3)=-1/(2*h);
% armamos la derivada primera asimétrica hacia atrás para el último punto
c(N,N)=3/(2*h);
c(N,N-1)=-4/(2*h);
c(N,N-2)=1/(2*h);
% armamos la derivada primera central hacia atrás para los puntos interiores
for i=2:N-1                 % i varía desde 2 hasta N-1
    c(i,i-1)=-1/(2*h);
    c(i,i+1)=1/(2*h);
end
disp('la matriz de coeficientes es ');disp(c);

% formamos el vector dy con los valores de la derivada
dy=c*y;
% armo una matriz para mostrar por columnas
% los valores de x con sus correspondientes valores de función y de derivada primera.
matriz=[x,y,dy];
disp(['       x  ','       y  ','      dy  ']);
disp(matriz);

%graficamos la función y la derivada
hold all      % para que superponga varias curvas en un mismo gráfico
plot(x,y)     % grafica en abscisas los valores del vector x y en las ordenadas los
              % del vector y
plot(x,dy)    %grafica en abscisas los valores del vector x y en las ordenadas los
              % del vector dy
```

- Incorpore leyendas en el gráfico para cada una de las curvas.

13) Modifique el programa anterior para que entregue un vector conteniendo todas las derivadas segundas de la función.

Ejercicios Teóricos

14) Demuestre cuál es orden del error de truncamiento local de la fórmula de derivada primera hacia adelante y hacia atrás.

15) Demuestre la obtención de la fórmula de derivación central para derivada segunda y su error de truncamiento local.

16) Para los datos de la siguiente tabla, ¿puede realizar extrapolación de Richardson cuando calcula la derivada segunda en el punto x=0.9? ¿Y en el punto x=1? Justifique, sin calcular las derivadas (nótese que son los datos del ejercicio 1).

x	0.8	0.9	1	1.1	1.2
f(x)	2.5537129	2.78927897	3	3.19062036	3.36464311

17) Demuestre cuál es el error de truncamiento local de las fórmulas centrales de derivadas tercera y cuarta.

18) Obtener la derivada primera central y su error de truncamiento.

19) Dados los puntos (x_0, y_0); (x_1, y_1); (x_2, y_2),:

- Obtener el polinomio interpolante de Newton.
- Derivar una vez el polinomio interpolante de Newton y evaluar esa derivada en x_1 y en x_2. Comparar ambos resultados con fórmulas de derivadas primeras numéricas obtenidas mediante combinaciones de Series de Taylor

- Derivar dos veces el polinomio interpolante de Newton y evaluar esa derivada en x_1 y comparar con fórmulas de derivadas segundas numéricas obtenidas mediante combinaciones de Series de Taylor.

20) Obtener el error de truncamiento de las siguientes derivadas numéricas:

$$a) \qquad f_s^{iv} = \frac{1}{h^4}\left[f_{s-2} - 4f_{s-1} + 6f_s - 4f_{s+1} + f_{s+2}\right]$$

$$b) \qquad f_s''' = \frac{1}{2h^3}\left[-f_{s-2} + 2f_{s-1} + 0f_s - 2f_{s+1} + f_{s+2}\right]$$

$$c) \qquad f_s'' = \frac{1}{h^2}\left[-\frac{1}{12}f_{s-2} + \frac{4}{3}f_{s-1} - \frac{5}{2}f_s + \frac{4}{3}f_{s+1} - \frac{1}{12}f_{s+2}\right]$$

$$d) \qquad f_s' = \frac{1}{h}\left[\frac{1}{12}f_{s-2} - \frac{2}{3}f_{s-1} + 0f_s + \frac{2}{3}f_{s+1} - \frac{1}{12}f_{s+2}\right]$$

21) Demostrar que la derivada primera hacia adelante al considerar tres puntos, resulta:

$$f_s' = \{[-3/(2h)]\cdot f_s + [4/(2h)]\cdot f_{s+1} + [-1/(2h)]f_{s+2}\}$$

y que su error de truncamiento local es: $Er = +\dfrac{h^2}{3} f_s'''$.

22) Demostrar que la derivada primera hacia atrás al considerar tres puntos, resulta:

$$f_s' = \{[3/(2h)]\cdot f_s + [-4/(2h)]\cdot f_{s-1} + [1/(2h)]f_{s-2}\}$$

y que su error de truncamiento local es: $Er = -\dfrac{h^2}{3} f_s'''$.

23) Demostrar que la derivada segunda asimétrica hacia adelante por cuatro puntos resulta

$$f_s'' = \left(\frac{1}{h^2}\right)\left(2f_s - 5f_{s+1} + 4f_{s+2} - 1f_{s+3}\right),$$

y su error de truncamiento local, son los siguientes: $Er = -\dfrac{22h^2}{24} f_\xi^{iv}$.

24) Demostrar que la derivada segunda asimétrica hacia atrás por cuatro puntos resulta igual a la derivada segunda hacia adelante por cuatro puntos con igual error de truncamiento local.

25) Demostrar la Extrapolación de Richardson.

Ejercicio Integrador

26) Dadas las funciones sen(x) y cos(x), realizar un programa en MATLAB que:

- Obtenga una versión discreta de dichas funciones considerando 101 puntos en el intervalo [0, PI]
- Grafique las funciones discretas y las derivadas primeras y segundas de dichas funciones discretas
- Grafique la función discreta que se obtiene de multiplicar las derivadas segundas de dichas funciones discretas.

27) Interpretar el siguiente programa en MATLAB que usa vectores x, y, d1 y d2; y matrices C1 y C2.

```
%
clc, clear
N=input('ingrese el n° de datos con los que va a trabajar, mayor o igual que 4: ');
y=zeros(N,1);
x=zeros(N,1);
x(1)=input('ingrese el valor de la abscisa del primer punto: ');
x(N)=input('ingrese el valor de la abscisa del último punto: ');
h=(x(N)-x(1))/(N-1);
for i=2:N
    x(i)=x(i-1)+h;
end
y=sin(x);
%
C1=zeros(N,N);
C1(1,1)=-3;
C1(1,2)=4;
C1(1,3)=-1;
C1(N,N)=3;
C1(N,N-1)=-4;
C1(N,N-2)=1;
for i=2:N-1
    C1(i,i-1)=-1;
    C1(i,i+1)=1;
end
C1=C1/(2*h);
d1=C1*y;
%
C2=zeros(N,N);
C2(1,1)=2;
C2(1,2)=-5;
C2(1,3)=4;
C2(1,4)=-1;
C2(N,N)=2;
C2(N,N-1)=-5;
C2(N,N-2)=4;
C2(N,N-3)=-1;
for i=2:N-1
    C2(i,i-1)=1;
    C2 (i,i)=-2;
    C2 (i,i+1)=1;
end
C2=C2/(h^2);
d2=C2*y;

