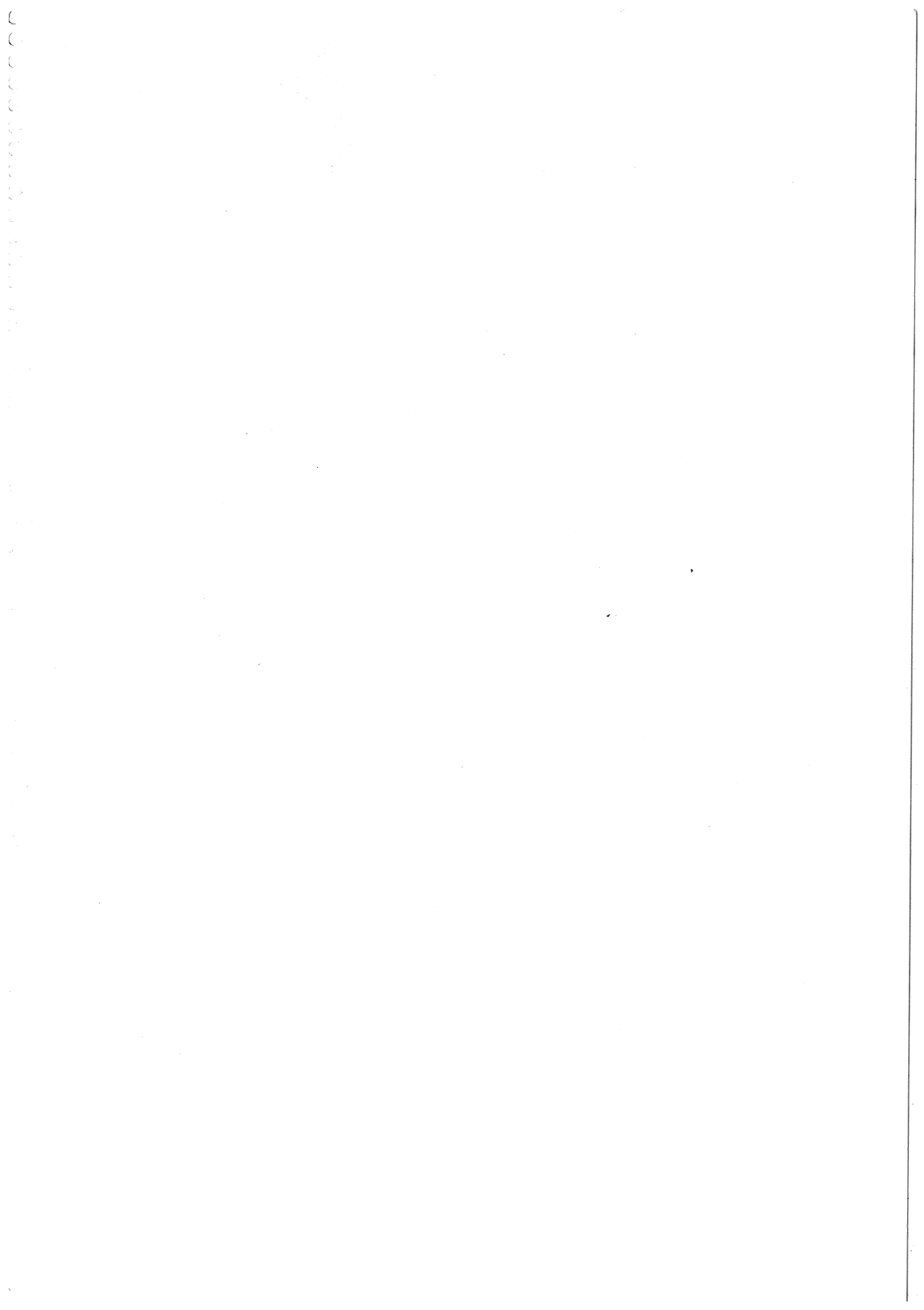


Analisis

Numérico II

$$\lambda = \alpha^2 \frac{k}{h^2}$$

Antonio Fernando Villarán Carrellán
José María Jiménez León
Víctor Manuel García Villarán



ÍNDICE

PROBLEMAS DE CONTORNO

MÉTODOS LINEALES

1.- MÉTODO DEL DISPARO

-CONCEPTOS TEÓRICOS1
-EJEMPLOS RESUELTOS2
-CÓDIGO FUENTE7

2.- MÉTODO DE LAS DIFERENCIAS FINITAS

-CONCEPTOS TEÓRICOS9
-EJEMPLOS RESUELTOS11
-CÓDIGO FUENTE16

MÉTODOS NO LINEALES

3.- MÉTODO DEL DISPARO

-CONCEPTOS TEÓRICOS18
-EJEMPLOS RESUELTOS20
-CÓDIGO FUENTE24

4.- MÉTODO DE LAS DIFERENCIAS FINITAS

-CONCEPTOS TEÓRICOS27
-EJEMPLOS RESUELTOS29
-CÓDIGO FUENTE33

5.- MÉTODO DE RAYLEIGH-RITZ

-CONCEPTOS TEÓRICOS36
-FUNCIONES SEGMENTARIAS LINEALES	37
-EJEMPLOS RESUELTOS39
-CÓDIGO FUENTE CON LINEALES.....	45
-FUNCIONES SEGMENTARIAS CÚBICAS.....	47
-EJEMPLOS RESUELTOS49
-CÓDIGO FUENTE CON CÚBICAS.....	55

ECUACIONES EN DERIVADAS PARCIALES

6.- ECUACIONES EN DERIVADAS PARCIALES ELÍPTICAS

-CONCEPTOS TEÓRICOS	59
-EJEMPLOS RESUELTOS	61
-CÓDIGO FUENTE	68

7.- ECUACIONES EN DERIVADAS PARCIALES HIPERBÓLICAS

-CONCEPTOS TEÓRICOS	72
-EJEMPLOS RESUELTOS	75
-CÓDIGO FUENTE	81

8.- ECUACIONES EN DERIVADAS PARCIALES PARABÓLICAS

-CONCEPTOS TEÓRICOS	83
-EJEMPLOS RESUELTOS	88
-CÓDIGO FUENTE	94

PROBLEMAS DE CONTORNO

1.- MÉTODO DEL DISPARO PARA PROBLEMAS LINEALES.

1.1. Conceptos Teóricos:

Supongamos que la función f en el problema de valor de frontera

$$y'' = f(x, y, y') \quad a \leq x \leq b, \quad y(a) = \alpha, y(b) = \beta,$$

es continua en el conjunto

$$D = \{ (x, y, y') \mid a \leq x \leq b, -\infty < y < \infty, -\infty < y' < \infty \},$$

y que $\partial f / \partial y$ y $\partial f / \partial y'$ son también continuas en D . Si

- i) $\partial f / \partial y (x, y, y') > 0$ para toda $(x, y, y') \in D$, y
- ii) existe una constante M , con

$$\left| \partial f / \partial y (x, y, y') \right| \leq M \text{ para toda } (x, y, y') \in D,$$

entonces, el problema de valor de frontera tiene una solución única.

Cuando $f(x, y, y')$ puede expresarse en la forma: $f(x, y, y') = p(x)y' + q(x)y + r(x)$, se llama lineal. Los problemas de este tipo se presentan muy frecuentemente en la práctica y esta representación permite simplificar considerablemente lo visto anteriormente.

Si el problema de valor de la frontera $y'' = p(x)y' + q(x)y + r(x)$, $a \leq x \leq b$, $y(a) = \alpha$ y $y(b) = \beta$, se satisface que:

- i) $p(x)$, $q(x)$, y $r(x)$ son continuas en $[a, b]$.
- ii) $q(x) > 0$ en $[a, b]$.

Para aproximar la solución única, garantizada si se satisfacen las hipótesis anteriores, consideremos primero los problemas de valor inicial

$y'' = p(x)y' + q(x)y + r(x)$, $a \leq x \leq b$, $y(a) = \alpha$ y $y'(a) = 0$, la solución se guarda en $y_1(x)$ y

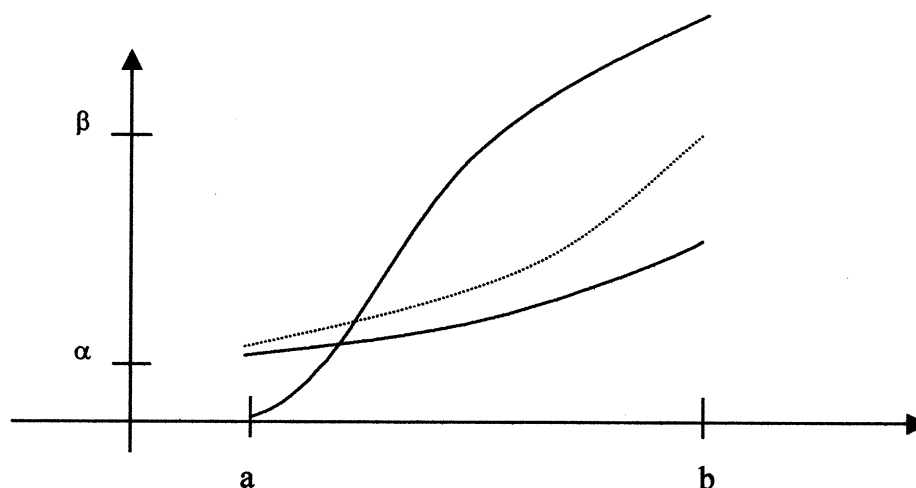
$y'' = p(x)y' + q(x)y$, $a \leq x \leq b$, $y(a) = 0$ y $y'(a) = 1$, la solución se guarda en $y_2(x)$.

y la solución final es: $y(x) = y_1(x) + (\beta - y_1(b) / y_2(b))y_2(x)$, (1)

Donde $y(x)$ es la solución única a nuestro problema de valor de la frontera, siempre y cuando, desde luego, $y_2(b) \neq 0$.

Este algoritmo usa la técnica de Runge-Kutta de cuarto orden para encontrar las aproximaciones de $y_1(x)$ e $y_2(x)$ y, una vez que estas aproximaciones están a la mano,

la solución del problema de valor de la frontera puede aproximarse usando la ecuación anterior (1). Gráficamente, el método tiene la apariencia mostrada en la siguiente figura:



Este algoritmo tiene la característica adicional de obtener aproximaciones para la derivada de la solución al problema de valor de la frontera, además de la solución del propio problema.

Resumiendo lo anterior, para solucionar dicho problema tenemos que realizar los siguientes pasos:

1. Resolver el primer problema de valor inicial con la función y los valores iniciales siguientes:

$$y'' = p(x)y' + q(x)y + r(x), \quad a \leq x \leq b, \quad y(a) = \alpha \quad y'(a) = 0$$

2. Resolver el segundo problema de valor inicial con la función y los valores iniciales siguientes:

$$y'' = p(x)y' + q(x)y \quad a \leq x \leq b, \quad y(a) = 0 \quad y'(a) = 1$$

3. Calcular la solución de la siguiente forma:

$y(x) = y_1(x) + (\beta - y_1(b) / y_2(b))y_2(x)$, (1), donde $y_1(x)$ es la solución al paso 1 e $y_2(x)$ es la solución al paso 2, y $(\beta - y_1(b) / y_2(b))$ es una constante.

1.2-Ejemplos de Prueba.

1.2.1.-Veamos el siguiente ejemplo:

$$y' = 3xy' + x^2y + 2 \cos(x)e^x;$$

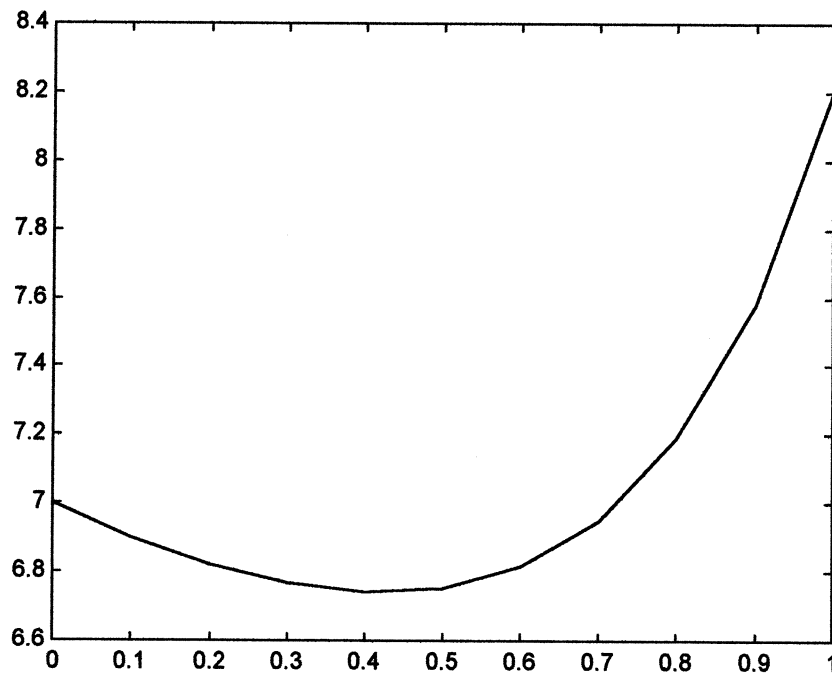
$$y(0) = 7;$$

$$y(1) = 8,2;$$

Si utilizamos $n=10$ divisiones del intervalo, se obtiene el siguiente resultado:

x_i	w_i
0	7.00000000000000
0.100000000000000	6.90188414001835
0.200000000000000	6.82401772855054
0.300000000000000	6.76903218600509
0.400000000000000	6.74244424374627
0.500000000000000	6.75335503680794
0.600000000000000	6.81572807434012
0.700000000000000	6.95055795713650
0.800000000000000	7.18948711503722
0.900000000000000	7.58086096440024
1.000000000000000	8.20000000000000

Gráficamente, el valor de la función aproximado por los valores w_i obtenidos con el método del disparo para problemas de contorno lineales, entre los valores 0 y 1 es:



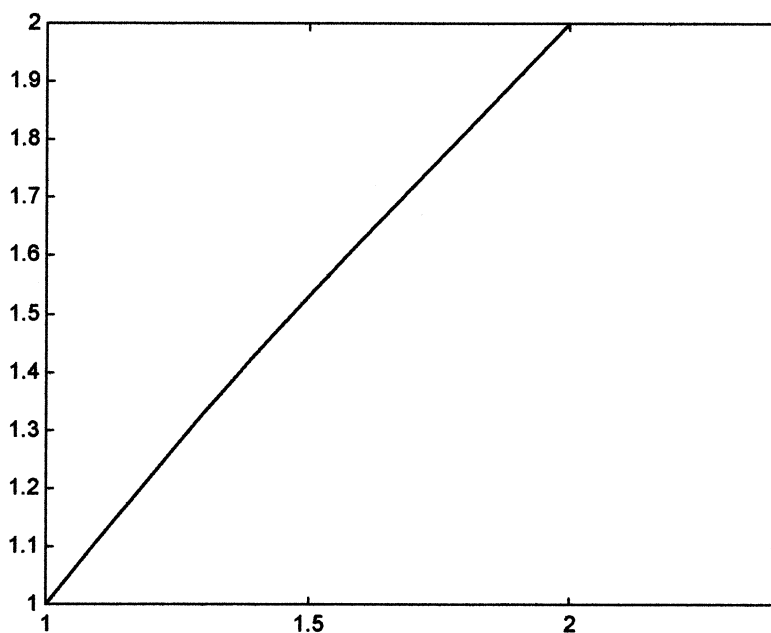
1.2.2.-Para el ejemplo:

$$y'' = (-2/t)y' + (2/t^2)y + \text{sen}(e^t)/t^2$$

sabiendo que $y(1)=1$, $y(2)=2$ y tomando $n=10$, se obtiene:

x_i	w_i
1.0000000000000000	1.0000000000000000
1.1000000000000000	1.11042106102213
1.2000000000000000	1.22030653320524
1.3000000000000000	1.32780982400108
1.4000000000000000	1.43181224407540
1.5000000000000000	1.53186325267262
1.6000000000000000	1.62818065255701
1.7000000000000000	1.72163604252841
1.8000000000000000	1.81364984960177
1.9000000000000000	1.90592625537020
2.0000000000000000	2.0000000000000000

Y la gráfica que se obtiene es:



1.2.3.-Para el siguiente problema de contorno:

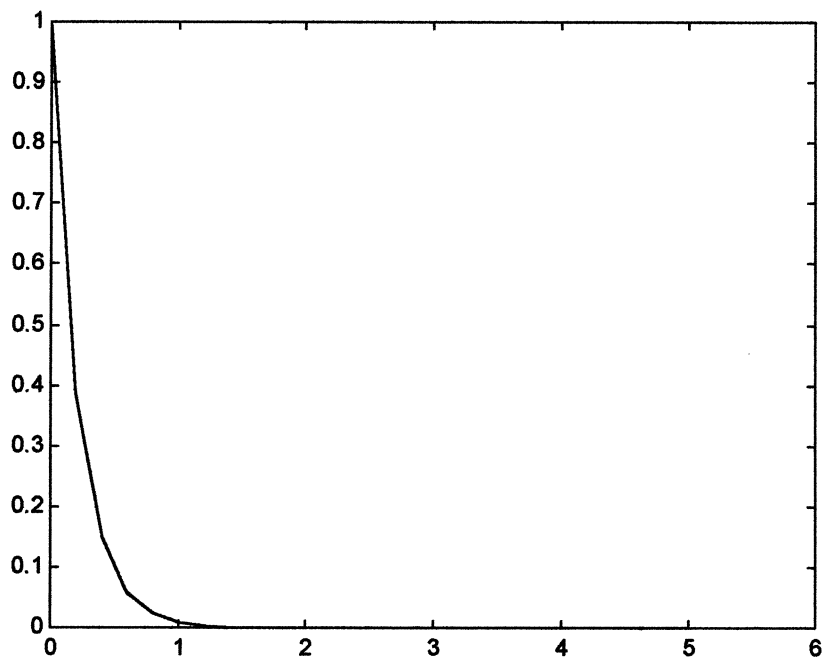
$$y'' = -4y' + 4y$$

$$y(0) = 1, y(5) = 0$$

y tomando $h=0.2$, es decir $n=25$; se obtienen las siguientes aproximaciones:

x_i	w_i
0	1.00000000000000
0.20000000000000	0.38673262196832
0.40000000000000	0.14956212089401
0.60000000000000	0.05784055115972
0.80000000000000	0.02236882800512
1.00000000000000	0.00865075550361
1.20000000000000	0.00334552935652
1.40000000000000	0.00129382533827
1.60000000000000	0.00050036446349
1.80000000000000	0.00019350725861
2.00000000000000	0.00007483556678
2.20000000000000	0.00002894135175
2.40000000000000	0.00001119256107
2.60000000000000	0.00000432852403
2.80000000000000	0.00000167397618
3.00000000000000	0.00000064737498
3.20000000000000	0.00000025035369
3.40000000000000	0.00000009681129
3.60000000000000	0.00000003742987
3.80000000000000	0.00000001446330
4.00000000000000	0.00000000557921
4.20000000000000	0.00000000214087
4.40000000000000	0.00000000080814
4.60000000000000	0.00000000028915
4.80000000000000	0.00000000008423
5.00000000000000	0

Y la gráfica es:



1.2.4.-Veamos un último ejemplo:

$$y'' = -(x+1)y' + 2y + (1-x^2)e^{-x}$$

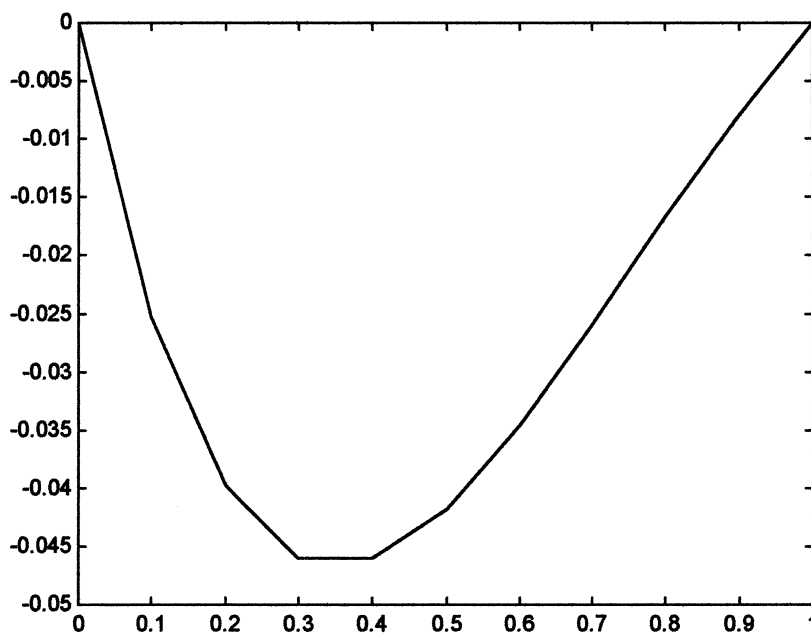
$$y(0)=y(1)=0$$

y tomemos $h=0.1$, con lo que el número de divisiones del intervalo $[0,1]$ será 10.

Se obtienen las siguientes aproximaciones:

x_i	w_i
0	1.000000000000000
0.100000000000000	-0.02518388290552
0.200000000000000	-0.03975421689201
0.300000000000000	-0.04603280438423
0.400000000000000	-0.04609888619191
0.500000000000000	-0.04177196301509
0.600000000000000	-0.03460757381340
0.700000000000000	-0.03460757381340
0.800000000000000	-0.01671390620517
0.900000000000000	-0.00786929400221
1.000000000000000	0

Y la aproximación de la gráfica de la función y entre 0 y 1 será por tanto la siguiente:



1.3.-Código fuente.

El código del algoritmo utilizado está compuesto por los siguientes ficheros:

- Fichero principal *displ.m*:

```
% Resolución de una ecuación lineal de 2° orden usando el
método del
% disparo con Runge-Kutta de orden 4.
clear;
% Se pide primero las condiciones iniciales del sistema de
2° orden:
aux1=input('Escriba los puntos donde se conoce su valor
[a,b]=');
aux2=input('Escriba los valores [y(a),y(b)]=');
t=input('Escriba el punto para calcular su valor t= ');
n=input('Escriba el número de divisiones (el valor
verdadero) n= ');

a=aux1(1);
b=aux1(2);
alfa=aux2(1);
beta=aux2(2);
to=a;          % Punto inicial to.
w1=[alfa,0]; % Valor inicial y(to).
h=(t-to)/n;
for k=1:n
    c(k)=to;    % Guardo estos valores para la
representación.
    d(k)=w1(1); % " " " " " "
    w1=r('y1',to,w1,h);
    to=to+h;
end
c(n+1)=to;
d(n+1)=w1(1);

to=a;
w2=[0,1];
h=(t-to)/n;
for k=1:n
    e(k)=w2(1); % Guardo estos valores para la
representación.
    w2=r('y2',to,w2,h);
    to=to+h;
end
e(n+1)=w2(1);

v=t; % Guardo el punto donde queremos calcular su valor.
to=a; % Cálculo de y1(b).
t=b;
```

```

y1b=[alfa,0];
h=(t-to)/n;
for k=1:n
    y1b=r('y1',to,y1b,h);
    to=to+h;
end

to=a;    % Cálculo de y2(b).
t=b;
y2b=[0,1];
h=(t-to)/n;
for k=1:n
    y2b=r('y2',to,y2b,h);
    to=to+h;
end

'La aproximación a la solución en',v,'es:'
,d(n+1)+(((beta-y1b(1))/y2b(1))*e(n+1))

for k=1:n+1
    w(k)=d(k)+(((beta-y1b(1))/y2b(1))*e(k));
    M(k,:)=[c(k) w(k)];
end
M
'Pulse una tecla para ver el gráfico...'
pause
plot(c,w);

```

- Fichero *R.m*:

```

function yn=r(s,t,w,h)
k1=h*feval(s,t,w);
k2=h*feval(s,t+h/2,w+k1/2);
k3=h*feval(s,t+h/2,w+k2/2);
k4=h*feval(s,t+h,w+k3);
yn=w+1/6*(k1+2*k2+2*k3+k4);

```

- Fichero *y1.m*: Contiene el primer sistema para resolver *y1*.

```

function yprima=f(t,x);
y=x(1);
z=x(2);

yprima=[z, (-2/t)*z+(2/(t^2))*y+((sin(exp(t)))/(t^2))];

```

- Fichero *y2.m*: Contiene el segundo sistema para resolver *y2*.

```

function yprima=f(t,x);
y=x(1);
z=x(2);
yprima=[z, (-2/t)*z+(2/(t^2))*y];

```

2.- MÉTODO DE LAS DIFERENCIAS FINITAS

2.1. Conceptos teóricos:

Aun cuando el método presentado anteriormente (disparo lineal) se puede usar para valores de frontera presentan problemas de inestabilidad. El método que vamos a presentar tiene mejores características de estabilidad, pero requieren generalmente de mayor trabajo para alcanzar una precisión específica.