hold all
plot(x,y, 'b')
plot(x,d1,'r')
plot(x,d2,'g')
```

- Interpretar el contenido de los vectores x e y.
- Interpretar las matrices C1 y C2
- Expresar que se obtiene con cada comando Plot
- Modificar el programa anterior para evitar el armado de la matriz C1 y el producto C1*y pero igualmente obtener el vector d1. Usar estructuras algorítmicas for-end y las if-else-end.
- Modificar el programa anterior para evitar el armado de la matriz C2 y el producto C2*y pero igualmente obtener el vector d2. Usar estructuras algorítmicas for-end y las if-else-end.

Guía de Estudio para Ecuaciones Diferenciales con Valores de Contorno.

Contenido

Guía de Estudio para Ecuaciones Diferenciales con Valores de Contorno. 1
Bibliografía recomendada: 1
Objetivo: 1
Definiciones 1
Discretización de Ecuaciones Diferenciales Ordinarias 2
Ejercicios de EDO 3
Ejercicio Integrador de EDO 7
Discretización de Ecuaciones Diferenciales a Derivadas Parciales 8
Ejercicios de Ecuaciones Diferenciales a Derivadas Parciales 9

Bibliografía recomendada:

J. Mathews, J. Fink (2000). *Métodos Numéricos con Matlab.* Prentice Hall.
S. Nakamura (1997). *Análisis Numérico y Visualización Gráfica con MATLAB.* Prentice Hall
S. Nakamura (1992). *Métodos Numéricos Aplicados con Software.* Prentice Hall
S. Chapra, R. Canale (1999). *Métodos Numéricos para Ingenieros*, Mc Graw Hill.
R. Burden, J. Faires (1998). *Análisis Numérico*, International Thomson Editores.

Objetivo:

Se pretende comprender utilizar las reglas de derivadas numéricas para obtener soluciones aproximadas de ecuaciones diferenciales con valores de contorno.

Se busca obtener soluciones discretas aproximadas de ecuaciones diferenciales con valores de contorno.

Definiciones

Las ecuaciones diferenciales relacionan las funciones incógnitas con sus derivadas.

Cuando las funciones incógnitas dependen de una única variable independiente, las ecuaciones diferenciales son denominadas ordinarias.

Cuando las funciones incógnitas dependen de más de una variable independiente, las ecuaciones diferenciales son a derivadas parciales.

Cuando la máxima derivada que aparece en la ecuación diferencial es mayor o igual a dos, las ecuaciones diferenciales pueden ser de valores iniciales o de valores de contorno.

Las ecuaciones diferenciales son de valores de contorno, cuando la solución se busca en un dominio cerrado de las variables independientes.

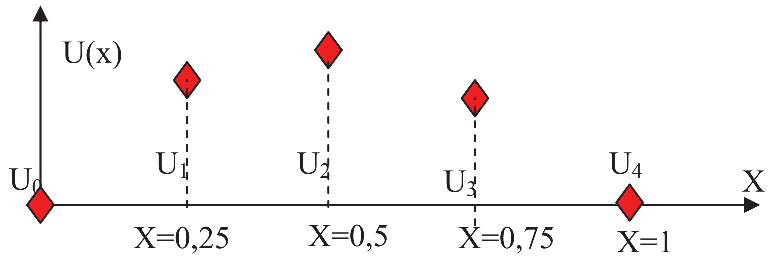
Para que la solución sea unívocamente determinada deben asociarse condiciones a cumplir por la función solución en puntos especiales. Cuando éstas condiciones están expresadas en las fronteras del dominio cerrado, la ecuación diferencial es de valores de contorno.

Discretización de Ecuaciones Diferenciales Ordinarias

Se busca *u(x)* solución de

$$-\frac{d^2u(x)}{dx^2}+u(x)-x=0 \quad en \;\; \Omega=\{x\,\varepsilon\, R: 0\leq x\leq 1\}$$
$$u(0)=0$$
$$u(1)=0$$

En vez de encontrar la solución exacta *u(x)* en cada uno y todos los puntos del dominio Ω, se plantea encontrar una ***solución aproximada en forma de función discreta***, sólo en algunos puntos elegidos del dominio y equidistantes entre sí, identificados con su abscisa X_k. Es decir, se busca $U(X_k)=U_k$ con k=0,N; función discreta que es una aproximación de la función continua *u(x)*.



Se divide el dominio Ω en N segmentos iguales y así quedan definidos N+1 puntos que incluyen a los bordes del dominio. En cada uno de los X_k se puede plantear la ecuación diferencial a resolver pero con una aproximación de la derivada segunda en forma de derivada numérica considerando la función discreta U_k. Así se puede escribir:

$$-\frac{1}{\Delta x^2}\left[U_{k-1}-2\cdot U_k+U_{k+1}\right]+U_k-X_k=0 \quad en\;\; X_k \qquad con \;\; k=1,(N-1)$$

siendo Δx=1/N la distancia entre los puntos. Es una ecuación algebraica cuyas incógnitas son las U_k.

De estas ecuaciones se pueden plantear tantas como puntos interiores; es decir N-1 ecuaciones y se tienen N+1 incógnitas. Además se tienen las dos ecuaciones correspondientes a las Condiciones de Contorno, que agregan dos ecuaciones más. Así se tienen N+1 ecuaciones con N+1 incógnitas.

Si se consideran 5 puntos en todo el dominio, que son 3 en el interior y uno en cada uno de los bordes, es posible plantear

en X_0 se debe cumplir que: $U_0=0$

en X_1 se debe cumplir que: $-\frac{1}{0{,}25^2}\left[U_0-2\cdot U_1+U_2\right]+U_1-X_1=0$

en X_2 se debe cumplir que: $-\frac{1}{0{,}25^2}\left[U_1-2\cdot U_2+U_3\right]+U_2-X_2=0$

en X_3 se debe cumplir que: $-\frac{1}{0{,}25^2}\left[U_2-2\cdot U_3+U_4\right]+U_3-X_3=0$

en X_4 se debe cumplir que: $U_4=0$

O bien en forma de sistema de ecuaciones lineales

$$\left[\frac{1}{0{,}25^2}\begin{pmatrix}2 & -1 & 0\\ -1 & 2 & -1\\ 0 & -1 & 2\end{pmatrix}+\begin{pmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{pmatrix}\right]\cdot\begin{Bmatrix}U_1\\ U_2\\ U_3\end{Bmatrix}=\begin{bmatrix}32+1 & -16 & 0\\ -16 & 32+1 & -16\\ 0 & -16 & 32+1\end{bmatrix}\cdot\begin{Bmatrix}U_1\\ U_2\\ U_3\end{Bmatrix}=\begin{Bmatrix}0{,}25\\ 0{,}5\\ 0{,}75\end{Bmatrix}$$

La solución aproximada resulta

$$\begin{Bmatrix}U_1\\ U_2\\ U_3\end{Bmatrix}=\begin{Bmatrix}0{,}03488525\\ 0{,}05632582\\ 0{,}05003676\end{Bmatrix}$$

La solución aproximada completa, que incluye las condiciones de borde es:

$$\begin{Bmatrix}U_0\\ U_1\\ U_2\\ U_3\\ U_4\end{Bmatrix}=\begin{Bmatrix}0\\ 0{,}03488525\\ 0{,}05632582\\ 0{,}05003676\\ 0\end{Bmatrix}$$

Justificar que con la solución obtenida una aproximación de la función derivada primera de esta solución se puede calcular mediante:

$$\begin{Bmatrix}U_0^{(1)}\\ U_1^{(1)}\\ U_2^{(1)}\\ U_3^{(1)}\\ U_4^{(1)}\end{Bmatrix}=\frac{1}{(2*0{,}25)}\begin{bmatrix}-3 & 4 & -1 & 0 & 0\\ -1 & 0 & 1 & 0 & 0\\ 0 & -1 & 0 & 1 & 0\\ 0 & 0 & -1 & 0 & 1\\ 0 & 0 & 1 & -4 & 3\end{bmatrix}\begin{Bmatrix}0\\ 0{,}03488525\\ 0{,}05632582\\ 0{,}05003676\\ 0\end{Bmatrix}$$

Ejercicios de EDO

1) Dada la siguiente ecuación diferencial ordinaria,

$$10\,\frac{d^2u(x)}{dx^2}+3(x-1)=0 \qquad si\ \ x\,\varepsilon\,\Omega=\{x\,\varepsilon\,R: 0\le x\le 1\}$$
$$u(0)=0$$
$$u(1)=5$$

Usando una aproximación central de la derivada segunda, hallar u(x) en forma discreta para la siguiente discretización propuesta con tres puntos interiores al dominio

X	0	0.25	0.5	0.75	1
u(x)	u_0	u_1	u_2	u_3	u_4

Para ello realice las siguientes actividades:

a) Identificar el paso h (o Δx) de la discretización propuesta.
b) En cada punto interior x_i escribir la aproximación de la ecuación diferencial utilizando la aproximación de la derivada segunda central.
c) En cada punto del contorno escribir la ecuación que resulta de imponer la condición de borde o valor de contorno.
d) Reconozca que se consiguió un sistema de ecuaciones lineales. ¿Cuántas ecuaciones y cuántas incógnitas tiene dicho sistema de ecuaciones lineales?.

e) Es posible aumentar el paso h utilizado y utilizando la misma aproximación para la derivada numérica encontrar una solución aproximada. ¿Cuál es el máximo valor de h que puede utilizarse?.

f) Es posible disminuir a la mitad el paso h utilizado, y utilizando la misma aproximación para la derivada numérica encontrar una solución aproximada. ¿Cuál es el orden del sistema de ecuaciones lineales al que se llega?. Exprese la matriz de coeficientes y el término independiente.

2) Dada la siguiente ecuación diferencial ordinaria con valores en la frontera o condiciones de borde,

$$EJ\,\frac{d^2u(x)}{dx^2} + P(x-L) = 0 \qquad si\ x\varepsilon\Omega = \{x\varepsilon R : 0 \le x \le L\}$$

$$u(0) = 0$$

$$\frac{du}{dx}(L) = 0$$

Usando una aproximación central de la derivada segunda, hallar u(x) en forma discreta para la siguiente discretización propuesta con tres puntos interiores al dominio con L=10; P=100; EJ=10000

X	0	0.25 L	0.50 L	0.75 L	L
u(x)	u_0	u_1	u_2	u_3	u_4

Para ello realice las siguientes actividades:

a) Identificar el paso h (o Δx) de la discretización propuesta.

b) En cada punto interior x_i escribir la aproximación de la ecuación diferencial utilizando la aproximación de la derivada segunda central.

c) En cada punto del contorno escribir la ecuación que resulta de imponer la condición de borde o valor de contorno usando derivada primera de igual orden que la derivada segunda central. ¿Qué posibilidades tiene para aproximar la derivada primera de la condición de borde?. Para la derivada primera utilizada, ¿cuántos puntos cómo mínimo debe haber en el interior del dominio?.

d) Reconozca que se consiguió un sistema de ecuaciones lineales. ¿Cuántas ecuaciones y cuántas incógnitas tiene dicho sistema de ecuaciones lineales?.

e) Es posible aumentar el paso h utilizado y utilizando la misma aproximación para la derivada numérica encontrar una solución aproximada. ¿Cuál es el máximo valor de h que puede utilizarse?.

f) Es posible disminuir a la mitad el paso h utilizado, y utilizando la misma aproximación para la derivada numérica encontrar una solución aproximada. ¿Cuál es el orden del sistema de ecuaciones lineales al que se llega?. Exprese la matriz de coeficientes y el término independiente.

g) Transformar el dominio x a dominio unitario entre cero y uno usando un cambio de variables, repetir la obtención del sistema de ecuaciones.

3) Dada la siguiente ecuación diferencial ordinaria con valores en la frontera o condiciones de borde,

$$-EA\,\frac{d^2u(x)}{dx^2} = \rho Ag \qquad si\ x\varepsilon\Omega = \{x\varepsilon R : 0 \le x \le L\}$$

$$u(0) = 0$$

$$EA\,\frac{du}{dx}(L) = Mg$$

con $\Omega = \{x\varepsilon\ R : 0 \le x \le L\}$, *EA=210000, L=5000, Mg=200,* y ρAg =100 encontrar una solución aproximada considerando tres puntos en el interior del dominio.

4) La ecuación diferencial en coordenadas polares que determina la posición de una membrana circular (como el parche de un tambor) de radio interno ra y radio externo rb, sometida a una tracción uniforme T y capaz de sostener una acción distribuida p(r), está dada por

$$T\frac{d^2u(r)}{dr^2}+\frac{T}{r}\frac{du}{dr}+p(r)=0, \qquad \text{si } r\in\Omega=\{r\in R: ra\le r\le rb\},$$
$$\text{con } u(ra)=u(rb)=0.$$

Plantee una solución aproximada usando derivación numérica y al menos cinco puntos interiores en el dominio Ω. Encuentre el sistema de ecuaciones lineales que debe resolverse. Como datos se puede considerar: ra=1; rb=10; T= 500; p(r)= 100.

5) Un elemento unidimensional que se encuentra en un dominio $\Omega=\{x\in R: 0\le x\le L\}$ tiene una conductividad térmica k(x)=0,5, un perímetro P, una sección transversal A y sus extremos sometidos a temperaturas $T(0)=T_0$ y $T(L)=T_L$. La distribución de temperatura T(x) relativa a la temperatura del medio a lo largo del elemento se puede determinar mediante la solución de la siguiente ecuación diferencial:

$$\frac{d}{dx}\left(k(x)\frac{dT(x)}{dx}\right)+\frac{hc\cdot P}{A}T(x)=S(x), \qquad \text{en } \Omega=\{x\in R: 0\le x\le L\},$$
$$T(0)=To \quad y \quad T(L)=T_L.$$

Aquí hc es el coeficiente de convección; T(x)=Te-T∞, con Te, temperatura del elemento y T∞ la temperatura del medio y S(x) es un término fuente de calor. Unidades: k (W/mK); hc (W/m^2K); P (m); A(m^2); S(W/ m^3); T(K).
Plantee una solución usando derivación numérica y al menos cinco puntos interiores en el dominio **Ω**. Encuentre el sistema de ecuaciones lineales que debe resolverse.

6) Un elemento unidimensional que se encuentra en un dominio $\Omega=\{x\in R: 0\le x\le L\}$ tiene una conductividad térmica k(x)=0,5, un perímetro P, una sección transversal A y sus extremos sometidos a temperaturas $T(0)=T_0$ y $T(L)=T_L$. La distribución de temperatura relativa a la temperatura del medio T(x) a lo largo del elemento se puede determinar mediante la solución de la siguiente ecuación diferencial:

$$\frac{d}{dx}\left(k(x)\frac{dT(x)}{dx}\right)+\frac{hc\cdot P}{A}T(x)=S(x), \qquad \text{en } \Omega=\{x\in R: 0\le x\le L\},$$
$$T(0)=T_0 \quad y \quad \frac{dT}{dx}(L)=0.$$

Aquí hc es el coeficiente de convección; T=Te-T∞, con Te, temperatura del elemento y T∞ la temperatura del medio y S(x) un término fuente de calor. Note que este ejercicio es como el anterior pero con diferente con condición de frontera en x=L ("borde adiabático").
Plantee una solución usando derivación numérica y al menos cinco puntos interiores en el dominio **Ω**. Encuentre el sistema de ecuaciones lineales que debe resolverse. Compare con el ejercicio anterior.

7) Dada la siguiente ecuación diferencial ordinaria con valores en la frontera o condiciones de borde,

$$10\frac{d^2u(x)}{dx^2}+\lambda\cdot u(x)=0 \qquad si\ x\varepsilon\Omega=\{x\varepsilon R: 0\le x\le 1\}$$
$$u(0)=0 \qquad\qquad u(1)=0$$

Usando una aproximación central de la derivada segunda, hallar u(x) en forma discreta para la siguiente discretización propuesta

X	0	0.25	0.5	0.75	1
u(x)	u_0	u_1	u_2	u_3	u_4

Encontrar el sistema de ecuaciones lineales que debe resolverse. Describa que tipo de sistemas de ecuaciones se obtiene en este caso.

8) Las vibraciones libres de de una cuerda que está fija a los puntos x=0 y x=L y que está sometida a una fuerza T en sus extremos, se puede determinar resolviendo la siguiente ecuación diferencial

$$T\frac{d^2v(x)}{dx^2}+\omega^2\cdot\rho(x)\cdot v(x)=0 \qquad si\,x\varepsilon\Omega=\{x\varepsilon R: 0\le x\le L\}$$
$$con\;\; v(0)=v(L)=0$$

Plantear una solución discreta usando derivación numérica y considerando que en el dominio **Ω** los datos disponibles son: L=12; T=500; ρ(x) dada por la siguiente función:

X	0	2,0	4,0	6,0	8,0	10	12
ρ(x)	10	20	40	80	40	20	10

Encontrar el sistema de ecuaciones lineales que debe resolverse. Describa que tipo de sistemas de ecuaciones se obtiene en este caso.

9) La ecuación diferencial en coordenadas polares que determina la frecuencia natural de vibración libre ω de una membrana circular de radio interno ra y radio externo rb, sometida a una tracción uniforme T, densidad por unidad de área ρ, está dada por:

$$T\frac{d^2u(r)}{dr^2}+\frac{T}{r}\frac{du}{dr}+\omega^2\rho\cdot u(r)=0 \qquad si\,r\varepsilon\Omega=\{r\varepsilon R: ra\le r\le rb\}$$
$$con\;\; u(ra)=u(rb)=0$$

Plantear una solución usando derivación numérica y al menos cinco puntos interiores en el dominio **Ω**. Encontrar el sistema de ecuaciones lineales que debe resolverse. Describa que tipo de sistemas de ecuaciones se obtiene en este caso.

10) La ecuación diferencial en coordenadas polares que determina la frecuencia natural de vibración libre ω de una membrana circular de radio externo rb, sometida a una tracción uniforme T, densidad por unidad de área ρ, está dada por:

$$T\frac{d^2u(r)}{dr^2}+\frac{T}{r}\frac{du}{dr}+\omega^2\rho\cdot u(r)=0 \qquad si\,r\varepsilon\Omega=\{r\varepsilon R: 0\le r\le rb\}$$
$$con\;\; \left.\frac{du}{dr}\right|_{r=0}=0 \quad y\;\; u(rb)=0$$

Plantear una solución usando derivación numérica y al menos seis puntos interiores en el dominio **Ω**. Considerar derivada segunda y derivada primera con error del orden de h^2. Encontrar el sistema de ecuaciones lineales que debe resolverse. El valor exacto de la menor frecuencia natural esta dado por $\omega=(2{,}404/rb)\sqrt{T/\rho}$ que se puede consultar en Timoshenko, S., Young D.H. Weaver, W. (1988) "Vibration Problems in Engineering".

11) Para la siguiente ecuación diferencial,

$$
\begin{aligned}
& -EA\,\frac{d^2 v(x)}{dx^2} - \omega^2 \cdot \rho A \cdot v(x) = 0 \qquad si\ x \varepsilon \Omega = \{x \varepsilon R : 0 \le x \le L\} \\
& v(0) = 0 \\
& EA\,\frac{\partial v}{\partial x}(L) = +\omega^2 \cdot M \cdot v(L)
\end{aligned}
$$

Con EA=210000; L=50000; rA=7,85 y M=200, plantear una solución usando derivación numérica y al menos siete puntos interiores en el dominio **Ω**. Encontrar el sistema de ecuaciones lineales que debe resolverse. Describa que tipo de sistemas de ecuaciones se obtiene en este caso.

Ejercicio Integrador de EDO

Se busca *u(x)* solución de

$$
\begin{aligned}
& -\frac{d^2 u(x)}{dx^2} + u(x) - x = 0 \quad en\ \ \Omega = \{x\, \varepsilon\, R : 0 \le x \le 1\} \\
& u(0) = 0 \\
& u(1) = 0
\end{aligned}
$$

La solución exacta de esta ecuación diferencial es

$$u(x) = x - \frac{senh(x)}{senh(1)}$$

En vez de encontrar la solución en cada uno y todos los puntos del dominio Ω, se plantea encontrar una solución aproximada en forma de función discreta sólo algunos puntos elegidos del dominio y equidistantes identificados con su abscisa X_k.

Dividir el dominio en N segmentos iguales, obtener la solución aproximada, aislar el valor de la solución aproximada en x=0,5 y compararla con el valor exacto. Al considerar el error para distintos niveles de *disicretización*; es decir, distinto número de segmentos en que se divide el dominio, se tiene

$$E_N = \left| U_{N/2} - (0{,}5 - \frac{senh(0{,}5)}{senh(1)}) \right| \ \text{y}\ E(abs)_N = \left| U_{N/2} - (0{,}5 - \frac{senh(0{,}5)}{senh(1)}) \right| / (0{,}5 - \frac{senh(0{,}5)}{senh(1)})$$

Cuyas evaluaciones se presentan en la siguiente Tabla

N	Δx	$U_{aprox}(0{,}5)$	E_N	$E(abs)_N$
2	0,5	0,05555556	0,001035002	1,83%
4	0,25	0,05632582	0,000264735	0,47%
8	0,125	0,05652399	6,65711E-05	0,12%
16	0,0625	0,05657389	1,66672E-05	0,03%

Si se asume una relación exponencial entre $E(abs)_N$ y Δx, la aproximación por mínimos cuadrados comprobar que resulta:

$$E(abs)_N = C \cdot \Delta x^P = e^{-2{,}6168} \cdot \Delta x^{1{,}9861}$$

Que indica una relación del orden de $\Delta x^{1{,}9861} \cong \Delta x^2$ que es el error de truncamiento local de la aproximación de derivada segunda utilizada.

Discretización de Ecuaciones Diferenciales a Derivadas Parciales

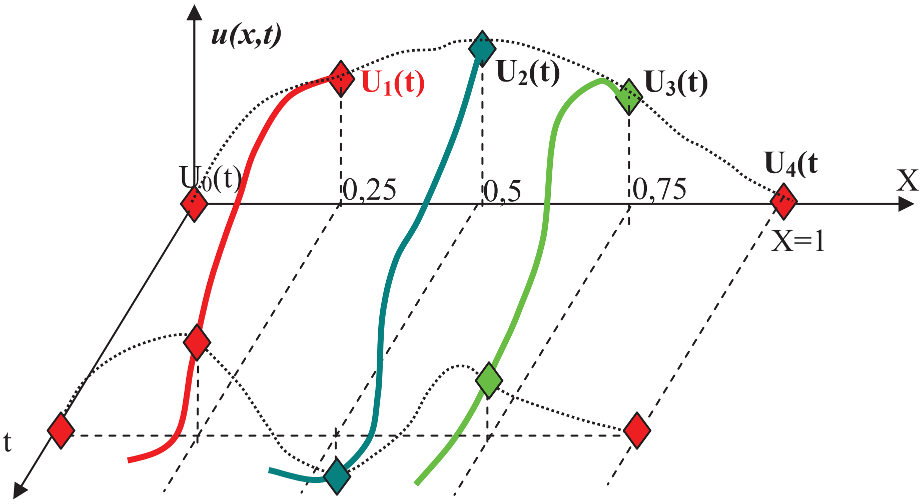
Se busca *u(x,t)* solución de

$$-12\frac{\partial^2 u(x,t)}{\partial x^2}+x^2\frac{\partial^2 u(x,t)}{\partial t^2}=0 \quad en\ \ \Omega=\{x\,\varepsilon\,R:0\le x\le 1\}$$

$$u(0,t)=0 \qquad\qquad u(x,0)=\sin(\pi\cdot x)$$

$$u(1,t)=0 \qquad\qquad \frac{\partial u}{\partial t}(x,0)=3$$

En vez de encontrar la solución exacta *u(x,t)* en cada uno y todos los puntos del dominio Ω, se plantea encontrar una ***solución aproximada en forma de función discreta en la variable x, aunque continua en la variable t***. Se pretende encontrar las funciones $U_k(t)=u(X_k,t)$ con k=0,N, en N+1 puntos elegidos del dominio x, identificados con su abscisa X_k.



Cuando se consideran derivadas parciales respecto a la variable x, para t constante, la función a derivar es una función discreta y se puede hacer derivadas numéricas.

Cuando se consideran derivadas parciales respecto a la variable t, para x constante, la función a derivar es una función continua y se puede hacer derivadas analíticas.

En cada uno de los X_k se puede plantear la ecuación diferencial a resolver pero con una aproximación de la derivada segunda respecto a x en forma de derivada numérica; y con la derivada respecto de la variable t en forma analítica evaluada en esa abscisa X_k.

Así se puede escribir:

en X_0 se debe cumplir que: $$U_0(t)=0$$

en X_1 se debe cumplir que: $$-\frac{12}{0{,}25^2}\left[U_0(t)-2\cdot U_1(t)+U_2(t)\right]+(X_1)^2\cdot\frac{d^2U_1(t)}{dt^2}=0$$

en X_2 se debe cumplir que: $$-\frac{12}{0{,}25^2}\left[U_1(t)-2\cdot U_2(t)+U_3(t)\right]+(X_2)^2\cdot\frac{d^2U_2(t)}{dt^2}=0$$

en X_3 se debe cumplir que: $$-\frac{12}{0{,}25^2}\left[U_2(t)-2\cdot U_3(t)+U_4(t)\right]+(X_3)^2\cdot\frac{d^2U_3(t)}{dt^2}=0$$

en X_4 se debe cumplir que: $$U_4(t)=0$$

o bien en forma del siguiente sistemas de ecuaciones diferenciales ordinarias

$$\frac{12}{0{,}25^2}\begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}\begin{Bmatrix} U_1(t) \\ U_2(t) \\ U_3(t) \end{Bmatrix} + \begin{bmatrix} 0{,}25^2 & 0 & 0 \\ 0 & 0{,}50^2 & 0 \\ 0 & 0 & 0{,}75^2 \end{bmatrix}\begin{Bmatrix} \dfrac{d^2U_1(t)}{dt^2} \\ \dfrac{d^2U_2(t)}{dt^2} \\ \dfrac{d^2U_3(t)}{dt^2} \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \end{Bmatrix}$$

con las condiciones iniciales $\begin{Bmatrix} U_1(0) \\ U_2(0) \\ U_3(0) \end{Bmatrix} = \begin{Bmatrix} sen(\pi \cdot 0{,}25) \\ sen(\pi \cdot 0{,}50) \\ sen(\pi \cdot 0{,}75) \end{Bmatrix}$ y $\begin{Bmatrix} \dfrac{dU_1}{dt}(0) \\ \dfrac{dU_2}{dt}(0) \\ \dfrac{dU_3}{dt}(0) \end{Bmatrix} = \begin{Bmatrix} 3 \\ 3 \\ 3 \end{Bmatrix}$

Este sistema se puede resolver con métodos analíticos o con métodos numéricos.

Ejercicios de Ecuaciones Diferenciales a Derivadas Parciales

12) Dada la siguiente ecuación diferencial,

$$T\frac{\partial^2 u(x,t)}{\partial x^2} = \rho(x)\frac{\partial^2 u(x,t)}{\partial t^2} \quad en\ \ \Omega = \{x\,\varepsilon\,R : 0 \le x \le L\}$$

$$u(0,t) = 0 \qquad\qquad u(x,0) = 0$$

$$u(L,t) = 0 \qquad\qquad \frac{\partial u}{\partial t}(x,0) = \sin(\pi \cdot x / L)$$

Usando una aproximación central de la derivada segunda, y **cinco puntos interiores** al dominio hallar el sistema de ecuaciones que resulta y permite encontrar la solución discreta aproximada de u(x,t).

13) Dada la siguiente ecuación diferencial a derivadas parciales,

$$\frac{\partial^2 u(x,t)}{\partial t^2} = c^2\,\frac{\partial^2 u(x,t)}{\partial x^2} \quad si\ x\varepsilon\Omega = \{x\varepsilon R : 0 \le x \le L\}$$

$$con\ \ u(0,t) = u(L,t) = 0 \quad \forall t$$

considerando una discretización en la variable x de la forma $x_i=x_0+i*\Delta x$ con i=0 hasta 5; manteniendo la continuidad en la variable t, comprobar que se obtiene un sistema que se puede escribir como:

$$\mathbf{M}\,\ddot{\vec{\mathbf{u}}}(t) + \mathbf{K}\,\vec{\mathbf{u}}(t) = 0$$

Con el vector **u** que agrupa las funciones incógnitas de la variable tiempo. Expresar el vector **u**, y las matrices **M** y **K**.

14) Dada la siguiente ecuación diferencial a derivadas parciales,

$$\frac{\partial u(x,t)}{\partial t} = D\frac{\partial^2 u(x,t)}{\partial x^2} \quad si\ x\varepsilon\Omega = \{x\varepsilon R : 0 \le x \le L\}$$

$$con\ \ u(0,t) = u(L,t) = 0 \quad \forall t$$

considerando una discretización en la variable x de la forma $x_i=x_0+i*\Delta x$ con i=0 hasta 5; manteniendo la continuidad en la variable t, comprobar que se obtiene un sistema que se puede escribir como:

$$\vec{\ddot{\mathbf{u}}}(t)+\mathbf{K}\,\vec{\mathbf{u}}(t)=0$$

Con el vector **u** que agrupa las funciones incógnitas de la variable tiempo. Expresar el vector **u**, y las matrices **M** y **K**.

15) Dada la siguiente ecuación diferencial a derivadas parciales,

$$\begin{aligned}
&-k\frac{\partial^2 u(x,t)}{\partial x^2}+m(x)\frac{\partial^2 u(x,t)}{\partial t^2}=p(x)\cdot g(t) \qquad si\ x\varepsilon\Omega=\{x\varepsilon R: 0\le x\le L\}\\
&u(0,t)=0 \qquad \forall t\\
&k(L)\frac{\partial u}{\partial x}(L)=-M_L\frac{\partial^2 u(L,t)}{\partial t^2} \qquad \forall t
\end{aligned}$$

considerando una discretización en la variable x de la forma $x_i=x_0+i*\Delta x$ con i=0 hasta 7; manteniendo la continuidad en la variable t, comprobar que se obtiene un sistema que se puede escribir como:

$$\mathbf{M}\,\vec{\ddot{\mathbf{u}}}(t)+\mathbf{K}\,\vec{\mathbf{u}}(t)=\vec{b}(t)$$

Con el vector **u** que agrupa las funciones incógnitas de la variable tiempo. Expresar el vector **u**, y las matrices **M** y **K**.

Guía de Estudio para Ecuaciones Diferenciales con Valores Iniciales.

Contenido

Guía de Estudio para Ecuaciones Diferenciales con Valores Iniciales. 1
Bibliografía recomendada: 1
Objetivo: 1
Problema de Valores Iniciales de Primer Orden 2
Ejercicios con Método de Euler 3
Ejercicios con Método de Euler en MATLAB 4
Ejercicios con Métodos de Runge Kutta de Segundo Orden 6
Ejercicios con Método de Euler en MATLAB 7
Ejercicios de Sistemas de Ecuaciones Diferenciales de primer Orden 10
Ejercicios de Ecuaciones Diferenciales de Segundo Orden Reducidas a Sistemas de primer orden 15
Ejercicios de Ecuaciones Diferenciales de Segundo Orden con el Método de Diferencia Central 18
Ejercicios Teóricos 21
Ejercicios Integradores 23
Ejercicios con Métodos Predictor Corrector y Multipasos 25

Bibliografía recomendada:

Apuntes de la Cátedra
J. Mathews, J. Fink (2000). *Métodos Numéricos con Matlab.* Prentice Hall.
S. Nakamura (1997). *Análisis Numérico y Visualización Gráfica con MATLAB.* Prentice Hall
S. Nakamura (1992). *Métodos Numéricos Aplicados con Software.* Prentice Hall
S. Chapra, R. Canale (1999). *Métodos Numéricos para Ingenieros*, Mc Graw Hill.
R. Burden, J. Faires (1998). *Análisis Numérico*, International Thomson Editores.

Objetivo:

Se pretende comprender la utilización de métodos numéricos para resolver ecuaciones diferenciales ordinarias con valores iniciales. En particular se busca

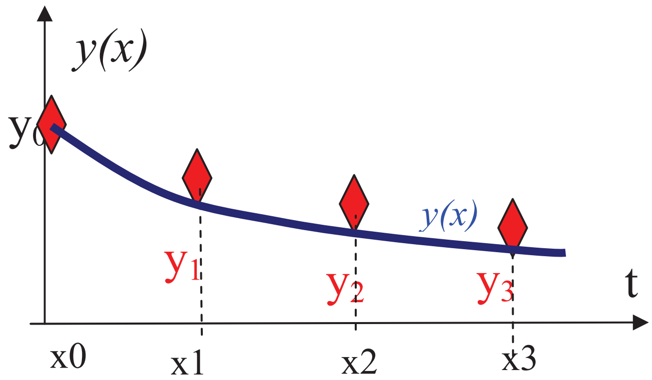
- *Obtener soluciones aproximadas discreta de ecuaciones diferenciales ordinarias de primer orden*
- *Resolver numéricamente sistemas de ecuaciones diferenciales ordinarias de primer orden.*
- *Reducir ecuaciones de segundo orden a sistemas de primer orden, y resolverlos*
- *Resolver sistemas de ecuaciones de segundo orden, como reducción a sistemas de primer orden.*
- *Resolver sistemas de ecuaciones de segundo orden mediante derivación segunda central.*
- *Comprender los métodos de Euler, Runge Kutta de segundo orden y Diferencia Central*
- *Poder implementar dichos métodos en códigos MATLAB.*
- *Obtener soluciones aproximadas discreta de ecuaciones diferenciales a derivadas parciales*

Problema de Valores Iniciales de Primer Orden

Se busca la función $y = y(x)$: $R \to R$ para todo valor de x mayor o igual a x_0, que satisface

$$\frac{dy(x)}{dx} = f(x,y) \qquad \forall\, x \geq x_0$$

$$y(x_0) = y_0$$



donde: *f(x,y)* es una función analítica conocida, [x_0; y_0] es un valor inicial conocido.

Mediante métodos numéricos se busca obtener una solución aproximada de la ecuación diferencial, en forma discreta $y_k \cong y(x_k)$,

Es importante destacar que se asume la existencia y unicidad de la solución de la ecuación diferencial de primer orden, para lo cual se pueden consultar:

Ecuaciones Diferenciales y Problemas con Valores en la Frontera, C. H. Edwards, D. E. Penny, Pearson Educación, Prentice Hall, 2009. ISBN 978-970-26-1285-8

Ecuaciones Diferenciales con Problemas de Valores en la Frontera, D. G. Zill, M. R. Cullen, Thomsom Learning, 2001. ISBN 970-686-133-5

Algunos ejemplos, para los cuales se conoce la solución exacta, son las siguientes ecuaciones diferenciales

$$\frac{dy(x)}{dx} = \frac{x - y(x)}{2} \quad con \quad y(0) = 4 \qquad \text{siendo la solución} \qquad y(x) = 6 \cdot e^{-x/2} - 2 + x$$

$$\frac{dy(x)}{dx} = 1 + (y(x))^2 \quad con \quad y(0) = 0 \quad x\varepsilon\left[0;\pi/2\right) \qquad \text{siendo la solución} \quad y(x) = \tan(x)$$

$$\frac{dy(x)}{dx} = 2 \cdot x \cdot (y(x))^2 \quad con \quad y(0) = 1 \quad x\varepsilon\left[0;1\right) \qquad \text{siendo la solución} \quad y(x) = \frac{1}{1 - x^2}$$

y los siguientes sistemas de ecuaciones diferenciales:

$$\frac{d}{dt}\begin{Bmatrix} u(t) \\ v(t) \end{Bmatrix} = \begin{Bmatrix} v(t) \\ (p/m)sen(\Omega \cdot t) - (k/m) \cdot u(t) \end{Bmatrix} \quad con \quad \begin{Bmatrix} u(0) \\ v(0) \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} \qquad t\ \varepsilon\ \left[0;+\infty\right)$$

cuya solución exacta es

$$\begin{Bmatrix} u(t) \\ v(t) \end{Bmatrix} = \frac{p}{k} \cdot \frac{1}{1 - (\Omega/\omega)^2} \cdot \begin{Bmatrix} -(\Omega/\omega)sen(\omega \cdot t) + sen(\Omega \cdot t) \\ \Omega \cdot (-\cos(\omega \cdot t) + \cos(\Omega \cdot t)) \end{Bmatrix} \quad con \quad \omega^2 = k/m \qquad t\ \varepsilon\ \left[0;+\infty\right)$$

$$\frac{d}{dt}\begin{Bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{Bmatrix} = \begin{Bmatrix} 13 - 9t \\ -15 + 7t \\ 7 - 6t \end{Bmatrix} + \begin{bmatrix} 3 & -2 & 0 \\ -1 & 3 & -2 \\ 0 & -1 & 3 \end{bmatrix} \begin{Bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{Bmatrix} \quad con \quad \begin{Bmatrix} x_1(0) \\ x_2(0) \\ x_3(0) \end{Bmatrix} = \begin{Bmatrix} a \\ b \\ c \end{Bmatrix} \qquad t\ \varepsilon\ \left[0;+\infty\right)$$

cuya solución exacta es

$$\begin{Bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{Bmatrix} = \begin{Bmatrix} 3t \\ 5 \\ 2t \end{Bmatrix} + \begin{bmatrix} 2 \cdot e^{t} & 3 \cdot e^{3t} & 2 \cdot e^{5t} \\ 2 \cdot e^{t} & 0 & -2 \cdot e^{5t} \\ e^{t} & -e^{-3t} & e^{5t} \end{bmatrix} \begin{Bmatrix} c1 \\ c2 \\ c3 \end{Bmatrix} \quad con \quad \begin{Bmatrix} x_1(0) \\ x_2(0) \\ x_3(0) \end{Bmatrix} = \begin{Bmatrix} a \\ b \\ c \end{Bmatrix} \qquad t\ \varepsilon\ \left[0;+\infty\right)$$

Ejercicios con Método de Euler

Ejercicio 1) Dada la ecuación diferencial

$$\frac{dy}{dx} + A \cdot y = 0, \qquad A \in \mathbb{R},$$
$$y(x_0) = y_0,$$

a) Identificar la función $f(x,y)$.
b) Encontrar el valor de las solución en x= 0.6, utilizando método de Euler hacia delante, para A=1 y $[x_0;y_0]$=[0; 10], con un paso h= 0.1. Una forma de organizar el cálculo es utilizando la siguiente tabla

x	y	k1=h*f(x,y)	y(x+h)	$y_{ex}(x)$	Error=abs($y_{ex}(x)$-y_n)
0	10				
0,1					
0,2					
0,3					
0,4					
0,5					
0,6					

c) Repetir el inciso anterior para pasos h=0.3

x	y	k1=h*f(x,y)	y(x+h)	$y_{ex}(x)$	Error=abs($y_{ex}(x)$-y_n)
0	10				
0,3					
0,6					

d) Repetir el inciso anterior para pasos h =0.6.

x	y	k1=h*f(x,y)	y(x+h)	$y_{ex}(x)$	Error=abs($y_{ex}(x)$-y_n)
0	10				
0,6					

e) Comparar las soluciones aproximadas en x=0,6 con el valor exacto que está dado por $y=10.e^{(-x)}$.
f) Obtener un valor mejorado de la aproximación en x=0.6, usando Extrapolación de Richardson y compararlo con la solución exacta.

Ejercicio 2) Dada la ecuación diferencial

$$\frac{dy(x)}{dx} = 2 \cdot y(x) - 2 \cdot x - 1 \qquad con \quad y(0) = 2$$

a) Identificar la función $f(x,y)$.
b) Obtener una solución aproximada en el intervalo [0,1], con el método de Euler adelante y el paso h=0.10:

x	y	k1=h*f(x,y)	y(x+h)	$y_{ex}(x)$	Error=abs($y_{ex}(x)$-y_n)
0	10				
0,1					
0,2					
0,3					
0,4					
0,5					
0,6					
0,7					
0,8					

0,9					
1,0					

c) Obtener una solución aproximada en el intervalo [0,1], con el método de Euler adelante y el paso h=0.25:

x	y	k1=h*f(x,y)	y(x+h)	$y_{ex}(x)$	Error=abs($y_{ex}(x)-y_n$)
0	10				
0,25					
0,5					
0,75					
1,0					

d) Evaluar el error relativo en cada etapa de la resolución de los incisos anteriores sabiendo que la solución exacta es $y_{ex}(x)=e^{2x}+x+1$

e) Para x=0.5 use extrapolación de Richardson, con las dos soluciones obtenidas con el Método de Euler, y comparar con la solución exacta.

Ejercicios con Método de Euler en MATLAB

Ejercicio 3) Dada la ecuación diferencial

$$\frac{dy}{dt} + (1/2)\cdot y = (1/2)\cdot t, \qquad y(0) = 4,$$

a) Identificar la función *f(x,y)*.

b) Elaborar un programa MATLAB que resuelva el ejercicio con el método de Euler, para valores de [*t0; y0*] y *Dt* elegidos

Una posibilidad es el siguiente archivo “Euler_00.m”