Los métodos que involucran diferencias finitas para resolver problemas de valor de frontera consisten en reemplazar cada una de las derivadas en la ecuación diferencial por una aproximación diferencia-cociente aproximada.

El problema lineal de segundo orden de valor de la frontera

$$y'' = p(x)y' + q(x)y + r(x), \quad a \leq x \leq b, \quad y(a) = \alpha \quad y \quad y'(b) = \beta \quad (1)$$

requiere que se usen aproximaciones diferencia-cociente para aproximar y' e y'' . Para hacer esto, seleccionamos un número $N > 0$ y dividimos el intervalo $[a, b]$ en $(N+1)$ subintervalos iguales, cuyos puntos son los puntos de red $x_i = a + ih$ para $i=0, 1, 2, \dots, N+1$. Escogiendo la constante h de esta manera facilitara la aplicación de un algoritmo de matrices, el cual requerirá de la solución de un sistema lineal con una matriz de $N \times N$.

En los puntos interiores de la red $x_i = 1, 2, 3, \dots, N$, la ecuación diferencial a aproximar es

$$y''(x_i) = p(x_i)y'(x_i) + q(x_i)y(x_i) + r(x_i) \quad (2)$$

Expandiendo y en un polinomio de Taylor de tercer grado alrededor de x , y evaluando en x_i y x_{i-1} , tenemos que:

$$y(x_{i+1}) = y(x_i + h) = y(x_i) + hy'(x_i) + (h^2/2)y''(x_i) + (h^3/6)y''' + (h^4/24)y^{(4)}(\xi_i^+),$$

para alguna ξ_i^+ , $x_i < \xi_i^+ < x_{i+1}$, y

$$y(x_{i-1}) = y(x_i - h) = y(x_i) - hy'(x_i) + (h^2/2)y''(x_i) - (h^3/6)y''' + (h^4/24)y^{(4)}(\xi_i^-),$$

para alguna ξ_i^- , $x_{i-1} < \xi_i^- < x_i$, suponiendo que $y \in C^{(4)} [x_{i-1}, x_{i+1}]$. Si se suman estas ecuaciones, se eliminan los términos que contienen $y'(x_i)$ e $y'''(x_i)$, y una simple manipulación algebraica da

$$y''(x_i) = 1/h^2 [y(x_{i+1}) - 2y(x_i) + y(x_{i-1})] - (h^2/24)[y^{(4)}(\xi_i^+) + y^{(4)}(\xi_i^-)]$$

El teorema del valor medio puede usarse para simplificar esto aun más:

$$y''(x_i) = 1/h^2 [y(x_{i+1}) - 2y(x_i) + y(x_{i-1}))] - (h^2/12) y^{(4)}(\xi_i) \quad (3)$$

para algún punto ξ_i , $x_{i-1} < \xi_i < x_{i+1}$. La ecuación (3) se llama fórmula de diferencia centrada para $y''(x_i)$.

Una fórmula de diferencia centrada para $y'(x_i)$ puede obtenerse de manera similar resultando en

$$y'(x_i) = 1/h [y(x_{i+1}) - y(x_{i-1}))] - (h/6) y'''(\eta_i), \quad (4)$$

para alguna η_i donde $x_{i-1} < \eta_i < x_{i+1}$.

El uso de estas fórmulas de diferencia centrada en la ecuación (2) da lugar a la ecuación

$$\frac{y(x_{i+1}) - 2y(x_i) + y(x_{i-1}))}{h^2} = p(x_i) \left[\frac{y(x_{i+1}) - y(x_{i-1}))}{2h} \right] + q(x_i)y(x_i) + r(x_i) - \frac{h^2}{12} [2p(x_i)y'''(\eta_i) - y^{(4)}(\xi_i)]$$

Un método de diferencia finita con error de truncamiento de orden $O(h^2)$ resulta de usar esta ecuación junto con las condiciones de frontera $y(a)=\alpha$ e $y(b)=\beta$ para definir

$$w_0 = \alpha \quad w_{n+1} = \beta \quad y$$

$$\left(\frac{2w_i - w_{i+1} - w_{i-1}}{h^2} \right) + p(x_i) \left(\frac{w_{i+1} - w_{i-1}}{2h} \right) + q(x_i)w_i = -r(x_i)$$

para cada $i=1, 2, 3, 4, \dots, N$.

En la forma que consideramos, la ecuación anterior se reescribe como

$$-\left(1 + \frac{h}{2}p(x_i)\right)w_{i-1} + (2 + h^2q(x_i))w_i - \left(1 - \frac{h}{2}p(x_i)\right)w_{i+1} = -h^2r(x_i),$$

y el sistema resultante de ecuaciones se obtiene en la forma matricial tridiagonal de $N \times N$ mostrada en la siguiente figura.

$$Aw = b$$

$$A = \begin{bmatrix} 2+h^2q(x_1) & -1+\frac{h}{2}p(x_1) & 0 & \Lambda & 0 \\ -1-\frac{h}{2}p(x_2) & 2+h^2q(x_2) & -1+\frac{h}{2}p(x_2) & 0 & M \\ 0 & 0 & 0 & 0 & 0 \\ M & 0 & 0 & 0 & -1+\frac{h}{2}p(x_{n-1}) \\ 0 & \Lambda & 0 & -1-\frac{h}{2}p(x_n) & 2+h^2q(x_n) \end{bmatrix}$$

$$w = \begin{bmatrix} w_1 \\ w_2 \\ M \\ w_{n-1} \\ w_n \end{bmatrix} \quad \text{y } b = \begin{bmatrix} -h^2r(x_1) + \left(1 + \frac{h}{2}p(x_1)\right)w_0 \\ -h^2r(x_2) \\ M \\ -h^2r(x_{n-1}) \\ -h^2r(x_n) + \left(1 + \frac{h}{2}p(x_n)\right)w_{n+1} \end{bmatrix}$$

El teorema siguiente da las condiciones bajo las cuales el sistema tridiagonal anterior tiene una solución única. Dicho teorema dice lo siguiente:

Supongamos que p , q y r son continuas en $[a, b]$. Si $q(x) \geq 0$ en $[a, b]$ entonces el sistema lineal tridiagonal anterior tiene una solución única siempre que $h < 2/L$ donde $L = \max_{a \leq x \leq b} |p(x)|$

Debe notarse que las hipótesis del teorema anterior garantizan una solución única al problema de valor de la frontera, pero no garantizan que $y \in C^{(4)}[a, b]$. Es necesario establecer que y'''' es continua en $[a, b]$ para asegurar que el error de truncamiento es de orden $O(h^2)$.

2.2. Aplicación del algoritmo.

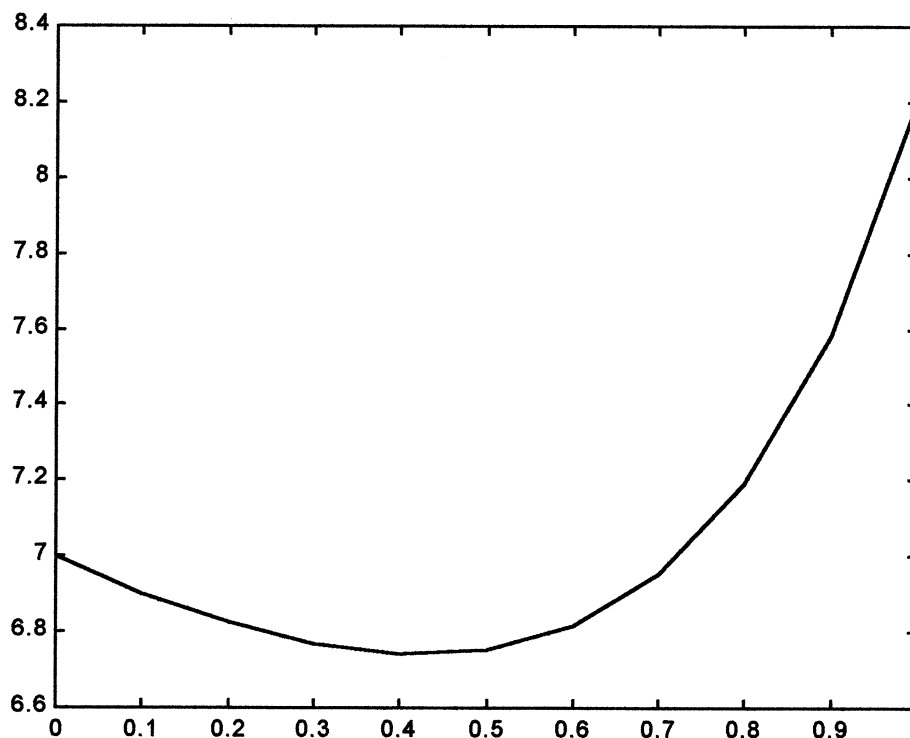
2.2.1. Tenemos la siguiente ecuación a la que le aplicaremos el algoritmo de las diferencias finitas.

$$y'' = 3ty' + t^2y + 2e^t \cos t$$

Para los valores siguientes: $y(0)=7$, $y(1)=8.2$ y tomaremos $n=9$. Aplicando el método de las diferencias finitas:

X_i	W_i
0	7.000000000000000
0.100000000000000	6.90249937227391
0.200000000000000	6.82505785565649
0.300000000000000	6.77032216643322
0.400000000000000	6.74381523679173
0.500000000000000	6.75463863012858
0.600000000000000	6.81675703568979
0.700000000000000	6.95118184690010
0.800000000000000	7.18962236974460
0.900000000000000	7.58062004552683
1.000000000000000	8.200000000000000

Para los resultados anteriores obtenemos la siguiente representación gráfica:



2.2.2. Para el siguiente ejemplo tenemos la siguiente ecuación:

$$y'' = (-2/t)y' + (2/t^2)y + \text{sen}(e^t)/t^2$$

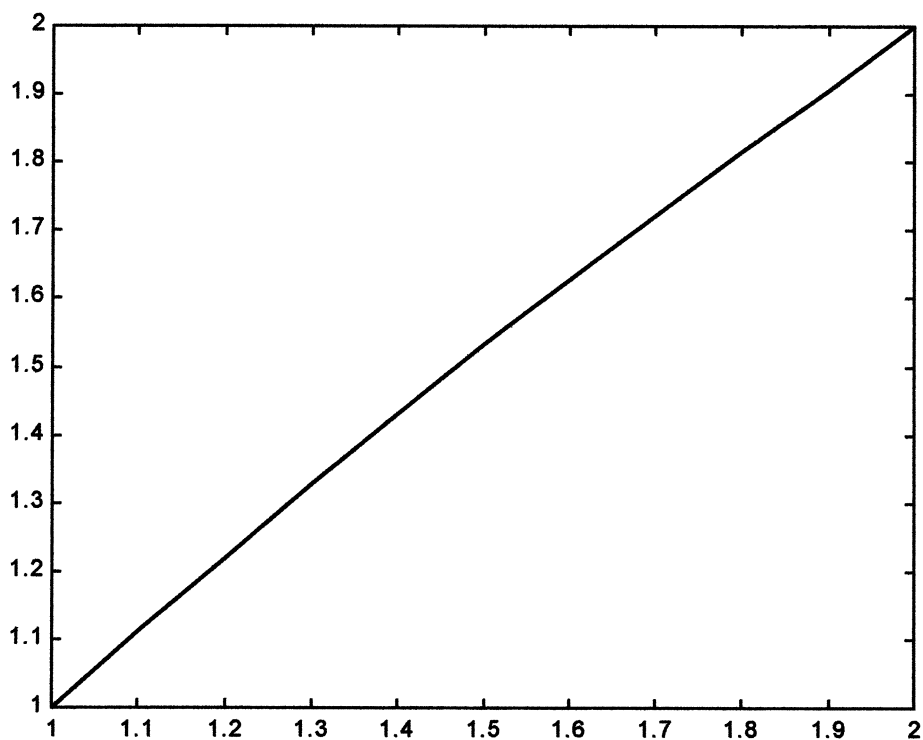
El estudio se realizará para los siguientes valores:

$$y(1)=1, y(2)=2 \text{ y } n=9.$$

Obtenemos la siguiente tabla de resultados aplicando el algoritmo:

Xi	Wi
1.000000000000000	1.000000000000000
1.100000000000000	1.11060379886904
1.200000000000000	1.22063879838612
1.300000000000000	1.32825623455605
1.400000000000000	1.43232920979823
1.500000000000000	1.53239695904773
1.600000000000000	1.62866993682988
1.700000000000000	1.72202166372665
1.800000000000000	1.81389078639129
1.900000000000000	1.90602035463667
2.000000000000000	2.000000000000000

Obtenemos la siguiente gráfica de resultados:



2.2.3. Tenemos la siguiente ecuación a resolver:

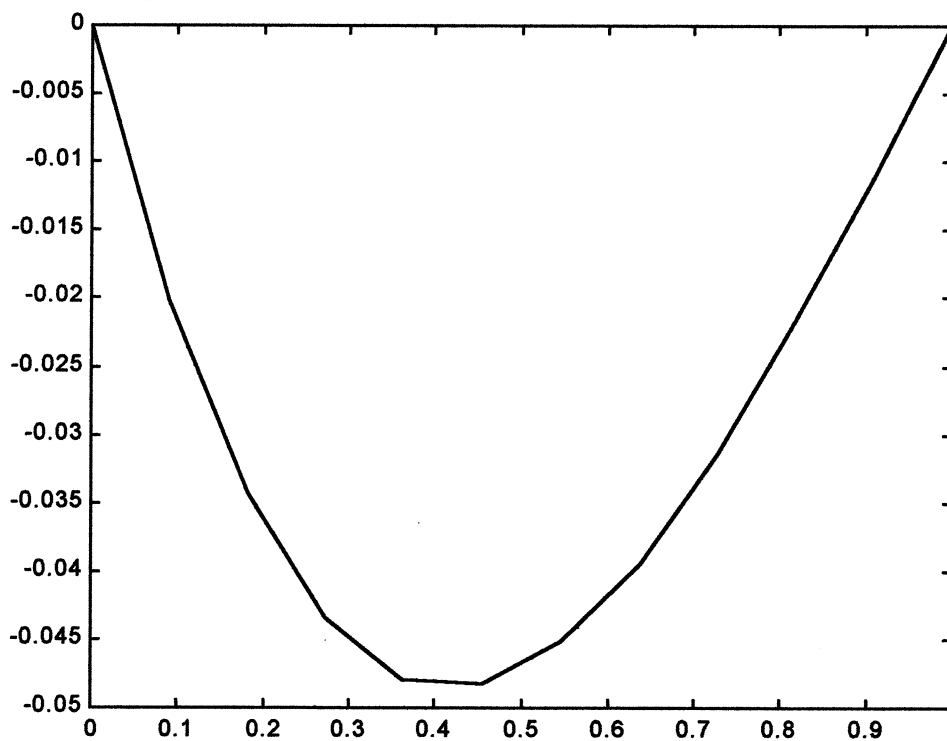
$$y'' = -(x+1)y' + 2y + (1-x^2)e^{-x}$$

para $0 \leq x \leq 1$, $y(0)=y(1)=0$, tomando $h=0.1$ con lo que utilizaremos $n=10$.

La tabla de resultados obtenidos es la que sigue:

X_i	W_i
0	0
0.100000000000000	-0.02526146034349
0.200000000000000	-0.03987702318038
0.300000000000000	-0.04617547063390
0.400000000000000	-0.04624224244002
0.500000000000000	-0.04190229313328
0.600000000000000	-0.03471588644127
0.700000000000000	-0.02598480227569
0.800000000000000	-0.01676652484221
0.900000000000000	-0.00789415214639
1.000000000000000	0

La gráfica correspondiente es la siguiente:



2.2.4. Dada la siguiente ecuación realizar el estudio pertinente aplicándole el método de los elementos finitos:

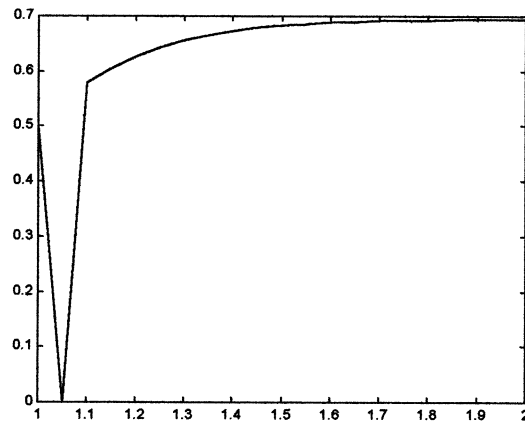
$$y'' = -\frac{4}{x}y' - \frac{2}{x^2}y + \frac{2}{x^2}\ln x$$

para $1 \leq x \leq 2$, con $y(1)=1/2$, $y(2)=\ln 2$ y $h=0.05$.

Obtenemos la siguiente tabla de resultados, que hemos obtenido:

Xi	Wi
1.0000000000000000	0.5000000000000000
1.0500000000000000	0
1.1000000000000000	0.57889540741350
1.1500000000000000	0.60586957044481
1.2000000000000000	0.62690662346739
1.2500000000000000	0.64328225465741
1.3000000000000000	0.65598701099456
1.3500000000000000	0.66579542096891
1.4000000000000000	0.67331687061853
1.4500000000000000	0.67903344928154
1.5000000000000000	0.68332837843330
1.5500000000000000	0.68650755962792
1.6000000000000000	0.68881604314363
1.6500000000000000	0.69045071166135
1.7000000000000000	0.69157011862770
1.7500000000000000	0.69230217013445
1.8000000000000000	0.69275015988417
1.8500000000000000	0.69299753741611
1.9000000000000000	0.69311169549457
1.9500000000000000	0.69314699327923
2.0000000000000000	0.69314718055995

La gráfica obtenida es la que sigue:



2.3. Código fuente.

Utilizaremos dos ficheros *coef.m* que es donde estarán p, q y r , y el programa principal que estará en *diferl.m*:

```
format long;
clear;
aux1=input('Escriba los puntos donde se conoce su valor [a,b]=');
aux2=input('Escriba los valores [y(a),y(b)]=');
n=input('Escriba el número de divisiones (n-1) n= ');

ao=aux1(1);
bo=aux1(2);
h=(bo-ao)/(n+1);
x=ao+h;
y=coef(x);
a(1)=2+((h^2)*y(2));
b(1)=-1+((h/2)*y(1));
d(1)=-((h^2)*y(3))+((1+(h/2)*y(1))*aux2(1));

for i=2:n-1
    x=ao+i*h;
    y=coef(x);
    a(i)=2+((h^2)*y(2));
    b(i)=-1+((h/2)*y(1));
    c(i)=-1-((h/2)*y(1));
    d(i)=-((h^2)*y(3));
end;

x=bo-h;
y=coef(x);
a(n)=2+(h^2)*y(2);
c(n)=-1-(h/2)*y(1);
d(n)=-((h^2)*y(3))+((1-(h/2)*y(1))*aux2(2));

l(1)=a(1);
```

```
u(1)=b(1)/a(1);
for i=2:n-1
    l(i)=a(i)-(c(i)*u(i-1));
    u(i)=b(i)/l(i);
end;

l(n)=a(n)-(c(n)*u(n-1));
z(1)=d(1)/l(1);
for i=2:n
    z(i)=(d(i)-c(i)*z(i-1))/l(i);
end;

w(1)=aux2(1);
w(n+2)=aux2(2);
w(n+1)=z(n);

i=n;
while i>=2
    w(i)=z(i-1)-(u(i-1)*w(i+1));
    i=i-1;
end;

for i=1:n+2
    x(i)=ao+(i-1)*h;
    %w(i)
end

for k=1:n+2
    M(k,:)=[x(k) w(k)];
end
M
plot(x,w);
```

3.-MÉTODO DEL DISPARO PARA PROBLEMAS NO LINEALES.

3.1-Conceptos teóricos.



La técnica del disparo para un problema no lineal de valor de frontera de 2º orden

$$(0) \quad y'' = f(x, y, y') \quad a \leq x \leq b, \quad y(a) = \alpha, \quad y(b) = \beta,$$

es similar a la del caso lineal, excepto en que la solución al problema no lineal no puede expresarse simplemente como una combinación lineal de las soluciones de dos problemas de valor inicial. En su lugar, necesitamos utilizar las soluciones a una sucesión de problemas de valor inicial de la forma

$$(1) \quad y'' = f(x, y, y') \quad a \leq x \leq b, \quad y(a) = \alpha, \quad y(b) = t,$$

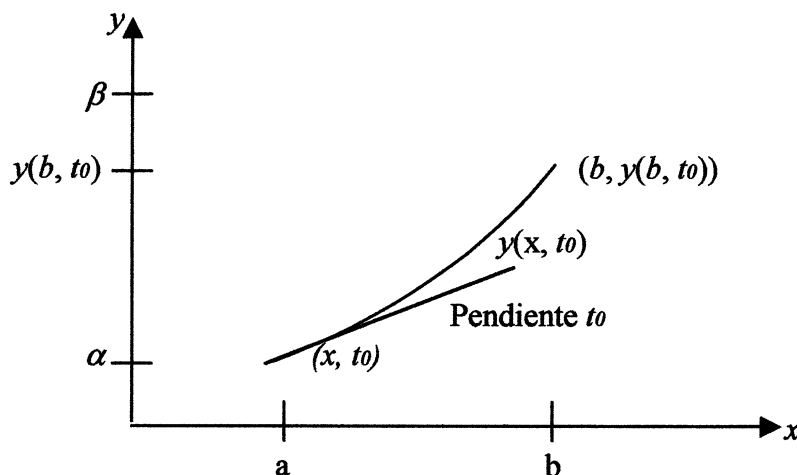
que involucran un parámetro t , para aproximar la solución de nuestro problema de valor de frontera. Hacemos esto escogiendo los parámetros $t = t_k$ de manera tal que se asegure que

$$(2) \quad \lim_{k \rightarrow \infty} y(b, t_k) = y(b) = \beta$$

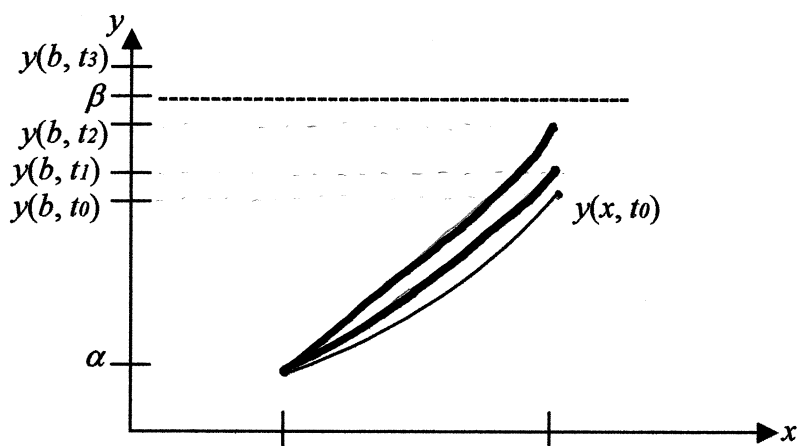
donde $y(x, t_k)$ denota la solución del problema de valor inicial (1) con $t = t_k$ e $y(x)$ denota la solución del problema de valor de frontera (0).

Esta técnica se llama método del “disparo” en analogía con el procedimiento de disparar objetos a un blanco estacionario (*ver la siguiente figura*). Empezamos con un parámetro t_0 que determina la elevación inicial a la cuál el objeto será disparado del punto (a, α) y a lo largo de la curva descrita por la solución del problema de valor inicial:

$$(3) \quad y'' = f(x, y, y') \quad a \leq x \leq b, \quad y(a) = \alpha, \quad y(b) = t_0.$$



Si $y(b, t_0)$ no está lo suficientemente cerca de β , tratamos de corregir nuestra aproximación escogiendo otra elevación t_1 y así sucesivamente hasta que $y(b, t_k)$ está lo suficientemente cerca para "pegarle" a β (ver la siguiente figura).



Si se usa a $y(x, t)$ para denotar a la solución del problema de valor inicial (1), el problema consiste en determinar t tal que

$$(4) \quad y(b, t) - \beta = 0.$$

Si deseamos emplear el método de la secante para resolver el problema, necesitamos escoger dos aproximaciones iniciales t_0 y t_1 y con éstas generar los términos siguientes de la sucesión:

$$t_k = t_{k-1} - \frac{(y(b, t_{k-1}) - \beta)(t_{k-1} - t_{k-2})}{y(b, t_{k-1}) - y(b, t_{k-2})}, \quad k=2, 3, \dots$$

Para usar el método más poderoso de Newton para generar la sucesión $\{t_k\}$ solamente es necesario un valor inicial t_0 . Sin embargo, la iteración tendrá la forma:

$$(5) \quad t_k = t_{k-1} - \frac{(y(b, t_{k-1}) - \beta)}{(dy/dt)(b, t_{k-1})} \quad \text{donde} \quad (dy/dt)(b, t_{k-1}) \equiv \frac{dy}{dt}(b, t_{k-1})$$

y requerirá conocer $(dy/dt)(b, t_{k-1})$. Ésto representa una dificultad, ya que no se tiene una representación explícita para $y(b, t)$; se saben solamente los valores $y(b, t_0), y(b, t_1), \dots, y(b, t_{k-1})$.

Supongamos que reescribimos el problema de valor inicial (1), enfatizando que la solución depende tanto de x como de t :

$$(6) \quad y''(x, t) = f(x, y(x, t), y'(x, t)) \quad a \leq x \leq b, \quad y(a, t) = \alpha, \quad y'(a, t) = t,$$

reteniendo la notación de primas para indicar diferenciación con respecto a x . Como estamos interesados en determinar $(dy/dt)(b,t)$ cuando $t_k = t_{k-1}$, tomamos primero la derivada parcial de (6) con respecto a t . Esto implica que

$$(\partial y''/\partial t)(x,t) = (\partial f/\partial t)(x,y(x,t),y'(x,t)) = (\partial f/\partial x)(x,y(x,t),y'(x,t)) \cdot (\partial x/\partial t) + \\ + (\partial f/\partial y)(x,y(x,t),y'(x,t)) \cdot (\partial y/\partial t)(x,t) + (\partial f/\partial y')(x,y(x,t),y'(x,t)) \cdot (\partial y'/\partial t)(x,t)$$

o como x y t son independientes,

$$(7) \quad (\partial y''/\partial t)(x,t) = (\partial f/\partial y)(x,y(x,t),y'(x,t)) \cdot (\partial y/\partial t)(x,t) + \\ + (\partial f/\partial y')(x,y(x,t),y'(x,t)) \cdot (\partial y'/\partial t)(x,t)$$

para $a \leq x \leq b$. Las condiciones iniciales dan:

$$(\partial y/\partial t)(a,t) = 0 \quad \text{y} \quad (\partial y'/\partial t)(a,t) = 1.$$

Si simplificamos la notación usando $z(x,t)$ para denotar a $(\partial y/\partial t)(x,t)$ y suponemos que se puede intercambiar el orden de diferenciación de x y t , tendremos el problema de valor inicial

$$(8) \quad z'' = (\partial f/\partial y)(x,y,y') \cdot z + (\partial f/\partial y')(x,y,y') \cdot z' \quad a \leq x \leq b, \quad z(a) = 0, \quad z'(a) = 1.$$

El método de Newton por lo tanto, requiere que los dos problemas de valor inicial, ecuaciones (1) y (8), sean resueltos en cada iteración. Con esto, se obtiene de la ecuación (5),

$$t_k = t_{k-1} - \frac{y(b, t_{k-1}) - \beta}{z(b, t_{k-1})}.$$

3.2-Ejemplos de aplicación del algoritmo.

3.2.1.-Consideremos el siguiente problema de contorno no lineal:

$$y'' = \frac{32 + 2x^3 - yy'}{8} \quad 1 \leq x \leq 3, \quad y(1) = 17, \quad y(3) = 43/3,$$

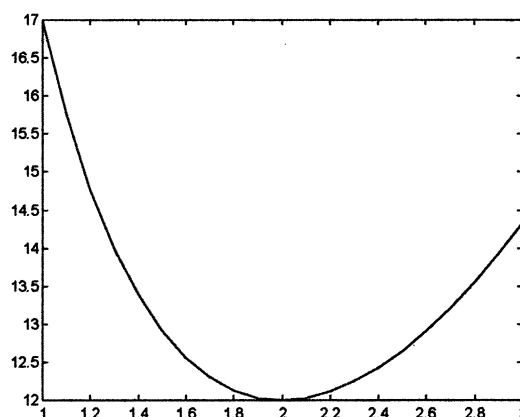
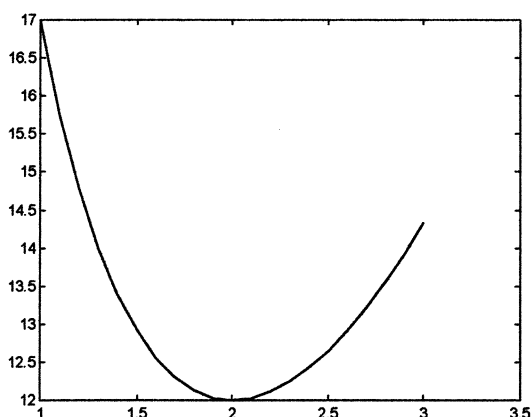
el cuál tiene solución exacta :

$$y(x) = x^2 + \frac{16}{x}$$

En la siguiente tabla se muestran los valores obtenidos para cada punto del intervalo tomando $h=0.1$ y los valores reales de la función que es solución del problema. Se toma una tolerancia de 10^{-6} como criterio de paro, es decir, cuando las soluciones obtenidas en cada iteración no difieran de la anterior en menos de 10^{-6} , se terminará el algoritmo.

x_i	W_i	$y(x_i)$	$ y(x_i) - w_i $
1.0000000000000000	17.000000000000000	17.000000000000000	0
1.1000000000000000	15.75549615760178	15.75545454545455	$0.41612147235526 \cdot 10^{-4}$
1.2000000000000000	14.77339118152161	14.77333333333333	$0.57848188278697 \cdot 10^{-4}$
1.3000000000000000	13.99775431510371	13.99769230769231	$0.62007411406384 \cdot 10^{-4}$
1.4000000000000000	13.38863181204382	13.38857142857143	$0.60383472389347 \cdot 10^{-4}$
1.5000000000000000	12.91672274127645	12.91666666666667	$0.56074609778634 \cdot 10^{-4}$
1.6000000000000000	12.56005064702693	12.56000000000000	$0.50647026931472 \cdot 10^{-4}$
1.7000000000000000	12.30180960866013	12.30176470588235	$0.44902777773359 \cdot 10^{-4}$
1.8000000000000000	12.12892813987447	12.12888888888889	$0.39250985576444 \cdot 10^{-4}$
1.9000000000000000	12.03108652581501	12.03105263157895	$0.33894236066701 \cdot 10^{-4}$
2.0000000000000000	12.00002892508545	12.00000000000000	$0.28925085446829 \cdot 10^{-4}$
2.1000000000000000	12.02907199629153	12.02904761904762	$0.24377243908802 \cdot 10^{-4}$
2.2000000000000000	12.11274752592619	12.11272727272727	$0.20253198918851 \cdot 10^{-4}$
2.3000000000000000	12.24653827841163	12.24652173913044	$0.16539281192607 \cdot 10^{-4}$
2.4000000000000000	12.42667988057069	12.42666666666667	$0.13213904020759 \cdot 10^{-4}$
2.5000000000000000	12.65001025203833	12.65000000000000	$0.10252038329384 \cdot 10^{-4}$
2.6000000000000000	12.91385378144209	12.91384615384616	$0.07627595932647 \cdot 10^{-4}$
2.7000000000000000	13.21593124057681	13.21592592592593	$0.05314650886490 \cdot 10^{-4}$
2.8000000000000000	13.55428900230881	13.55428571428571	$0.03288023098236 \cdot 10^{-4}$
2.9000000000000000	13.92724290283245	13.92724137931035	$0.01523522099589 \cdot 10^{-4}$
3.0000000000000000	14.33333333135378	14.33333333333333	$0.00001979548969 \cdot 10^{-4}$

La gráfica que se obtiene en azul es la aproximación obtenida, y la roja es la función real:



Si observamos el error en la aproximación, podemos observar que es menor que en la solución aproximada obtenida por el método de las diferencias finitas. Esto puede ser debido a que éste precisa de un mayor trabajo para alcanzar una precisión específica. No obstante, presenta menos problemas de estabilidad que el método del disparo (tanto en el caso lineal como en el no lineal).

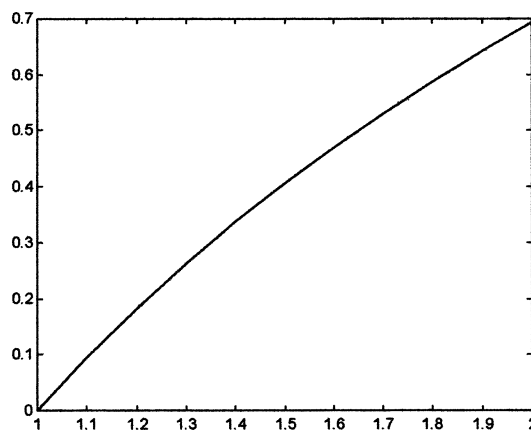
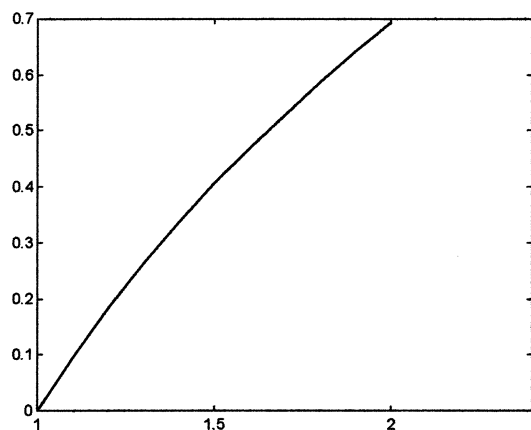
3.2.2.- Resolvamos el siguiente problema:

$$y'' = -(y')^2 - y + \ln x, \quad 1 \leq x \leq 2, \quad \text{para } y(1)=0, \quad y(2)=\ln(2) \text{ y } n=10.$$

Consideraremos una tolerancia de 10^{-6} ; y se comparará con la función real que es $y = \ln(x)$.

x_i	w_i	$y=\ln(x_i)$	$ y(x_i) - w_i $
1	0	0	0
1.1000000000000000	0.09531002489781	0.09531017980432	$0.15490651844585 \cdot 10^{-6}$
1.2000000000000000	0.18232135393911	0.18232155679395	$0.20285484500504 \cdot 10^{-6}$
1.3000000000000000	0.26236406271263	0.26236426446749	$0.20175485704321 \cdot 10^{-6}$
1.4000000000000000	0.33647205766145	0.33647223662121	$0.17895976589521 \cdot 10^{-6}$
1.5000000000000000	0.40546496043813	0.40546510810816	$0.14767003281291 \cdot 10^{-6}$
1.6000000000000000	0.47000351496060	0.47000362924574	$0.11428513352341 \cdot 10^{-6}$
1.7000000000000000	0.53062816921185	0.53062825106217	$0.08185031674657 \cdot 10^{-6}$
1.8000000000000000	0.58778661317074	0.58778666490212	$0.05173138328818 \cdot 10^{-6}$
1.9000000000000000	0.64185386172132	0.64185388617239	$0.02445107094573 \cdot 10^{-6}$
2.0000000000000000	0.69314718044504	0.69314718055995	$0.00011490608465 \cdot 10^{-6}$

Al igual que antes, la gráfica que se obtiene en azul es la aproximación obtenida, y la roja es la función real:



Se puede observar en la tabla y en las gráficas, que se obtiene una aproximación muy buena, y que en algunos valores de w_i , se obtienen hasta 7 cifras decimales exactas. Y se puede también ver, que el error es menor para este ejemplo que para el anterior.