```
function Euler_00
% y' =  -(1/2)y + (1/2)t     con y(0)=4
% yex= 6 e^(-t/2) - 2 + t
clc,clear
% Datos
y0=4;      % Valores Iniciales
t0=0;
Dt=0.01;  % incremento de tiempo
NDt=1000;  % cantidad de Dt a realizar
% Dimensionamiento
t=zeros(NDt,1);    % vector para el tiempo
y=zeros(NDt,1);    % vector para solución y(t)
% Inicialización del primer estado solución
t(1)=t0;
y(1)=y0;
%%
% EULER
for j=1:NDt-1
    k1=Dt*(-(1/2)*y(j) + (1/2)*t(j));
    y(j+1)  = y(j)+k1;
    t(j+1)  = t(j)+Dt;
    %%
end
%
figure(1)
plot(t,y(:,1),'b');
end
```

c) Analizando el código anterior determinar:
- cual es la dimensión de los vectores que se grafican
- cual es el intervalo de la variable independiente donde se grafica la solución obtenida
- identificar las líneas del código que "trabajan" con los valores iniciales
- identificar las líneas del código que "trabajan" con la función f(t,y) asociada a la ecuación diferencial de primer orden que se resulte.

d) Modificar el programa MATLAB elaborado anteriormente para incluir el cálculo de la solución exacta en los mismos valores de la variable independiente t en lo que se calcula la solución aproximada con el método de Euler. Para ello se debe considerar la incorporación de al menos las siguientes lineas de código:

```
yex(j+1)=6*exp(-t(j+1)/2) - 2 + t(j+1);
```

e) Obtener la función error como la diferencia de ambas soluciones y comprobar que la gráfica de la función error es la siguiente y que la norma infinito es 5.5 E-03.
Para ello se debe considerar la incorporación de al menos las siguientes líneas de código

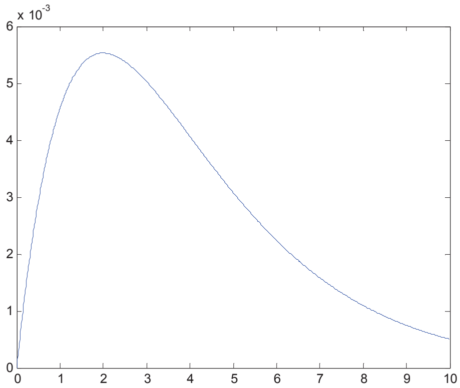
```
yex(j)=6*exp(-t(j)/2) - 2 + t(j);
er(j)=abs(yex(j)-y(j));
```

o bien

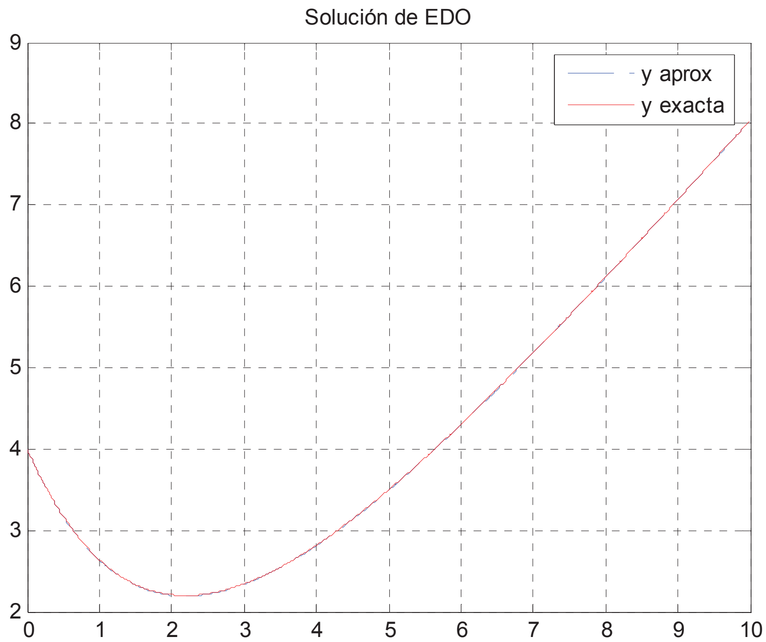
```
yex(j+1)=6*exp(-t(j+1)/2) - 2 +
t(j+1);
er(j+1)=abs(yex(j+1)-y(j+1));
```

y de

```
normer=norm(er,inf)
```



f) Comprobar que las graficas de la solución aproximada con Euler y la de la solución exacta son las siguientes:



g) Modificar el programa MATLAB elaborado anteriormente para considerar la siguiente *function*:

```
function fy=f_ord_1(z,x)
  fy=(-(1/2)*z + (1/2)*x);
end
```

cambiar la forma de calcular k1 y obtener los mismos resultados. Discutir la ventaja de la *function*.

Ejercicios con Métodos de Runge Kutta de Segundo Orden

Los métodos de Runge Kutta de Segundo orden se pueden expresar mediante el siguiente algoritmo

$$Conocidos\left(x_m, y_m\right) \;\; y \;\; asumidos \;\; (h, \; \omega) \;\; se \;\; calcula$$

$$k_1 = h\, f\left(x_m, y_m\right)$$

$$x_G = x_m + \frac{h}{2\omega} \qquad y_G = y_m + \frac{1}{2\omega} k_1$$

$$k_2 = h\, f(x_G, y_G)$$

$$y_{m+1} = y_m + (1-\omega)\, k_1 + \omega\, k_2$$

$$x_{m+1} = x_m + h$$

Ejercicio 4) Dada la ecuación diferencial

$$\frac{dy(x)}{dx} = 2 \cdot y(x) - 2 \cdot x - 1 \qquad con \quad y(0) = 2$$

a) Identificar la función *f(x,y)*.
b) Obtener una solución aproximada en el intervalo [0,1], con el método de Euler Mejorado y el paso h=0.10:

x	y	k1= h*f(x,y)	xg= x + h/(2w)	yg= y+k1/(2w)	k2= h*f(x,y)	Y(x+h)= y+(1-w)k1+w*k2
0	10					
0,1						
0,2						
0,3						
0,4						
0,5						
0,6						
0,7						
0,8						
0,9						
1,0						

c) Obtener una solución aproximada en el intervalo [0,1], con el método de Euler Mejorado y los siguientes pasos h=0.25

x	y	k1= h*f(x,y)	xg= x + h/(2w)	yg= y+k1/(2w)	k2= h*f(x,y)	Y(x+h)= y+(1-w)k1+w*k2
0	10					
0,25						
0,50						
0,75						
1,0						

d) Evaluar el error relativo en cada etapa de la resolución de los incisos anteriores sabiendo que la solución exacta es $y_{ex}(x)=e^{2x}+x+1$
e) Para x=0.5 use extrapolación de Richardson, con las dos soluciones obtenidas con el Método de Euler Mejorado, y comparar con la solución exacta.

f) Rehacer el ejercicio completo con el Método de Euler Modificado. Comparar con los resultados obtenidos con el Método de Euler Mejorado.

Ejercicios con Método de Euler en MATLAB

Ejercicio 5) Dada la ecuación diferencial

$$\frac{dy}{dt} + (1/2)\cdot y = (1/2)\cdot t, \qquad y(0) = 4,$$

a) Identificar la función *f(x,y)*.
b) Elaborar un programa MATLAB que resuelva el ejercicio con el método de Euler Mejorado o Modificado, para valores de [*t0; y0*] y *Dt* elegidos
Una posibilidad es el siguiente archivo "RK2_00.m"

```
function RK2_00
% y' =  -(1/2)y + (1/2)t    con y(0)=4
% yex= 6 e^(-t/2) - 2 + t
clc,clear
% Datos
y0=4;      % Valores Iniciales
t0=0;
Dt=0.01;   % incremento de tiempo
NDt=1000;   % cantidad de Dt a realizar
w=0.5;     % w=0.5 es Euler Mejorado
%          % w=1   es Euler Modificado;
% Dimensionamiento
t=zeros(NDt,1);    % vector para el tiempo
y=zeros(NDt,1);    % vector para solución y(t)
% Inicialización del primer estado solución
t(1)=t0;
y(1)=y0;
% Runge-Kutta 2do Orden
for j=1:NDt-1
    k1=Dt*(-(1/2)*y(j) + (1/2)*t(j));
      yg    = y(j) + k1/(2*w);
      tg    = t(j) + Dt/(2*w);
    k2=Dt*(-(1/2)*yg + (1/2)*tg);
    y(j+1)  = y(j)+(1-w)*k1+w*k2;
    t(j+1)  = t(j)+Dt;
end
figure(1)
plot(t,y(:,1),'b');
grid on
end
```

h) Analizando el código anterior determinar:
   - cual es la dimensión de los vectores que se grafican
   - cual es el intervalo de la variable independiente donde se grafica la solución obtenida
   - identificar las líneas del código que "trabajan" con los valores iniciales
   - identificar las líneas del código que "trabajan" con la función f(t,y) asociada a la ecuación diferencial de primer orden que se resulte.

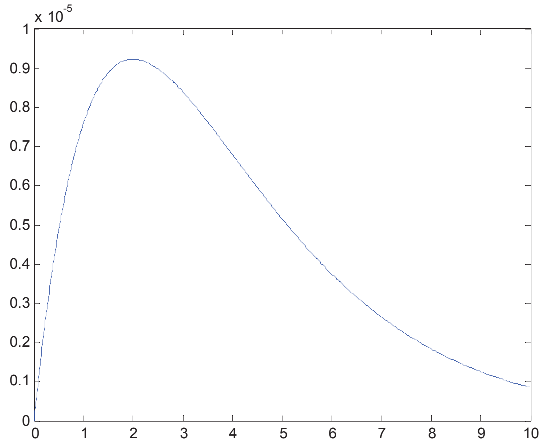
i) Modificar el programa MATLAB elaborado anteriormente para incluir el cálculo de la solución exacta en los mismos valores de la variable independiente t en lo que se calcula la solución aproximada con el método de Euler. Para ello se debe considerar la incorporación de al menos las siguientes líneas de código:

```
for i=1:NDt
    yex(i)=6*exp(-t(i)/2) - 2 + t(i);
    er(i)=abs(yex(i)-y(i));
end
```

j) Obtener la función error como la diferencia de ambas soluciones y comprobar que la gráfica de la función error es la siguiente y que la norma infinito es 1.77 E-04.

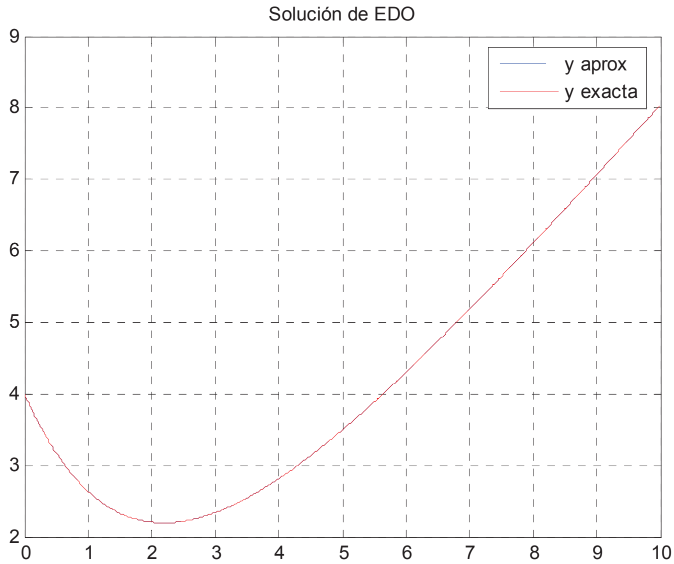
Para ello se debe considerar la incorporación de al menos las siguientes líneas de código

```
for i=1:NDt
  yex(i)=6*exp(-t(i)/2)-2+t(i);
  er(i)=abs(yex(i)-y(i));
end
normer=norm(er,inf)
figure(3)
plot(t,er)
```



Comparar con la misma solución obtenido con el método de Euler y el mismo paso.

k) Comprobar que las graficas de la solución aproximada con Euler Mejoradoy la de la solución exacta son las siguientes:



Se pueden considerar las siguientes líneas de código

```
figure (4)
plot (t,y(:,1),'--b',t,yex,'r' )
grid on
title ('Solución de EDO')
legend('y aprox', 'y exacta')
```

l) Modificar el programa MATLAB elaborado anteriormente para considerar la siguiente *function*:

```
function fy=f_ord_1(z,x)
  fy=(-(1/2)*z + (1/2)*x);
end
```

cambiar la forma de calcular k1 , k2 y obtener los mismos resultados. Discutir la ventaja de la *function*.

Ejercicio 6) Dado el siguiente código en MATLAB

```
function zz
clc,clear
y0=4;
t0=0;
Dt=0.02;
NDt=500;
w=0.0;

t=zeros(NDt,1);
y=zeros(NDt,1);
t(1)=t0;
y(1)=y0;

for j=1:NDt-1
   k1=Dt*f_ord_1(y(j),t(j));
   if (w==0)
      k2=0;
   else
      yg    = y(j) + k1/(2*w);
      tg    = t(j) + Dt/(2*w);
      k2=Dt*f_ord_1(yg,tg);
   end
   y(j+1)  = y(j)+(1-w)*k1+w*k2;
   t(j+1)  = t(j)+Dt;
end

figure(1)
plot(t,y(:,1),'b');
grid on
end

function fy=f_ord_1(z,x)
  fy=(-(1/2)*z + (1/2)*x);
end
```

- ¿cual es la dimensión de los vectores que se grafican?
- ¿cual es el valor del primer punto que se grafica?
- el parámetro "w" igual a cero o distinto de cero, selecciona el accionar en el if-else-end. Para cuando es nulo ¿qué método queda definido?. ¿y para cuando es 0.5?
- ¿cual es el significado de la grafica que se realiza?
- si se pretende resolver con este código la EDO $\frac{dy}{dx} = -2 \cdot y(x) \quad con \quad y(1) = 25$

con el método de Euler Modificado, ¿Cuáles son las líneas que se deben cambiar?

Ejercicios de Sistemas de Ecuaciones Diferenciales de primer Orden

Ejercicio 7) Dado el siguiente sistema de ecuaciones diferenciales

$$\frac{dy_1}{dt} = -10 \cdot y_1(t) + 4 \cdot y_2(t) \qquad y_1(0) = 5$$
$$\frac{dy_2}{dt} = -4 \cdot y_1(t) + 0 \cdot y_2(t) \qquad y_2(0) = 3$$

, con las condiciones iniciales: $y_1(0) = 5$, $y_2(0) = 3$

La solución exacta del problema es: $\mathbf{y}(t) = \frac{1}{3}\begin{Bmatrix}1\\2\end{Bmatrix} e^{-2*t} + \frac{14}{3}\begin{Bmatrix}1\\1/2\end{Bmatrix} e^{-8*t}$

Aplicar el Método de Euler y obtener la siguiente aproximación

t	**0**	0,01	0,02	0,03	0,04	0,05
y1	**5**					
y2	**3**					
k1 _y1	0,01*(-10***5**+4***3**)					
_y2	0,01*(-4***5**+0***3**)					
y1_(n+1)	**5**+(-0,38)					
y2_(n+1)	**3**+(-0,2)					
tn+1	**0**+0,1					

Considerar el siguiente programa MATLAB

```
function RK2_03_con_Euler_Vectorial
% y1' =  (-10y1 + 4 y2)     con y1(0)=5
% y2' =  (-4 y1 + 0 y2)     con y2(0)=3
% yex1= (1/3)*exp(-2*t)+(14/3)*exp(-8*t));
% yex2= (2/3)*exp(-2*t)+(7/3)*exp(-8*t));
clc,clear
% Datos
y10=5;      % Valores Iniciales
y20=3;
t0=0;
Dt=0.01;  % incremento de tiempo
NDt=200;  % cantidad de Dt a realizar
% Dimensionamiento
t=zeros(1,NDt);    % vector fila para el tiempo
y=zeros(2,NDt);    % Matriz para el vector solución y(t)
ya=zeros(2,1);     % Vector columna auxiliar
k1=zeros(2,1);
% Inicialización del primer estado solución
t(1)=t0;
y(1,1)=y10;
y(2,1)=y20;
% Euler
for j=1:NDt-1
   ya=y(:,j);
   ta=t(j);
   k1=Dt*f_sist_1(ya,ta);
   y(:,j+1)  = ya + k1;
   t(j+1)    = ta + Dt;
end
figure(1)
plot(t,y(1,:),'b');grid on
end
function [fy]=f_sist_1(z,x)
  fy(1,1)=(-10*z(1) + 4*z(2));
  fy(2,1)=(-4*z(1)+0*z(2));
end
```

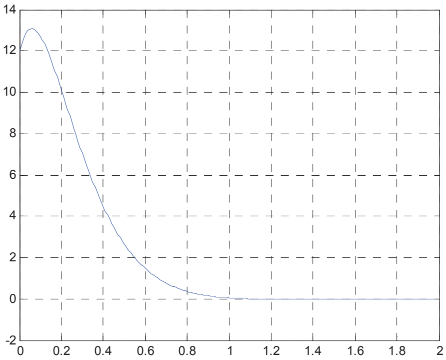
- Notar que los valores de la variable tiempo se "guardan" en un vector fila.
- Las soluciones y1(t) e y2(t) se "guardan" en una matriz y(2,NDt) donde cada columna tiene el vector solución para un tiempo determinado.
- La *function* `[fy]=f_sist_1(z,x)` calcula el vector de derivadas primeras
- Identificar las líneas de código correspondientes al cálculo con el método de Euler.
- Identificar las líneas donde se grafica la solución para y1(t).
- Utilizando el programa MATLAB, verificar los cálculos realizados para completar la Tabla anterior.

Ejercicio 8) Para el siguiente sistema de ecuaciones lineales de primer orden, encuentre las soluciones aproximadas con el Método de Euler a partir de la solución inicial, modificando el programa anterior.

$$\frac{dy_1}{dt} = -8\cdot y_1(t) + 6\cdot y_2(t)\,,\qquad \frac{dy_2}{dt} = -2\cdot y_1(t) + 9\cdot y_2(t)$$

con las condiciones iniciales: $y_1(0) = 12$, $y_2(0) = 22$

Para un Dt=0,01 comprobar que la $y_1(t)$ tiene la siguiente gráfica que se presenta en la figura.



Ejercicio 9) Dado el siguiente sistema de ecuaciones diferenciales

$$\frac{dy_1}{dt} = -10\cdot y_1(t) + 4\cdot y_2(t)\,,\qquad \frac{dy_2}{dt} = -4\cdot y_1(t) + 0\cdot y_2(t)$$

, con las condiciones iniciales: $y_1(0) = 5$, $y_2(0) = 3$

La solución exacta del problema es: $\mathbf{y}(t) = \frac{1}{3}\begin{Bmatrix}1\\2\end{Bmatrix} e^{-2*t} + \frac{14}{3}\begin{Bmatrix}1\\1/2\end{Bmatrix} e^{-8*t}$

Aplicar el Método de Euler Modificado (w=1) y obtener la siguiente aproximación

t	**0**	0,01	0,02	0,03	0,04
y1	**5**	4,635	4,2977		
y2	**3**	2,8076	2,6292		
k1 _y1	0,01*(-10***5**+4***3**)=-0,38				
_y2	0,01*(-4***5**+0***3**)=-0,2				
tg	**0**+0,01/2=0,005				
yg1	**5** -0,38/2= **4,81**				
yg2	**3** -0,2/2= **2,9**				
k2 _y1	0,01*(-10***4,81**+4***2,9**)= -0,365				
_y2	0,01*(-4***4,81**+0***2,9**)= -0,1924				
tg	**0**+0,01				
y1 _(n+1)	**5** -0,365= **4,635**				
y2 _(n+1)	**3** -0,1924= **2,8076**				

Considerar el siguiente programa MATLAB, que es una variante del presentado en el Ejercicio 7

```
function RK2_03_con_Euler_Vectorial
% y1' =  (-10y1 + 4 y2)    con y1(0)=5
% y2' =  (-4 y1 + 0 y2)    con y2(0)=3
% yex1= (1/3)*exp(-2*t)+(14/3)*exp(-8*t));
% yex2= (2/3)*exp(-2*t)+(7/3)*exp(-8*t));
clc,clear
% Datos
y10=5;     % Valores Iniciales
y20=3;
t0=0;
Dt=0.01;  % incremento de tiempo
NDt=200;  % cantidad de Dt a realizar
% Dimensionamiento
t=zeros(1,NDt);   % vector fila para el tiempo
y=zeros(2,NDt);   % Matriz para el vector solución y(t)
yg=zeros(2,1);    % Vector columna auxiliar
ya=zeros(2,1);
k1=zeros(2,1);
k2=zeros(2,1);
% Inicialización del primer estado solución
t(1)=t0;
y(1,1)=y10;
y(2,1)=y20;
% Euler Modificado
for j=1:NDt-1
   ya=y(:,j);
   ta=t(j);
   k1=Dt*f_sist_1(ya,ta)

   yg    = ya + k1/(2);
   tg    = ta + Dt/(2);
   k2=Dt*f_sist_1(yg,tg);