3.2.3.-Consideremos el siguiente problema de contorno no lineal:

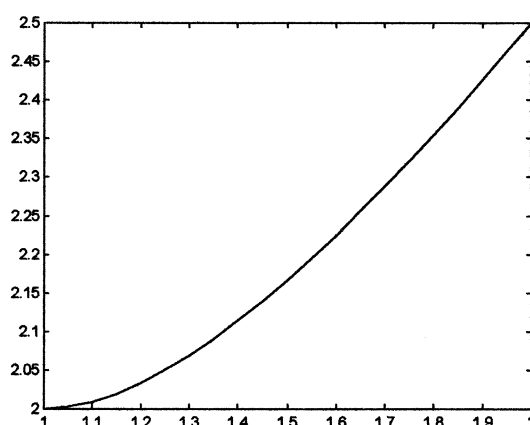
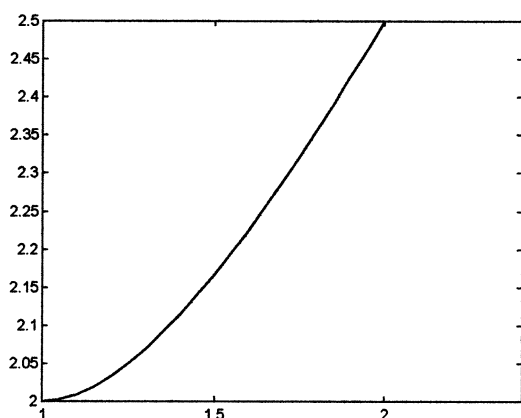
$$y'' = 2y^3 - 6y - 2x^3 \quad 1 \leq x \leq 2 \quad \text{para} \quad y(1)=2, \quad y(2)=5/2$$

Se utilizará $h=0.05$, es decir $n=20$; y una tolerancia de 10^{-6} . Se comparará con la solución real que es:

$$y(x) = x + \frac{1}{x}$$

x_i	w_i	$y(x_i)$	$ y(x_i) - w_i $
1.0000000000000000	2.0000000000000000	2.0000000000000000	0
1.0500000000000000	2.00238134028864	2.00238095238095	$0.03879076881930 \cdot 10^{-5}$
1.1000000000000000	2.00909156196760	2.00909090909091	$0.06528766900438 \cdot 10^{-5}$
1.1500000000000000	2.01956604470064	2.01956521739130	$0.08273093397015 \cdot 10^{-5}$
1.2000000000000000	2.03333426832811	2.03333333333333	$0.09349947736048 \cdot 10^{-5}$
1.2500000000000000	2.05000099351595	2.05000000000000	$0.09935159459573 \cdot 10^{-5}$
1.3000000000000000	2.06923178513392	2.06923076923077	$0.10159031513801 \cdot 10^{-5}$
1.3500000000000000	2.09074175252952	2.09074074074074	$0.10117887798700 \cdot 10^{-5}$
1.4000000000000000	2.11428670250924	2.11428571428571	$0.09882235243452 \cdot 10^{-5}$
1.4500000000000000	2.13965612267014	2.13965517241379	$0.09502563513841 \cdot 10^{-5}$
1.5000000000000000	2.16666756801075	2.16666666666667	$0.09013440851469 \cdot 10^{-5}$
1.5500000000000000	2.19516213395526	2.19516129032258	$0.08436326748296 \cdot 10^{-5}$
1.6000000000000000	2.22500077813606	2.22500000000000	$0.07781360591430 \cdot 10^{-5}$
1.6500000000000000	2.25606131088737	2.25606060606061	$0.07048267609555 \cdot 10^{-5}$
1.7000000000000000	2.28823591676055	2.28823529411765	$0.06226429052347 \cdot 10^{-5}$
1.7500000000000000	2.32142910083628	2.32142857142857	$0.05294077038975 \cdot 10^{-5}$
1.8000000000000000	2.35555597720389	2.35555555555556	$0.04216483353581 \cdot 10^{-5}$
1.8500000000000000	2.39054083483087	2.39054054054054	$0.02942903316594 \cdot 10^{-5}$
1.9000000000000000	2.42631592966320	2.42631578947368	$0.01401895124431 \cdot 10^{-5}$
1.9500000000000000	2.46282046226486	2.46282051282051	$0.00505556529973 \cdot 10^{-5}$
2.0000000000000000	2.49999970840456	2.50000000000000	$0.02915954353533 \cdot 10^{-5}$

Las gráficas de la solución obtenida y de la función real, se muestran a continuación:



3.3.- Código fuente.

El código está compuesto por los siguientes ficheros:

- Fichero *displ.m*: es el que contiene el programa principal.

```
%Método del disparo para problemas de contorno no lineales.
clear;
a=input('Escriba el punto inicial a= ');
b=input('Escriba el punto final b= ');
alfa=input('Escriba el valor de y(a)= ');
beta=input('Escriba el valor de y(b)= ');
n=input('Escriba el número de divisiones (n+1) n= ');
tol=input('Escriba la tolerancia=');

% Resolución del primer sistema:

fin=0;
to=(beta-alfa)/(b-a);
while (fin==0)

    ybto=ec2(a,[alfa to],b,n,1,0); % 0 por poner algo.

% Resolución del primer sistema para h/2.Éste resultado nos
% servirá para resolver el segundo sistema.
    aux=ec2(a,[alfa to],b,2*n,1,0);

% Resolución del segundo sistema.Necesitaremos pasarle esta
% variable aux.
    zbto=ec2(a,[0 1],b,n,2,aux);

    t1=to-((ybto(n,1)-beta)/zbto(n,1));

    if (abs(t1-to)<tol)
        t=t1;
        fin=1;
    else
        to=t1;
    end
end

'La solución es:',sol=ec2(a,[alfa t],b,n,1,0)
'pulse una tecla para ver el gráfico de la función'
pause
plot(sol(:,3),sol(:,1));
hold on;
```

- Fichero *ec2.m*:

```
% Esta función calcula la solución de un sistema de 2°
%orden utilizando Runge-Kut.
```

```

function valor=ec2(to,w,t,n,sist,yyp)
% to es el punto inicial.
% w es el valor inicial y(to)=[  ] ;
% t es el punto para calcular su valor.
% n es el número de divisiones.
% sist es una variable que indica si se resuelve el primer
% sistema ó el segundo.
h=(t-to)/(n-1);
for k=1:n
    M(k,:)=w;
    a(k,:)=to;
    if (sist==1)
        w=r(to,w,h);
    elseif (sist==2)
        w=r2(to,w,h,yyp,k);
    else
        error('ERROR EN LA FUNCIÓN ec2:Debe especificar
sistema 1 ó 2. ');
        break;
    end
    to=to+h;
end
valor=[M a];

```

- Fichero *f.m*:

```

function valor=f(x,v) %f(t,y,z)
%Sustituir z por y' en la ecuación y'='....
y=v(1);
z=v(2);

valor=[z,2*(y^3)-6*y-2*(x^3)];

```

- Fichero *f2.m*:

```

function valor=f2(x,v,yyp);
z=v(1);
q=v(2);
y=yyp(1);
yp=yyp(2);
valor=[q,fy(x,y,yp)*z+fyp(x,y,yp)*q];

```

- Fichero *fy.m*:

```

function valor=fy(x,y,yp) % Derivada de y' respecto de y.

valor=6*(y^2)-6;

```

- Fichero *fyp.m*:

```
function valor=fyp(x,y,yp) %Derivada de y' respecto de y
%prima.
```

```
valor=0;
```

- Fichero *R.m*:

```
function yn=r(x,w,h)
k1=h*f(x,w);
k2=h*f(x+h/2,w+k1/2);
k3=h*f(x+h/2,w+k2/2);
k4=h*f(x+h,w+k3);
yn=w+1/6*(k1+2*k2+2*k3+k4);
```

- Fichero *R2.m*:

```
function yn=r2(x,w,h,yyp,i)
k1=h*f2(x,w,yyp((2*i)-1,:));
k2=h*f2(x+h/2,w+k1/2,yyp(2*i,:));
k3=h*f2(x+h/2,w+k2/2,yyp(2*i,:));
k4=h*f2(x+h,w+k3,yyp((2*i)-1,:));
yn=w+1/6*(k1+2*k2+2*k3+k4);
```

4.- MÉTODO DE LAS DIFERENCIAS FINITAS NO LINEAL

4.1. Conceptos teóricos.

Sea la siguiente ecuación para el problema general de valor de frontera no lineal:

$$y'' = f(x, y, y'), a \leq x \leq b, y(a) = \alpha, y(b) = \beta,$$

para el desarrollo del procedimiento de resolución, supondremos que f satisface:

- i) f y las derivadas parciales $f_y \equiv \partial f / \partial y$ y $f_{y'} \equiv \partial f / \partial y'$ son todas continuas en $D = \{(x, y, y') \mid a \leq x \leq b, -\infty < y < \infty, -\infty < y' < \infty\}$;
- ii) $f_y(x, y, y') \geq \delta > 0$ en D para alguna $\delta > 0$;
- iii) existen constantes k y L con

$$k = \max_{(x, y, y') \in D} |f_y(x, y, y')|, \quad L = \max_{(x, y, y') \in D} |f_{y'}(x, y, y')|.$$

Esto nos asegura que existe una única solución a la ecuación.

Dividimos a $[a, b]$ en $(N+1)$ subintervalos iguales cuyos puntos extremos están en los puntos de la red $x_i = a + ih$ para $i=0, 1, \dots, N+1$.

Suponemos que la solución tiene una cuarta derivada acotada, lo que nos permite reemplazar a $y''(x_i)$ y a $y'(x_i)$ en cada una de las ecuaciones

$$y''(x_i) = f(x_i, y(x_i), y'(x_i))$$

por la fórmula de diferencia centrada apropiada para obtener para cada $i=1, 2, \dots, N$,

$$\frac{y(x_{i+1}) - 2y(x_i) + y(x_{i-1}))}{h^2} = f\left(x_i, y(x_i), \frac{y(x_{i+1}) - y(x_{i-1}))}{2h} - \frac{h^2}{6} y'''(\eta_i)\right) + \frac{h^2}{12} y^{(4)}(\xi_i),$$

para algunas ξ_i y η_i en el intervalo (x_{i-1}, x_{i+1}) .

El método resulta de eliminar los términos de error y de emplear las condiciones de frontera:

$$w_0 = \alpha \quad w_{N+1} = \beta,$$

y

$$-\frac{w_{i+1} - 2w_i + w_{i-1}}{h^2} + f\left(x_i, w_i, \frac{w_{i+1} - w_{i-1}}{2h}\right) = 0,$$

para cada $i=1, 2, \dots, N$.