   y(:,j+1)  = ya + k2;
   t(j+1)    = ta + Dt;
end
for i=1:NDt
    yex(i)=(1/3)*exp(-2*t(i))+(14/3)*exp(-8*t(i));
    er(i)=abs(yex(i)-y(1,i));
end
normer=norm(er,inf)
figure(1)
plot(t,y(1,:),'b');
grid on
end
function [fy]=f_sist_1(z,x)
  fy(1,1)=(-10*z(1)  + 4*z(2));
  fy(2,1)=(-4*z(1)+0*z(2));
end
```

- Las soluciones y1(t) e y2(t) se “guardan” en una matriz y(2,NDt) donde cada columna tiene el vector solución para un tiempo determinado.
- La *function* `[fy]=f_sist_1(z,x)` calcula el vector de derivadas primeras
- Identificar las líneas de código correspondientes al cálculo con el método de Euler Modificado.
- Identificar las líneas donde se calcula la solución exacta de y1(t) y el error cometido.
- Utilizando el programa MATLAB, verificar los cálculos realizados para completar la Tabla anterior.
- Para otro sistema de EDO de 2 ecuaciones con dos incógnitas, ¿Cuáles son las lineas que se deben cambiar?

Ejercicio 10) Para el siguiente sistema de ecuaciones lineales de primer orden, encuentre las soluciones aproximadas a partir de la solución inicial, y compare con la solución exacta modificando el programa anterior.

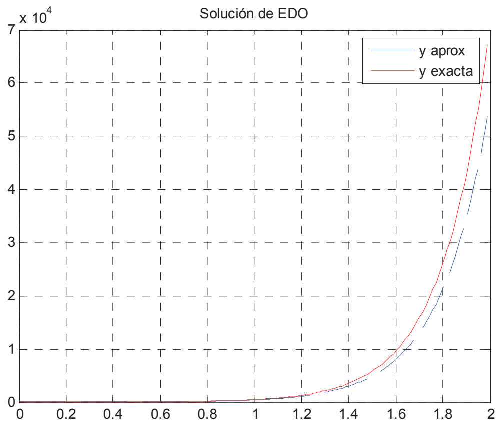
$$\frac{d}{dt}\begin{Bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{Bmatrix} = \begin{Bmatrix} 13-9t \\ -15+7t \\ 7-6t \end{Bmatrix} + \begin{bmatrix} 3 & -2 & 0 \\ -1 & 3 & -2 \\ 0 & -1 & 3 \end{bmatrix}\begin{Bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{Bmatrix} \quad con \quad \begin{Bmatrix} x_1(0) \\ x_2(0) \\ x_3(0) \end{Bmatrix} = \begin{Bmatrix} 5 \\ 3 \\ 2 \end{Bmatrix} \quad t\ \varepsilon\ [0;+\infty)$$

cuya solución exacta es

$$\begin{Bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{Bmatrix} = \begin{Bmatrix} 3t \\ 5 \\ 2t \end{Bmatrix} + \begin{bmatrix} 2\cdot e^{t} & 3\cdot e^{3t} & 2\cdot e^{5t} \\ 2\cdot e^{t} & 0 & -2\cdot e^{5t} \\ e^{t} & -e^{-3t} & e^{5t} \end{bmatrix}\begin{Bmatrix} 0,6 \\ 0,2 \\ 1,6 \end{Bmatrix} \quad con \quad \begin{Bmatrix} x_1(0) \\ x_2(0) \\ x_3(0) \end{Bmatrix} = \begin{Bmatrix} 5 \\ 3 \\ 2 \end{Bmatrix} \quad t\ \varepsilon\ [0;+\infty)$$

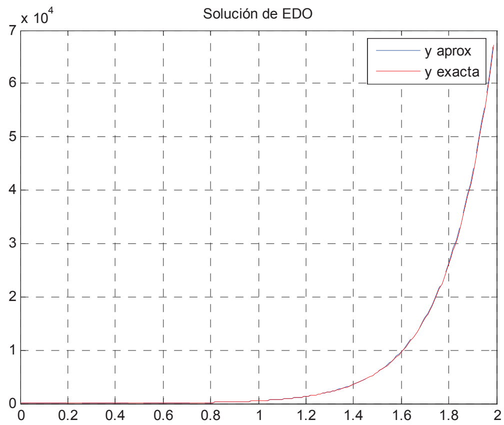
Comprobar que con un Dt=0,01 se logran las siguientes gráficas
y que la norma infinito de la función error entre $y_1(t)$ y la $y_{1exacta}(t)$ es la indicada debajo de cada figura.

Con Euler



Norma Infinito del error=1,3578E+4

Con Euler Modificado



Norma Infinito del error=737.26

Ejercicio 11) Para el siguiente sistema de ecuaciones lineales de primer orden, encuentre las soluciones aproximadas a partir de la solución inicial, para los distintos métodos y pasos h indicados.

$$\frac{dy_1}{dt} = 1\cdot y_2 \qquad \frac{dy_2}{dt} = -100\cdot y_1$$

, con las condiciones iniciales:

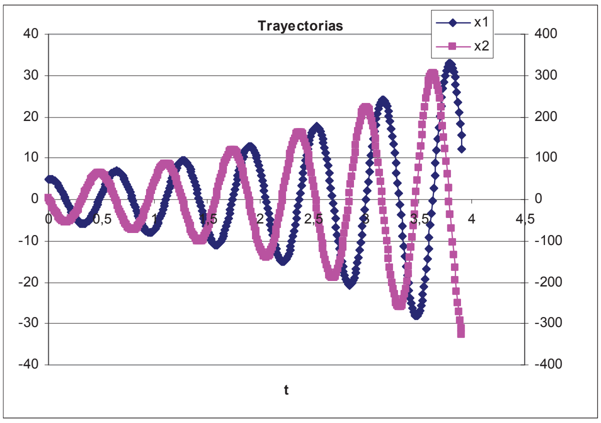
$$y_1(0) = 5 \qquad y_2(0) = 3$$

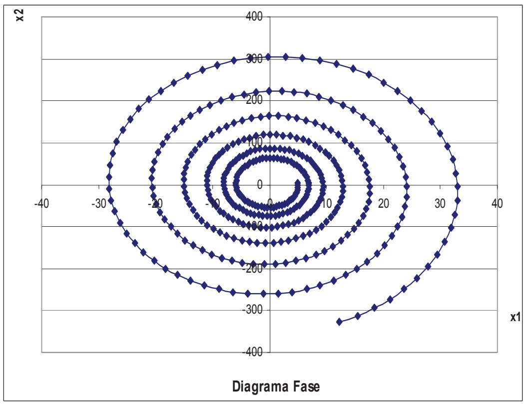
a) Comprobar que con h=0,01 las soluciones encontradas son:

	Método de Euler		Método de Euler Mejorado	
t	x1	x2	x1	x2
0	5	3	5	3
1,00E-02	5,03E+00	-2,00E+00	5,01E+00	-2,02E+00
2,00E-02	5,01E+00	-7,03E+00	4,96E+00	-7,01E+00
3,00E-02	4,94E+00	-1,20E+01	4,86E+00	-1,19E+01
4,00E-02	4,82E+00	-1,70E+01	4,72E+00	-1,67E+01
5,00E-02	4,65E+00	-2,18E+01	4,53E+00	-2,14E+01
6,00E-02	4,43E+00	-2,64E+01	4,29E+00	-2,58E+01

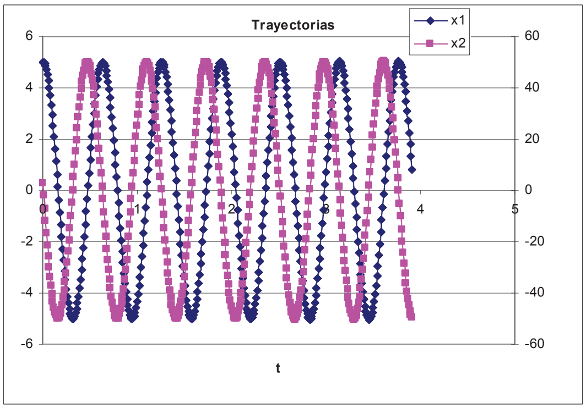
b) Comprobar que las denominadas Trayectorias (curvas y1(t) e y2(t); y la respuesta en el espacio de estado (curva y2 como función de y1) son las siguientes:

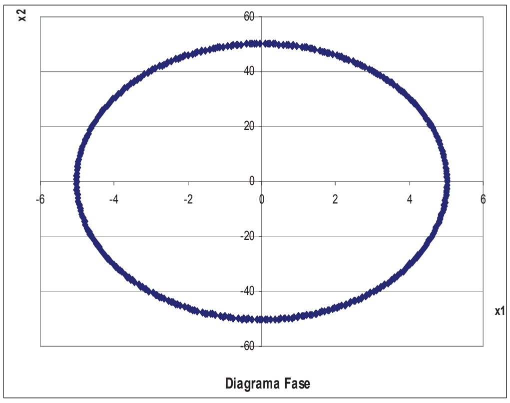
Con el Método de Euler y h=0,01





Con el Método de Euler Modificado y h=0,01





Ejercicios de Ecuaciones Diferenciales de Segundo Orden Reducidas a Sistemas de primer orden

Ejercicio 12) Considérese la ecuación diferencial de segundo orden del péndulo simple,

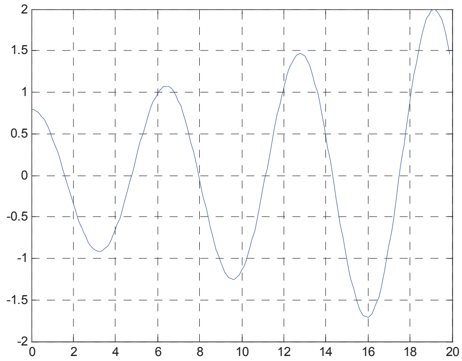
$$\frac{d^2\theta}{dt^2} + \frac{g}{L}\theta = 0 \qquad \theta(t_0) = \theta_0$$

$$\frac{d\theta}{dt} = \beta_0 \ \ en\ t = t_0$$

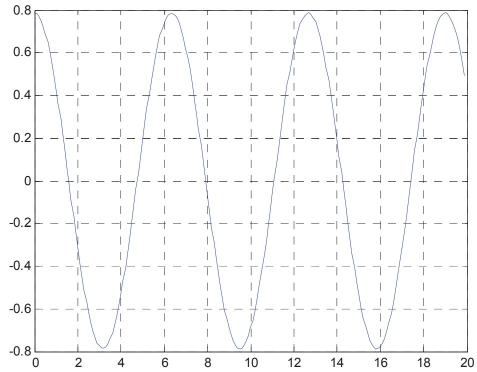
donde $\theta(t)$ es la posición angular medida respecto de la vertical; θ_0 la posición inicial y β_0 la velocidad inicial.
Verifique que es posible transformar dicha ecuación de segundo orden en un sistema de primer orden cuyo vector de incógnitas es $y(t) = \begin{Bmatrix} \theta(t) \\ \alpha(t) \end{Bmatrix}$,

a) encuentre el vector *f (t, y)* que resulta de escribir el sistema de primer orden anterior, al considerar la definición del vector *y(t);* y el vector de valores iniciales *y(0)*.

b) Resuelva con el método de Euler hacia adelante. Tome como datos $\theta(0)=\pi/4$; $\alpha(0)=0$; L= 10; g=9.81 y h=0.1 y resuelva hasta t=10.
   - Comprobar a siguiente gráfica para $\theta(t)$.
   - Verifique si el valor de $\theta(3.1)$ es próximo a $-\pi/4$.
   - Estimar el valor del período.



c) Resuelva con el método de Euler Modificado. Tome como datos $\theta(0)=\pi/4$; $\alpha(0)=0$; L= 10; g=9.81 y h=0.1 y resuelva hasta t=10.
   - Comprobar a siguiente gráfica para $\theta(t)$.
   - Verifique si el valor de $\theta(3.1)$ es próximo a $-\pi/4$.
   - Estimar el valor del período.



d) Asociar los resultados anteriores con el período $2\pi\sqrt{L/g} = 2\pi\sqrt{10/9{,}810} = 6{,}34 seg$.

e) La calidad de la respuesta obtenida depende del valor del paso de tiempo asumido. Corroborar la calidad de la respuesta verificando la conservación de la energía mecánica, que está dada por:

$$\frac{1}{2}m\left(L\cdot\frac{d\theta}{dt}\right)^2 + mgL(1-\cos(\theta(t)) = \frac{1}{2}m\left(L\cdot\beta_0\right)^2 + mgL(1-\cos(\theta_0) = Em0$$

Ejercicio 13) La posición *u(t)* de una masa m unida a un resorte de constante elástica k, donde actúa una carga sin(Ω*t), siendo Ω la frecuencia de excitación, viene dada por la ecuación diferencial:

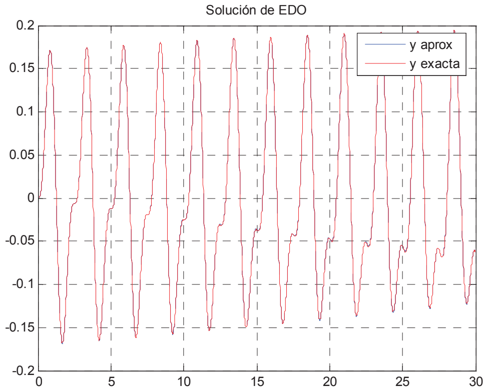
$$m\frac{d^2u}{dt^2}+k\cdot u=sen(\Omega\cdot t) \qquad u(t_0)=u_0$$

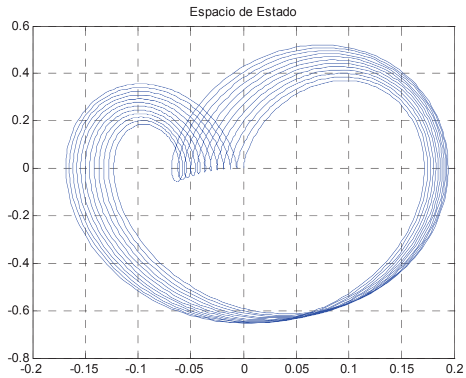
$$\frac{du}{dt}=v_0 \;\; en\; t=t_0$$

siendo u_0 la posición inicial y v_0 la velocidad inicial.

a) Transforme esta ecuación de segundo orden en un sistema de primer orden, identificando el vector *y(t)* , sus valores iniciales y la *f(t,y)*.

b) Resolver este sistema mediante un método de Runge Kutta de segundo orden, para m=65; k=400; c=0; p=80; Ω=5; u_0=v_0=0; Dt=0.01. Comprobar las siguientes gráficas:



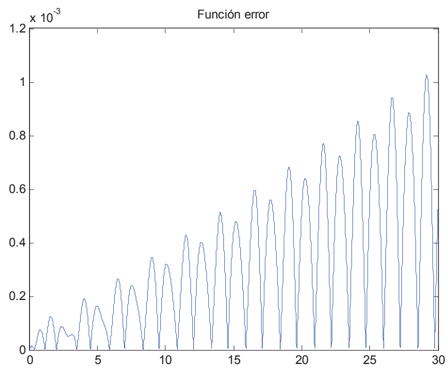


Se define espacio de estado a la gráfica de la velocidad en función de la posición *u(t)*.

La solución exacta es

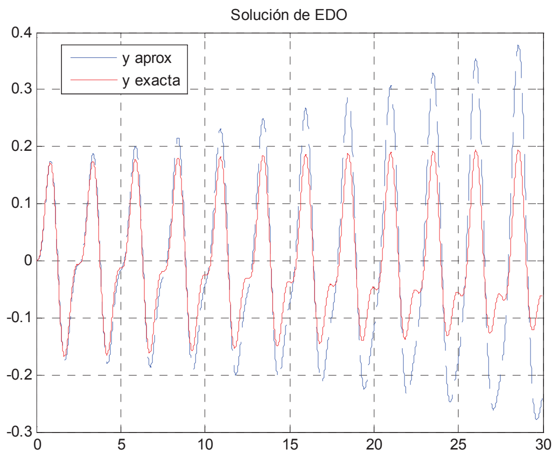
$$u_{ex}(t)=\frac{p}{k}\cdot\frac{1}{(1-(\Omega/\omega s)^2)}(\sin(\Omega*t)-\frac{\Omega}{\omega s}\cdot\sin(\omega s*t))$$

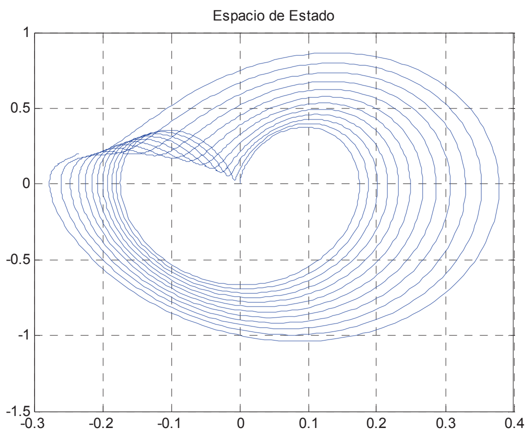
$$con \quad \omega s=\sqrt{k/m}$$



Al superponer las gráficas de la solución exacta y de la aproximada no hay diferencia a simple vista; pero al graficar el error en función del tiempo se obtiene la gráfica adjunta, de donde se puede observar que el error es menor a 1.1E-3 en el intervalo analizado.

c) Resolver este sistema mediante un método de Euler, para los mismos datos. Comprobar las siguientes gráficas y que la función error es menor a 0.2.





d) La calidad de la respuesta obtenida depende del valor del paso de tiempo asumido. Corroborar la calidad de la respuesta verificando la conservación del teorema de Impulso y Cantidad de Movimiento que está dada por:

$$\int_{t1}^{t2} \sum F(t) \cdot dt = m\left.\frac{du}{dt}\right|_{t2} - m\left.\frac{du}{dt}\right|_{t1}$$

Ejercicio 14) Dada la siguiente ecuación diferencial de segundo orden,

$$\frac{d^2u(t)}{dt^2} + 23 \cdot \frac{du(t)}{dt} + (45 + 12 \cdot u(t)^2) \cdot u(t) = 12 \cdot \sin(6 \cdot t) \qquad u(3) = -32$$

$$\frac{du(t)}{dt} = 43 \; en \; t = 3$$

Transformar dicha ecuación de segundo orden en un sistema de primer orden cuyo vector de incógnitas es *y(t)*; y encontrar el vector de valores iniciales *y(0)* y el vector *f (t, y)* que resulta de escribir el sistema de primer orden.

Ejercicio 15) Dada la siguiente ecuación diferencial de segundo orden,

$$\frac{d^2\theta}{dt^2} + \frac{g}{L}\sin(\theta(t)) = 0 \qquad \theta(t_0) = \theta_0$$

$$\frac{d\theta}{dt} = \beta_0 \; en \; t = t_0$$

Transformar dicha ecuación de segundo orden en un sistema de primer orden cuyo vector de incógnitas es *y(t)*; y encontrar el vector de valores iniciales *y(0)* y el vector *f (t, y)* que resulta de escribir el sistema de primer orden.

Ejercicio 16) Dada la siguiente ecuación diferencial de segundo orden

$$16\frac{d^2\theta_1(t)}{dt^2} + 4(\theta_2 + \theta_1) = 5t \qquad 32\frac{d^2\theta_2(t)}{dt^2} + 9\theta_1 = -3t^2$$

$$\text{con} \begin{Bmatrix} \theta_1 \\ \theta_2 \end{Bmatrix}_{t0} = \begin{Bmatrix} \pi/8 \\ -\pi/4 \end{Bmatrix} \; y \; \begin{Bmatrix} \dfrac{d\theta_1}{dt} \\ \dfrac{d\theta_2}{dt} \end{Bmatrix}_{t0} = \begin{Bmatrix} 0{,}1 \\ -0{,}2 \end{Bmatrix}$$

Transformar dicha ecuación de segundo orden en un sistema de primer orden cuyo vector de incógnitas es *y(t)*; y encontrar el vector de valores iniciales *y(0)* y el vector *f (t, y)* que resulta de escribir el sistema de primer orden.

Ejercicio 17) Dada la siguiente ecuación diferencial de segundo orden,

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 2 \end{bmatrix}\begin{Bmatrix} \ddot{x}_1(t) \\ \ddot{x}_2(t) \\ \ddot{x}_3(t) \end{Bmatrix} + \begin{bmatrix} 4 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 3 \end{bmatrix}\begin{Bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \end{Bmatrix} + \begin{bmatrix} 0 & 4 & 1 \\ 4 & 2 & 0 \\ 1 & 0 & 1 \end{bmatrix}\begin{Bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{Bmatrix} = \begin{Bmatrix} 5e^t + 8e^{2t} + \cos t \\ -8e^{2t} + 4e^t \\ -\cos t - 3sent + e^t \end{Bmatrix}$$

$$con \begin{Bmatrix} x_1(0) \\ x_2(0) \\ x_3(0) \end{Bmatrix} = \begin{Bmatrix} 1 \\ 2 \\ 1 \end{Bmatrix} \qquad \begin{Bmatrix} \dot{x}_1(0) \\ \dot{x}_2(0) \\ \dot{x}_3(0) \end{Bmatrix} = \begin{Bmatrix} 1 \\ 4 \\ 0 \end{Bmatrix}$$

Transformar dicha ecuación de segundo orden en un sistema de primer orden cuyo vector de incógnitas es *y(t)*; y encontrar el vector de valores iniciales *y(0)* y el vector *f (t, y)* que resulta de escribir el sistema de primer orden.

Ejercicios de Ecuaciones Diferenciales de Segundo Orden con el Método de Diferencia Central

Ejercicio 18) Dado el siguiente sistema de ecuaciones diferenciales de segundo orden

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 2 \end{bmatrix} \begin{Bmatrix} \ddot{x}_1(t) \\ \ddot{x}_2(t) \\ \ddot{x}_3(t) \end{Bmatrix} + \begin{bmatrix} 4 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 3 \end{bmatrix} \begin{Bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \end{Bmatrix} + \begin{bmatrix} 0 & 4 & 1 \\ 4 & 2 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{Bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{Bmatrix} = \begin{Bmatrix} 5e^t + 8e^{2t} + \cos t \\ -8e^{2t} + 4e^t \\ -\cos t - 3sent + e^t \end{Bmatrix}$$

$$con \quad \begin{Bmatrix} x_1(0) \\ x_2(0) \\ x_3(0) \end{Bmatrix} = \begin{Bmatrix} 1 \\ 2 \\ 1 \end{Bmatrix} \qquad \begin{Bmatrix} \dot{x}_1(0) \\ \dot{x}_2(0) \\ \dot{x}_3(0) \end{Bmatrix} = \begin{Bmatrix} 1 \\ 4 \\ 0 \end{Bmatrix}$$

Se puede demostrar que la solución exacta es $\begin{Bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{Bmatrix} = \begin{Bmatrix} e^t \\ 2e^{2t} \\ \cos(t) \end{Bmatrix}$

Es posible escribirlo en forma de generalizada como

$$\mathbf{M}\,\ddot{\mathbf{u}}(t) + \mathbf{C}\,\dot{\mathbf{u}}(t) + \mathbf{K}\,\mathbf{u}(t) = \mathbf{R}(t),$$

con valores iniciales conocidos $\mathbf{u}(0), \dot{\mathbf{u}}(0)$, y con $\ddot{\mathbf{u}}(0)$ que se puede obtener de satisfacer la ecuación diferencial vectorial en el instante inicial t=0.

Para los sistemas que se escriben de esta forma se puede obtener una solución aproximada mediante el Método de Diferencia Central de la siguiente manera:

- ✓ Se aproximan la $\ddot{\mathbf{u}}(t)$ y la $\dot{\mathbf{u}}(t)$ con derivadas numéricas centrales y se reemplazan en el sistema original.

$$\mathbf{M}\frac{1}{\Delta t^2}\left(\mathbf{u}(t-\Delta t) - 2\mathbf{u}(t) + \mathbf{u}(t+\Delta t)\right) + \mathbf{C}\frac{1}{2\Delta t}\left(-\mathbf{u}(t-\Delta t) + \mathbf{u}(t+\Delta t)\right) + \mathbf{K}\mathbf{u}(t) = \mathbf{R}(t)$$

- ✓ Agrupando los términos semejantes, es posible aproximar $\mathbf{u}(t+\Delta t)$, para cualquier t ≥ t_0 planteado en la discretización, resolviendo el sistema

$$\left(\mathbf{M} + \frac{\Delta t}{2}\mathbf{C}\right)\mathbf{u}(t+\Delta t) = \Delta t^2 \mathbf{R}(t) - \left(\Delta t^2 \cdot \mathbf{K} - 2\cdot\mathbf{M}\right)\mathbf{u}(t) - \left(\mathbf{M} - \frac{\Delta t}{2}\mathbf{C}\right)\mathbf{u}(t-\Delta t)$$

o bien en forma explícita

$$\mathbf{u}(t+\Delta t) = \quad \mathbf{b}_g(t) \ + \mathbf{D}_E(\Delta t)\cdot\mathbf{u}(t) \ + \mathbf{H}_E(\Delta t)\mathbf{u}(t-\Delta t)$$

con

$$\mathbf{G}_E = \left(\mathbf{M} + \frac{\Delta t}{2}\mathbf{C}\right)^{-1}; \quad \mathbf{D}_E = \mathbf{G}_E\cdot\left(2\mathbf{M} - \Delta t^2\mathbf{K}\right); \quad \mathbf{H}_E = \mathbf{G}_E\cdot\left(\frac{\Delta t}{2}\mathbf{C} - \mathbf{M}\right); \quad \mathbf{b}_g(t) = \Delta t^2\cdot\mathbf{G}_E\cdot\mathbf{R}(t)$$

Se debe destacar que las matrices $\mathbf{G_E}$ y $\mathbf{H_E}$, como así también la matriz inversa que las define, se calcula sólo una vez al inicio del método.

- ✓ Para el valor inicial de t (t=t0), primero se halla una aproximación de $\mathbf{u}(t_0 - \Delta t)$ mediante Serie de Taylor:

$$\mathbf{u}(t_0 - \Delta t) = \left(\mathbf{u}(t_0) - \Delta t\,\dot{\mathbf{u}}(t_0) + \frac{1}{2}\Delta t^2\,\ddot{\mathbf{u}}(t_0)\right).$$

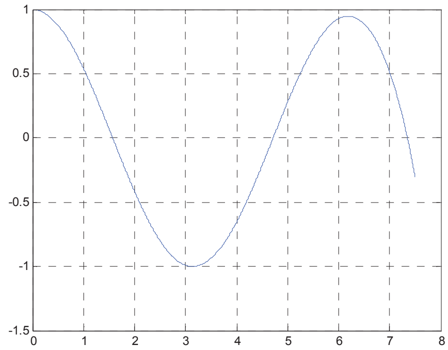
Desarrollar un programa en MATLAB que encuentre la solución del problema mediante el Método de Diferencia Central.

Una posible opción es el siguiente código

```
function Dif_cen
clc,clear
% Datos
Dt=0.003;  % incremento de tiempo
NDt=2500;  % cantidad de Dt a realizar
dim=3;     % cantidad de funciones incógnitas
M= [1  0  0
    0  -1 0
    0  0  2];
C= [4  0  0
    0  -1 0
    0  0  3];
K= [0  4  1
    4  2  0
    1  0  1];
% Valores Iniciales o actuales
tac=0;
yac= [1;  2;   1]
vac= [1;  4;   0]
fua(1,1)=5*exp(tac)+8*exp(2*tac)+cos(tac);
fua(2,1)=-8*exp(2*tac)+4*exp(tac);
fua(3,1)=-cos(tac)-3*sin(tac)+exp(tac);
% Inicialización
G=inv(M+(Dt/2)*C)
D=G*(2*M-Dt^2*K)
H=G*((Dt/2)*C-M)
acel= inv(M)*(fua-K*yac-C*vac);  % cálculo de aceleración
yan=yac-Dt*vac+((Dt^2)/2)*acel;  % cálculo de sol anterior
%
t(1)=tac;            % Almacenamiento para luego graficar
y(:,1)=yac;
% Diferencia Central
for j=2:NDt
  bac=fun_ind(tac,G,Dt); % actualización de término independiente

  ynu=bac + D*yac + H*yan;   % Calculo con Dif Central
  tnu=tac+Dt;
  t(j)=tnu;             % Almacenamiento para luego graficar
  y(:,j)=ynu;

  yan=yac;              % actualización de estado anterior
  yac=ynu;              % actualización de estado actual
  tac=tnu;
end
figure(1)
plot(t,y(3,:),'b');
grid on
end

function [fy]=fun_ind(x,G,Dt)
  r(1,1)= 5*exp(x)  +8*exp(2*x) +cos(x);
  r(2,1)=-8*exp(2*x)+4*exp(x);
  r(3,1)=-cos(x)    -3*sin(x) +exp(x);
  fy=Dt^2*G*r;
end
```

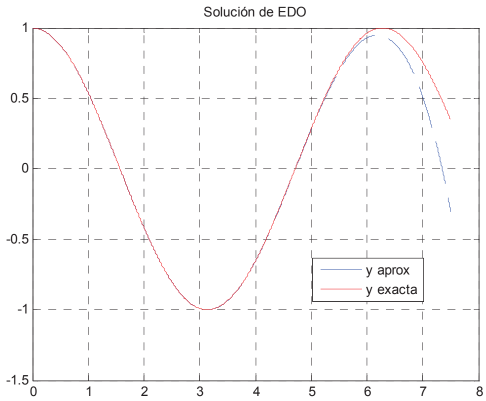
- Identificar las líneas donde se definen los datos.
- Identificar las líneas donde se definen los valores iniciales.

- Identificar el incremento de tiempo propuesto y calcular cual es el valor final de tiempo hasta el que se calcula la solución aproximada.
- Identificar las líneas donde se aplica el método diferencia central.
- Identificar donde se usa la *function* `[fy]=fun_ind(x,G,Dt)` e interpretar su utilidad.
- Si se cambian las condiciones iniciales de $\mathbf{u}(0),\ \dot{\mathbf{u}}(0),$ a otros valores (por ejemplo nulos): ¿qué líneas se deben incorporar estos nuevos datos?.
- Si se cambia el término independiente del sistema de ecuaciones diferenciales, ¿ qué líneas se deben actualizar a los nuevos datos?.
- Comprobar que la solución aproximada de la tercera componente del vector solución es la grafica adjunta: Indique las líneas que generan dicha gráfica .



Comprobar que incluyendo las siguientes líneas se origina la gráfica adjunta para la tercera componente del vector solución.