El sistema no lineal de $N \times N$ que se obtiene a partir de este método:

$$\begin{aligned} 2w_1 - w_2 + h^2 f\left(x_1, w_1, \frac{w_2 - \alpha}{2h}\right) - \alpha &= 0, \\ -w_1 + 2w_2 - w_3 + h^2 f\left(x_2, w_2, \frac{w_3 - w_1}{2h}\right) &= 0, \\ &\vdots \\ -w_{N-2} + 2w_{N-1} - w_N + h^2 f\left(x_{N-1}, w_{N-1}, \frac{w_N - w_{N-2}}{2h}\right) &= 0, \\ -w_{N-1} + 2w_N + h^2 f\left(x_N, w_N, \frac{\beta - w_{N-1}}{2h}\right) - \beta &= 0. \end{aligned}$$

tendrá una solución única siempre que $h < 2/L$, como se demuestra en el libro de Séller. Para aproximar la solución de este sistema usamos el método de Newton para sistemas no lineales. Iterando se genera una sucesión que convergerá a la solución del sistema anterior cuando la aproximación inicial esté lo suficiente cerca de la solución $(w_1, w_2, \dots, w_N)^1$, y que la matriz Jacobiana del sistema sea no-singular. Sin embargo la matriz Jacobiana es tridiagonal y las suposiciones presentadas al principio de esta discusión aseguran que J es una matriz no-singular.

$$J(w_1, \dots, w_N) = \begin{bmatrix} 2+h^2 f_y\left(x_1, w_1, \frac{w_2 - \alpha}{2h}\right) & -1+h f_y\left(x_1, w_1, \frac{w_2 - \alpha}{2h}\right) & 0 & \dots & 0 \\ -1+h f_y\left(x_2, w_2, \frac{w_3 - w_1}{2h}\right) & 2+h^2 f_y\left(x_2, w_2, \frac{w_3 - w_1}{2h}\right) & \cdot & & \cdot \\ 0 & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & 0 \\ \cdot & \cdot & \cdot & & -1+h f_y\left(x_{N-1}, w_{N-1}, \frac{w_N - w_{N-2}}{2h}\right) \\ 0 & \cdot & \cdot & \cdot & 0 & -1+h f_y\left(x_N, w_N, \frac{\beta - w_{N-1}}{2h}\right) & 2+h^2 f_y\left(x_N, w_N, \frac{\beta - w_{N-1}}{2h}\right) \end{bmatrix}$$

El método de Newton para sistemas no lineales requiere que en cada iteración, se resuelva el sistema lineal de $N \times N$:

$$J(w_1, \dots, w_N)(v_1, \dots, v_n)^t = -\left(2w_1 - w_2 - \alpha + h^2 f\left(x_1, w_1, \frac{w_2 - \alpha}{2h}\right)\right),$$

$$\begin{aligned}
 & -w_1 - 2w_2 - w_3 + h^2 f\left(x_2, w_2, \frac{w_3 - w_1}{2h}\right), \dots, \\
 & -w_{N-2} - 2w_{N-1} - w_N + h^2 f\left(x_{N-1}, w_{N-1}, \frac{w_N - w_{N-2}}{2h}\right), \\
 & -w_{N-1} - 2w_N + h^2 f\left(x_N, w_N, \frac{\beta - w_{N-1}}{2h}\right) - \beta)'
 \end{aligned}$$

para v_1, \dots, v_N , ya que

$$w_i^{(k)} = w_i^{(k-1)} + v_i, \text{ para cada } i=1,2,\dots,N.$$

Como f es tridiagonal aplicaremos el método de Crout para resolverla de forma más eficiente.

4.2. Ejercicios de aplicación.

4.2.1.- Resolver la siguiente ecuación $y'' = 7xy^3 - \sin(y') + 7x^2$ para los valores iniciales $y(0)=1$, $y(1)=27$, una tolerancia de 10^{-6} y un número máximo de iteraciones de 500.

Los resultados obtenidos para las diferentes n que tomamos nos muestra, como era de esperar, que a mayor n más se asemejan los resultados a la gráfica y original, y como suponemos los valores obtenidos en cada x_i también, aunque no tenemos la función original para poder compararlos.

Metemos en la tabla siguiente los valores obtenidos por el método:

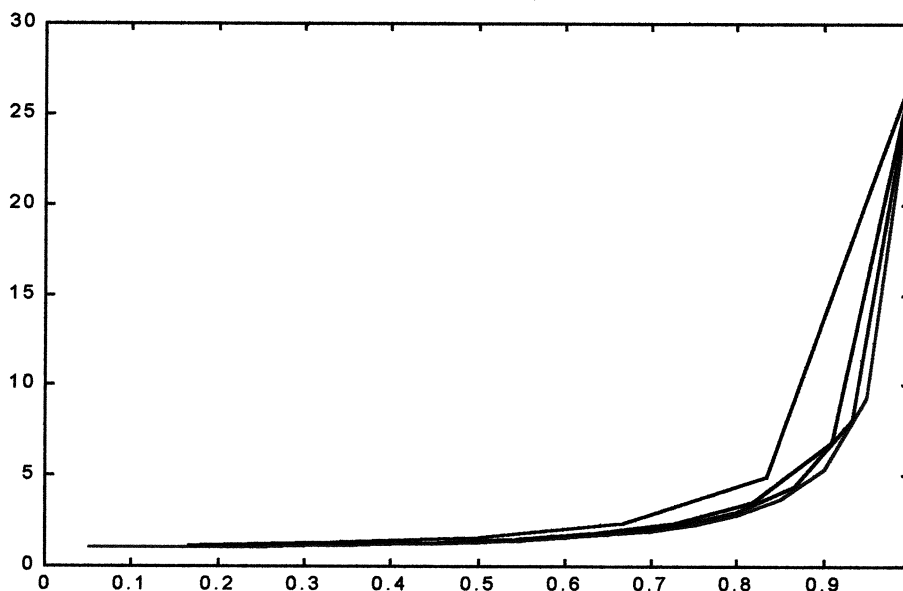
X(i)	W(i)	X(i)	W(i)
	N=19		N=14
0	1.0000000000000000	0	1.0000000000000000
0.0500000000000000	1.01505481627080	0.0666666666666667	1.02406753406852
0.1000000000000000	1.03032201195240	0.1333333333333333	1.04890688204803
0.1500000000000000	1.04689465086351	0.2000000000000000	1.07735577084635
0.2000000000000000	1.06599870946639	0.2666666666666667	1.11278003641983
0.2500000000000000	1.08901889675960	0.3333333333333333	1.15928461313663
0.3000000000000000	1.11754934701357	0.4000000000000000	1.22215123665453
0.3500000000000000	1.15347858021992	0.4666666666666667	1.30870942225589
0.4000000000000000	1.19912944102852	0.5333333333333333	1.43014006855808
0.4500000000000000	1.25749229610274	0.6000000000000000	1.60542774327574
0.5000000000000000	1.33262579869066	0.6666666666666667	1.86984548482695
0.5500000000000000	1.43037194862283	0.7333333333333333	2.28776881858245
0.6000000000000000	1.55966551729256	0.8000000000000000	2.99184231392639
0.6500000000000000	1.73485341809485	0.8666666666666667	4.38236647585258
0.7000000000000000	1.97899879444793	0.9333333333333333	8.06407462514157
0.7500000000000000	2.32753605323118	1.0000000000000000	27.00000000000000
0.8000000000000000	2.84977026382137		
0.8500000000000000	3.70487730081769		
0.9000000000000000	5.32988874265032		
0.9500000000000000	9.35392729655448		
1.0000000000000000	27.00000000000000		

X(i)	W(i)	X(i)	W(i)
	N=10		N=5
0	1	0	1.0000000000000000
0.1000000000000000	1.04995684527298	0.1666666666666667	1.13547565130580
0.2000000000000000	1.10375777891541	0.3333333333333333	1.30193709544993
0.3000000000000000	1.17339679768933	0.5000000000000000	1.60561618353215
0.4000000000000000	1.27568735517520	0.6666666666666667	2.36135383350943
0.5000000000000000	1.43761676121863	0.8333333333333333	4.92441108295398
0.6000000000000000	1.71287052288739	1.0000000000000000	27.0000000000000000
0.7000000000000000	2.23176613157244		
0.8000000000000000	3.31980542287510		
0.9000000000000000	6.49533323846172		
1	27		

En la gráfica siguiente dibujamos las gráficas obtenidas para distintas N, a menor N menos perfecta el gráfica resultante y más se aleja de la original:

Leyenda:

—— Para N=10, —— Para N=5, —— Para N=14 y —— Para N=19.



Aunque no queda reflejado en las tablas también pudimos comprobar que el número de operaciones y el tiempo empleado en calcular los resultados aumenta al tiempo que lo hace la N, como era de esperar, al dividir el intervalo en mayor número de partes y tener que realizar mayor número de operaciones.

También estuvimos probando con el número máximo de iteraciones, en principio 100, y no podíamos llegar a la solución con $n=19$, ya que se realizaban mayor número de iteraciones que las impuestas, con lo que fuimos probando hasta establecer este número en 500.

4.2.2. Resolveremos la siguiente ecuación, que nos servirá para comparar posteriormente el método de las diferencias finitas, con el método del disparo. Para este caso tomaremos las siguientes consideraciones sobre la ecuación que nos ocupa:

$y(1)=17, y(3)=43/3, n=20$ y la tolerancia será 10^{-6} con la ecuación :

$$y'' = \frac{32 + 2x^3 - yy'}{8}$$

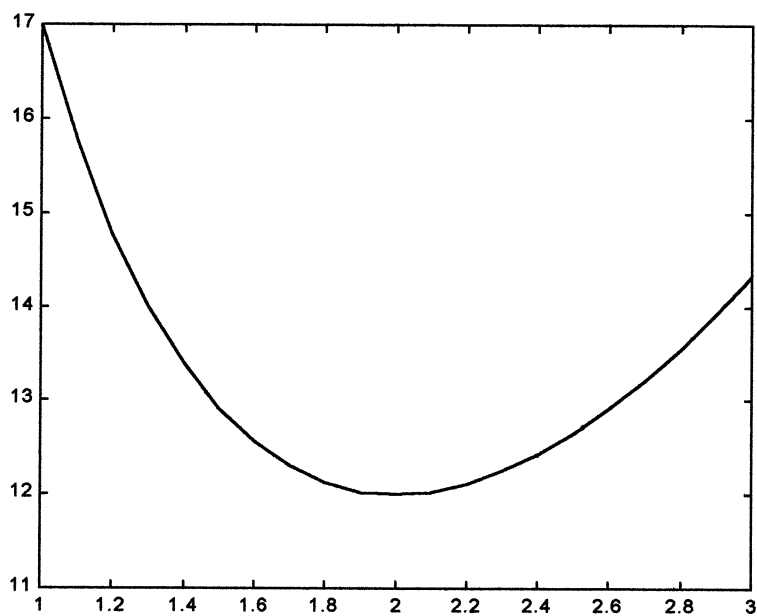
Los resultados que obtenemos son los que siguen:

X(i)	W(i)	Y(X(i))	ERROR
1	17	17	-
1.1000000000000000	15.75450253529389	15.755455	$9,524 \cdot 10^{-4}$
1.2000000000000000	14.77173965356983	14.773333	$1,593 \cdot 10^{-3}$
1.3000000000000000	13.99567743713438	13.997692	$2,014 \cdot 10^{-3}$
1.4000000000000000	13.38629656205142	13.388571	$2,274 \cdot 10^{-3}$
1.5000000000000000	12.91425241252499	12.916667	$2,214 \cdot 10^{-3}$
1.6000000000000000	12.55753822750665	12.560000	$2,246 \cdot 10^{-3}$
1.7000000000000000	12.29932628240155	12.301765	$2,438 \cdot 10^{-3}$
1.8000000000000000	12.12652886701783	12.128889	$2,360 \cdot 10^{-3}$
1.9000000000000000	12.02881380980783	12.031053	$2,239 \cdot 10^{-3}$
2.0000000000000000	11.99791542246046	12.000000	$2,085 \cdot 10^{-3}$
2.1000000000000000	12.02714237122990	12.029048	$1,906 \cdot 10^{-3}$
2.2000000000000000	12.11101980435058	12.112727	$1,706 \cdot 10^{-3}$
2.3000000000000000	12.24502486731439	12.246522	$1,496 \cdot 10^{-3}$
2.4000000000000000	12.42538836268850	12.426667	$1,278 \cdot 10^{-3}$
2.5000000000000000	12.64894403015380	12.650000	$1,056 \cdot 10^{-3}$
2.6000000000000000	12.91301262278355	12.913846	$8,334 \cdot 10^{-4}$
2.7000000000000000	13.21531175549766	13.215926	$6,141 \cdot 10^{-4}$
2.8000000000000000	13.55388507997838	13.554286	$4,006 \cdot 10^{-4}$
2.9000000000000000	13.92704611844152	13.927241	$1,953 \cdot 10^{-4}$
3.0000000000000000	14.33333333333333	14.333333	-

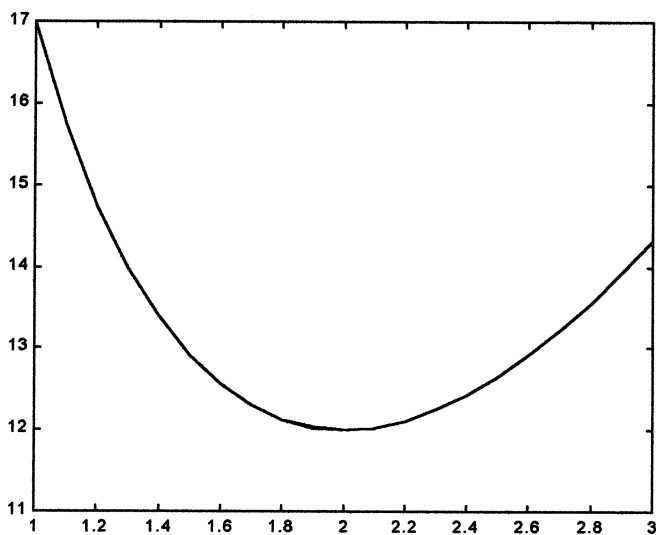
Con un número de operaciones realizado de 62519.