```
for j=1:NDt
    y3ex(j)=cos(t(j));
end

figure (4)
plot (t,y(3,:),'--b',t,y3ex,'r'
title ('Solución de EDO')
legend('y aprox', 'y exacta')
grid on
figure (5)
plot (t,y3ex,'r' )
title ('Solución de EDO')
grid on
```



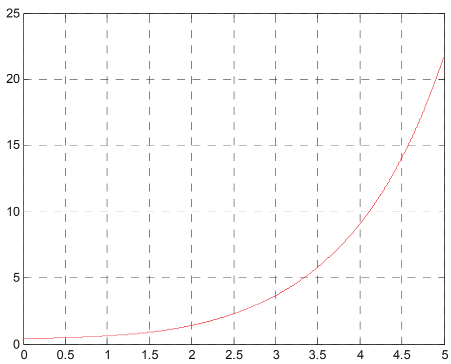
)

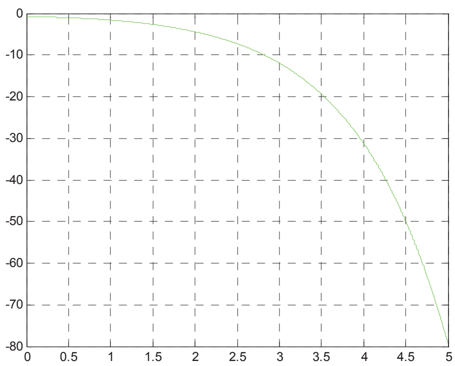
Ejercicio 19) Dado el siguiente sistema de ecuaciones diferenciales de segundo orden

$$
\begin{aligned}
16\frac{d^2\theta_1(t)}{dt^2}+4(\theta_2+\theta_1)&=5t \\
32\frac{d^2\theta_2(t)}{dt^2}+9\theta_1&=-3t^2
\end{aligned}
\quad \text{con} \begin{Bmatrix}\theta_1\\ \theta_2\end{Bmatrix}_{t0}=\begin{Bmatrix}\pi/8\\ -\pi/4\end{Bmatrix}\ y\ \begin{Bmatrix}\dfrac{d\theta_1}{dt}\\ \dfrac{d\theta_2}{dt}\end{Bmatrix}_{t0}=\begin{Bmatrix}0{,}1\\ -0{,}2\end{Bmatrix}
$$

Modificar el programa anterior y comprobar que una solución aproximada para Dt=0.001 es

para $\theta_1(t)$ para $\theta_2(t)$





Cambiar Dt a 0.01 y comprobar que las soluciones son muy próximas.

Ejercicios Teóricos

Ejercicio 20) El método de Euler hacia adelante halla una solución aproximada mediante

$$x_{n+1} = x_n + h,$$
$$y_{n+1} = y_n + h \cdot f(x_n, y_n) = y_n + h \cdot f_n,$$
$$\forall n \geq 0,$$

siendo *n=0* el valor inicial conocido. El número *n* es un entero positivo.

a) Interprete gráficamente el método de Euler hacia adelante.
b) Demuestre que el error de truncamiento local es del orden de h^2, usando serie de Taylor.
c) Demuestre que el error de truncamiento local es del orden de h^2, usando derivación numérica.

Ejercicio 21) Dada la ecuación diferencial

$$\frac{dy}{dx} + A \cdot y = 0, \qquad A \in \mathrm{R},$$
$$y(x_0) = y_0,$$

a) identifique la función *f(x,y)*.
b) Comprobar que el método de Euler hacia delante para esta EDO calcula la solución aproximada en $x_{m+1} = x_m + h$, de las siguientes formas equivalentes:

$$y_{m+1} = (1 - h \cdot A) \cdot y_m$$
$$y_{m+1} = (1 - h \cdot A)^{m+1} \cdot y_0$$

Demuestre que el paso *h* debe ser tal que |1-*h***A*|<1 para que el método de Euler dé una solución siempre decreciente. Explicar que ocurre para *h* que pertenecen al intervalo (*0; 2/A*) y para los *h* que son mayores a *2/A*.

c) Con un esquema de <u>derivada numérica hacia atrás</u> demostrar el error de truncamiento local del **Método de Euler hacia atrás**y comprobar que la <u>fórmula “implícita”</u> para calcular un nuevo valor aproximado está dado por:

$$(1 + h \cdot A) \cdot y_{m+1} = y_m$$
$$x_{m+1} = x_m + h$$

d) Comprobar que es posible calcular cualquier valor aproximado en la forma

$$(1 + h \cdot A)^{m+1} \cdot y_{m+1} = y_0$$
$$x_{m+1} = x_m + h$$

Explicar que ocurre para *h* que pertenecen al intervalo (*0; 2/A*) y para los *h* que son mayores a *2/A*.

e) Conociendo la solución exacta de la ecuación diferencial, y comparando la solución numérica con la exacta para algún valor de x, encontrar las constantes C y p del error absoluto expresado en la forma:

$$\varepsilon(x) = \left| y_{ex}(x) - y_{aprox}(x) \right| = C \cdot \Delta x^p$$

Comparar con la expresión del error de truncamiento local.

Ejercicio 22) Describa, interprete geométricamente y deduzca al Método de Euler Mejorado y su error
Ejercicio 23) Describa, interprete geométricamente y deduzca al Método de Euler Modificado y su error.
Ejercicio 24) Los métodos de *de Runge Kutta de segundo orden* se pueden plantear genéricamente mediante

Conocidos (x_m, y_m) *y asumidos* $(h,\ \omega)$ *se calcula*

$$k_1 = h\, f(x_m, y_m)$$

$$x_G = x_m + \frac{h}{2\omega} \qquad y_G = y_m + \frac{1}{2\omega} k_1$$

$$k_2 = h\, f(x_G, y_G)$$

$$y_{m+1} = y_m + (1-\omega)\, k_1 + \omega\, k_2$$

$$x_{m+1} = x_m + h$$

a) Demuestre que los *métodos de Runge Kutta de segundo orden* tienen error de truncamiento local del orden de h^3. (Consultar los apuntes de la cátedra: *Generalización de los Métodos de Runge Kutta de segundo orden*).

b) Obtenga los métodos de Euler Modificado y Mejorado como casos particulares del *método de Runge Kutta de segundo orden*.

Ejercicio 25) Para el siguiente sistema de ecuaciones diferenciales

$$\frac{d\mathbf{y}}{dt} = -\mathbf{A}\cdot\mathbf{y}(t) \text{, con: } \mathbf{y}(0) = \begin{Bmatrix} y_{10} \\ y_{20} \end{Bmatrix} \text{ siendo } \mathbf{y}(t) = \begin{Bmatrix} y_1(t) \\ y_2(t) \end{Bmatrix} \quad y \quad \mathbf{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

a) Con un esquema de derivada numérica hacia adelante comprobar que la fórmula "explicita" para **Método de Euler** adelante está dado por:

$$\mathbf{y}(t_{m+1}) = \mathbf{G}_E \cdot \mathbf{y}(t_m) \qquad con\ \ \mathbf{G}_E = (\mathbf{I} - \Delta t\cdot\mathbf{A}) = \left(\begin{bmatrix} 1-\Delta t\cdot a & \Delta t\cdot b \\ \Delta t\cdot c & 1-\Delta t\cdot d \end{bmatrix}\right)$$

$$t_{m+1} = t_m + \Delta t$$

Cada nueva solución se obtiene con el producto de una matriz por el vector solución anterior. Analizar el radio espectral (mayor autovalor en valor absoluto) de la Matriz G_E

b) Con un esquema de derivada numérica hacia atrás comprobar que la fórmula "implícita" para **Método de Euler** atrás está dado por:

$$\mathbf{G}_I \cdot \mathbf{y}(t_{m+1}) = \mathbf{y}(t_m) \qquad con\ \ \mathbf{G}_I = (\mathbf{I} + \Delta t\cdot\mathbf{A}) = \left(\begin{bmatrix} 1+\Delta t\cdot a & \Delta t\cdot b \\ \Delta t\cdot c & 1+\Delta t\cdot d \end{bmatrix}\right)$$

$$t_{m+1} = t_m + \Delta t$$

Cada nueva solución se obtiene resolviendo un sistema de ecuaciones lineales en el que el término independiente es el vector solución anterior. Analizar el radio espectral (mayor autovalor en valor absoluto) de la Matriz $G_{I.}$

c) Comprobar que para resolver con un método de Runge Kutta de segundo orden se puede operar de la siguiente forma:

Dados los valores iniciales $[\,t_m \quad \mathbf{y}(t_m)\,]$

Se calcula k1 $\mathbf{k}_1 = \Delta t\cdot\mathbf{A}\cdot\mathbf{y}(t_m)$

Se calcula el estado auxiliar con *w* distinto de cero $t_G = t_m + \Delta t/(2w)$, $\mathbf{y}(t_G) = \mathbf{y}(t_m) + \mathbf{k}_1/(2w)$

Se calcula k1 $\mathbf{k}_2 = \Delta t\cdot\mathbf{A}\cdot\mathbf{y}(t_G)$

Se calcula la nueva solución $t_{m+1} = t_m + \Delta t$, $\mathbf{y}(t_{m+1}) = \mathbf{y}(t_m) + (1-w)\cdot\mathbf{k}_1 + w\cdot\mathbf{k}_2$

Se debe notar que tanto $\mathbf{y}(t_m)$, $\mathbf{k}_1$ como $\mathbf{k}_2$ son vectores de dos componentes en este caso. Y que tanto *w*, t_m como Δt son escalares.

Ejercicios Integradores

Ejercicio 26) Dada la siguiente ecuación diferencial a derivadas parciales,

$$\frac{\partial u(x,t)}{\partial t} = D\frac{\partial^2 u(x,t)}{\partial x^2} \quad si\; x\varepsilon\Omega = \{x\varepsilon R : 0 \le x \le L\}$$
$$con\;\; u(0,t) = u(L,t) = 0 \quad \forall t \qquad u(x,0) = \sin(\pi \cdot x / L) \quad \forall x$$

considerando una discretización en la variable x de 6 puntos equidistantes en todo el dominio; manteniendo la continuidad en la variable t, comprobar que se obtiene un sistema que se puede escribir como:

$$\dot{\vec{\mathbf{u}}}(t) + \mathbf{K}\,\vec{\mathbf{u}}(t) = 0$$

Con el vector **u** que agrupa las funciones incógnitas de la variable tiempo. Usar el método de Euler Mejorado para resolver este sistema de ecuaciones diferenciales ordinarias de segundo orden. Implementar un programa en MATLAB para resolver este ejercicio

Ejercicio 27) Dada la siguiente ecuación diferencial a derivadas parciales,

$$\frac{\partial^2 u(x,t)}{\partial t^2} = c^2\frac{\partial^2 u(x,t)}{\partial x^2} \quad si\; x\varepsilon\Omega = \{x\varepsilon R : 0 \le x \le L\}$$
$$con\;\; u(0,t) = u(L,t) = 0 \quad \forall t \qquad u(x,0) = \sin(\pi \cdot x / L) \quad \forall x \quad en\; t = 0$$
$$\left.\frac{\partial u(x,t)}{\partial t}\right|_{t=0} = 0 \qquad \forall x \quad en\; t = 0$$

considerando una discretización en la variable *x* de 5 puntos equidistantes en todo el dominio; manteniendo la continuidad en la variable t, comprobar que se obtiene un sistema que se puede escribir como:

$$\mathbf{M}\,\ddot{\vec{\mathbf{u}}}(t) + \mathbf{K}\,\vec{\mathbf{u}}(t) = 0$$

Con el vector **u** que agrupa las funciones incógnitas de la variable tiempo. Usar método de diferencia central en el tiempo o el método de Euler Modificado para resolver este sistema de ecuaciones diferenciales ordinarias de segundo orden. Implementar un programa en MATLAB para resolver este ejercicio

Ejercicio 28) Dada la siguiente ecuación diferencial a derivadas parciales,

$$-k\frac{\partial^2 u(x,t)}{\partial x^2} + m(x)\frac{\partial^2 u(x,t)}{\partial t^2} = p \cdot \sin(3t) \qquad si\; x\varepsilon\Omega = \{x\varepsilon R : 0 \le x \le L\}$$
$$u(0,t) = 0 \qquad \forall t \qquad\qquad u(x,0) = x \quad \forall x$$
$$k(L)\frac{\partial u}{\partial x}(L) = -M_L\frac{\partial^2 u(L,t)}{\partial t^2} \quad \forall t \qquad\qquad \frac{\partial u}{\partial t}(x,0) = 0 \quad \forall x$$

considerando una discretización en la variable *x* de 5 puntos equidistantes en todo el dominio; manteniendo la continuidad en la variable t, comprobar que usando derivada numérica se obtiene un sistema que se puede escribir como:

$$\mathbf{M}\,\ddot{\vec{\mathbf{u}}}(t) + \mathbf{K}\,\vec{\mathbf{u}}(t) = \vec{b}(t)$$

Con el vector **u** que agrupa las funciones incógnitas de la variable tiempo. Usar método de diferencia central en el tiempo o el método de Euler para resolver este sistema de ecuaciones diferenciales ordinarias de segundo orden. Implementar un programa en MATLAB para resolver este ejercicio

Ejercicio 29) El movimiento *u(x,t)* de una cuerda perfectamente flexible, de longitud *L*, y sometida a una tracción *T* y una acción *p(x,t)* por unidad de longitud, es la solución de la siguiente ecuación diferencial:

$$T\frac{\partial^2 u(x,t)}{\partial x^2} + p(x)\cdot g(t) = \rho(x)\frac{\partial^2 u(x,t)}{\partial t^2} \qquad si\; x\varepsilon\Omega = \{x\varepsilon R : 0 \le x \le L\}$$

$$con\;\; u(0,t) = u(L,t) = 0 \quad \forall t$$

$$con\;\; u(x,0) = \frac{\partial u}{\partial t}(x,0) = 0 \;\; en \quad t = 0$$

Usando un esquema de derivación numérica en la variable *x*, demostrar que es posible obtener el siguiente sistema discreto aproximado

$$M\cdot \ddot{u}(t) + K\cdot u(t) = p(t)$$

Siendo $u(t)^T = \{ u_1(t), u_2(t), u_3(t), u_4(t), u_5(t) \}$

$$M = \begin{pmatrix} \rho(x_1) & 0 & 0 & 0 & 0 \\ 0 & \rho(x_2) & 0 & 0 & 0 \\ 0 & 0 & \rho(x_3) & 0 & 0 \\ 0 & 0 & 0 & \rho(x_4) & 0 \\ 0 & 0 & 0 & 0 & \rho(x_5) \end{pmatrix} \quad K = (T/h^2)\begin{pmatrix} 2 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 2 \end{pmatrix} \quad p(t) = \begin{Bmatrix} p(x_1) \\ p(x_2) \\ p(x_3) \\ p(x_4) \\ p(x_5) \end{Bmatrix} g(t)$$

Usar el método de diferencia central en el tiempo para resolver este sistema de ecuaciones diferenciales ordinarias de segundo orden. Implementar un programa en MATLAB para resolver este ejercicio.

Ejercicios con Métodos Predictor Corrector y Multipasos

Ejercicio 30) Dada la ecuación diferencial el esquema de solución de ecuaciones diferenciales ordinarias llamado Método de Heun, consiste en un esquema de tipo predictor corrector.
En la fase predictora se obtiene una aproximación inicial con el método de Euler hacia adelante.

La fase correctora consiste en aplicar en forma iterativa el llamado método implícito de Trapecios para encontrar una sucesión de soluciones aproximadas hasta tanto éstas sean tan próximas como se desee.

a) Demuestre que el método de trapecios tiene un error de truncamiento local del orden de h^3. Utilice un esquema de integración numérica, para integrar entre x_n y x_{n+1} la ecuación $\frac{dy}{dx} = f(x, y)$.
b) Explique por qué se trata de un método implícito.
c) Explique si es un método multipaso o de un solo paso.
d) Aplique el método para resolver la siguiente ecuación diferencial:

$$\frac{dy}{dx} = -x \cdot y^2$$
$$y(0) = 2$$

Resuelva usando un paso h=0,25, para encontrar la solución en x=0,25; x=0,5; y x=0,75. Aplique 3(tres) iteraciones de la fase correctora en cada caso.

Ejercicio 31) Demuestre que la siguiente aproximación para obtener la solución de una ecuación diferencial ordinaria de primer orden

$$y_{m+1} = y_{m-1} + \frac{h}{3} \cdot (f_{m+1} + 4f_m + f_{m-1})$$

tiene error de truncamiento local del orden de h^5. Utilice el esquema de integración numérica de Simpson para integrar $\frac{dy}{dx} = f(x)$ entre x_{m-1} y x_{m+1}.

- Indique si es un método implícito o explícito.
- Indique si sería posible incluirlo como fase correctora en un esquema predictor corrector.
-

Ejercicio 32) Dada la ecuación diferencial el método del Punto Medio permite calcular la aproximación de la solución de una ecuación diferencial de primer orden mediante

$$y_{m+1} = y_{m-1} + 2h \cdot f(x_m; y_m) = y_{m-1} + 2h \cdot f_m \, .$$

a) Interprete gráficamente este método.
b) Utilizando un esquema de derivación numérica central, demuestre que el error de truncamiento local es del orden de h^3.
c) Explique si se trata de un método implícito o explícito; si es de un paso o multipaso.
d) Aplique el método del punto medio para resolver la ecuación diferencial:

$y' = x^2 - y$
$y(0) = 0{,}5$.