Obtenemos los resultados y comparamos con la función real, utilizando la tolerancia dada, observamos que el error es pequeño, pero este se hace menor en los extremos, esto puede ser debido a que en los extremos se parten de valores conocidos.

La gráfica resultante es la siguiente:



Ahora dibujamos la gráfica obtenida (en azul) junto con la real (en rojo), y como era de suponer se superponen una con otra y vemos que son casi idénticas:



4.2.3. Por último realizamos el siguiente ejercicio, resolveremos la siguiente ecuación

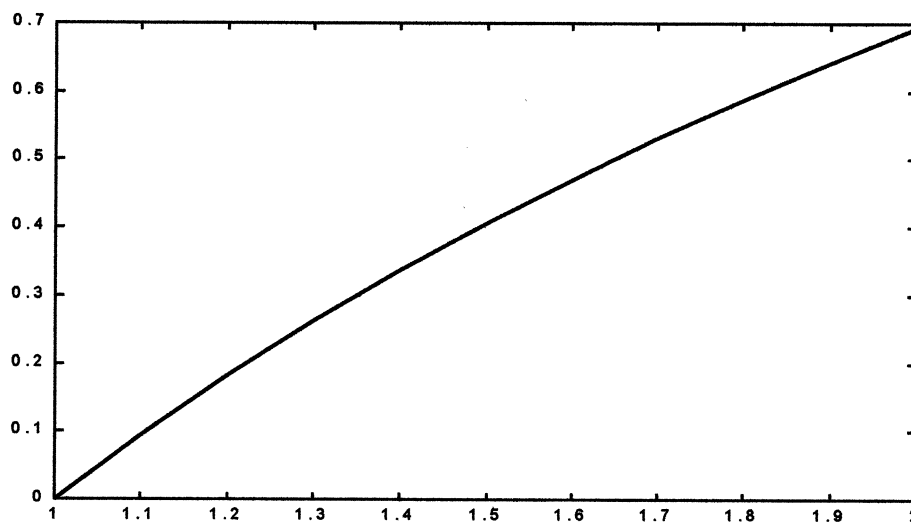
$$y'' = -(y')^2 - y + \ln x, \quad 1 \leq x \leq 2, \quad \text{para } y(1)=0, y(2)=\ln(2) \text{ y } n=10.$$

La tolerancia será de 10^{-6} .

Obtenemos los siguientes resultados:

X(i)	W(i)
1	0
1.1000000000000000	0.09534385581396
1.2000000000000000	0.18237246066301
1.3000000000000000	0.26242174505753
1.4000000000000000	0.33652929267920
1.5000000000000000	0.40551713259829
1.6000000000000000	0.47004763479629
1.7000000000000000	0.53066236914348
1.8000000000000000	0.58780981032359
1.9000000000000000	0.64186552734593
2.0000000000000000	0.69314718055995

La gráfica que obtenemos es la siguiente:



4.3. Código fuente.

Utilizaremos tres ficheros y otro para el programa principal `cnl.m`.

En el fichero `"f.m"` pondremos la función y' , el `"fy.m"` la derivada de y' con respecto de y' .

Y por último en `"fyprima.m"` pondremos la derivada de y' con respecto de y' .

El fichero `cnl.m` contendrá el siguiente código fuente:

```
% a y b => puntos extremos.
% alfa y beta => condiciones de frontera.
% n => numero de divisiones.
% tol => tolerancia
% max => n° maximo de iteraciones
```

```

clear;
a=input('INTRODUCE a:');
b=input('INTRODUCE b:');
alfa=input('INTRODUCE y(a):');
beta=input('INTRODUCE y(b):');
n=input('INTRODUCE n (n-1):');
max=input('INTRODUCE EL NUMERO MAXIMO DE ITERACIONES:');
tol=input('INTRODUCE TOLERANCIA: ');

h=(b-a)/(n+1);
w(1)=alfa;
w(n+2)=beta;

for i=1:n
    w(i+1)=alfa+(i+1)*((beta-alfa)/(b-a))*h;
end
k=1; %contador del n° de iteraciones.

while (k<=max)
    x=a+h;
    t=(w(3)-alfa)/(2*h);
    A(1)=2+(h^2)*fy(x,w(2),t); %fy es la derivada de f respecto de y
    B(1)=-1+(h/2)*fyprima(x,w(2),t); %fyprima es la derivada de f respecto de yprima
    D(1)=-2*w(2)-w(3)-alfa+(h^2)*f(x,w(2),t); %f es la funcion.

    for i=2:n-1
        x=a+i*h;
        t=(w(i+2)-w(i))/(2*h);
        A(i)=2+(h^2)*fy(x,w(i+1),t);
        B(i)=-1+(h/2)*fyprima(x,w(i+1),t);
        C(i)=-1-(h/2)*fyprima(x,w(i+1),t);
        D(i)=-2*w(i+1)-w(i+2)-w(i)+(h^2)*f(x,w(i+1),t);
    end

    x=b-h;
    t=(beta-w(n))/(2*h);
    A(n)=2+(h^2)*fy(x,w(n+1),t);
    C(n)=-1-(h/2)*fyprima(x,w(n+1),t);
    D(n)=-2*w(n+1)-w(n)-beta+(h^2)*f(x,w(n+1),t);

    %/crout vamos a resolver un sistema lineal tridiagonal

    L(1)=A(1);
    U(1)=B(1)/A(1);

    for i=2:n-1
        L(i)=A(i)-C(i)*U(i-1);
        U(i)=B(i)/L(i);
    end
end

```

```

L(n)=A(n)-C(n)*U(n-1);
Z(1)=D(1)/L(1);

for i=2:n
    Z(i)=(D(i)-C(i)*Z(i-1))/L(i);
end
V=0;
V(n)=Z(n);
w(n+1)=w(n+1)+V(n);

i=n-1;
while (i>=1)
    V(i)=Z(i)-(U(i)*V(i+1));
    w(i+1)=w(i+1)+V(i);
    i=i-1;
end
%fin de crout.

if (norm(V,inf)<=tol)
    for i=0:n+1
        x(i+1)=a+((i)*h);
    end
    break;
end
k=k+1;
end

for i=0:n+1
    M(i+1,:)= [x(i+1) w(i+1)];
end

'nº de iteraciones realizadas',k
M
plot(x,w);

```

5.- MÉTODO DE RAYLEIGH-RITZ

5.1. Conceptos teóricos.

Un problema lineal de valor de frontera de dos puntos que es importante en el análisis de las fuerzas aplicadas a vigas está dado por la ecuación diferencial

$$-\frac{d}{dx}\left(p(x)\frac{dy}{dx}\right) + q(x)y = f(x) \text{ para } 0 \leq x \leq 1,$$

con condiciones de frontera $y(0)=y(1)=0$.

Como en el caso de muchos problemas de valor de frontera que describen fenómenos físicos, la solución a la ecuación de la viga satisface una propiedad **variacional**. El principio variacional para la ecuación de la viga es fundamental en el desarrollo del método de Rayleigh-Ritz y caracteriza a la solución de la ecuación de la viga como la función que minimiza cierta integral sobre todas las funciones en $C_0^2[0,1]$, el conjunto de aquellas funciones u en $C^2[0,1]$ con la propiedad de que $u(0)=u(1)=0$. El siguiente teorema da la caracterización:

Sea $p \in C^1[0,1]$, $q, f \in C[0,1]$, y $p(x) \geq \delta > 0$, $q(x) \geq 0$ para $0 \leq x \leq 1$.

La función $y \in C_0^2[0,1]$, es la única solución de la ecuación diferencial

$$-\frac{d}{dx}\left(p(x)\frac{dy}{dx}\right) + q(x)y = f(x) \text{ para } 0 \leq x \leq 1,$$

si y sólo si y es la única función en $C_0^2[0,1]$, que minimiza la integral

$$I[u] = \int_0^1 \left\{ p(x)[u'(x)]^2 + q(x)[u(x)]^2 - 2f(x)u(x) \right\} dx.$$

en el procedimiento de Rayleigh-Ritz, la integral I se minimiza, no sobre todas las funciones en $C_0^2[0,1]$, sino sobre un conjunto más pequeño de funciones que consisten de combinaciones lineales de cierta base de funciones $\Phi_1, \Phi_2, \dots, \Phi_n$. Las funciones de la base $\{\Phi_i\}_{i=1}^n$ deben ser linealmente independientes y satisfacer

$$\Phi_i(0) = \Phi_i(1) = 0 \text{ para cada } i=1,2,\dots,n.$$

Una aproximación $\Phi(x) = \sum_{i=1}^n c_i \Phi_i(x)$ a la solución $y(x)$ de la ecuación de la viga se obtiene entonces encontrando constantes c_1, c_2, \dots, c_n que minimicen $I\left[\sum_{i=1}^n c_i \Phi_i\right]$.

De la ecuación de la integral,

$$I[\Phi] = I \left[\sum_{i=1}^n c_i \Phi_i \right]$$

$$= \int_0^1 \left\{ p(x) \left[\sum_{i=1}^n c_i \Phi'_i(x) \right]^2 + q(x) \left[\sum_{i=1}^n c_i \Phi_i(x) \right]^2 - 2f(x) \sum_{i=1}^n c_i \Phi_i(x) \right\} dx,$$

y para que ocurra un mínimo es necesario, al considerar I como función de c_1, c_2, \dots, c_n , tener

$$\frac{\partial I}{\partial c_j} = 0 \text{ para cada } j=1,2,\dots,n.$$

Diferenciando lo anterior, se obtiene

$$\frac{\partial I}{\partial c_j} = \int_0^1 \left\{ 2p(x) \sum_{i=1}^n c_i \Phi'_i(x) \Phi'_j(x) + 2q(x) \sum_{i=1}^n c_i \Phi_i(x) \Phi_j(x) - 2f(x) \Phi_j(x) \right\} dx,$$

y sustituyendo

$$0 = \sum_{i=1}^n \left[\int_0^1 \left\{ p(x) \Phi'_i(x) \Phi'_j(x) + q(x) \Phi_i(x) \Phi_j(x) \right\} dx \right] c_i - \int_0^1 f(x) \Phi_j(x) dx,$$

para cada $j=1,2,\dots,n$.

Las ecuaciones descritas pueden considerarse como un sistema lineal $Ac=b$, donde la matriz simétrica A está dada por

$$a_{ij} = \int_0^1 \left[p(x) \Phi'_i(x) \Phi'_j(x) + q(x) \Phi_i(x) \Phi_j(x) \right] dx$$

y b se define por

$$b_i = \int_0^1 f(x) \Phi_i(x) dx.$$

-Podemos distinguir entre dos tipos de funciones base:

5.2. Funciones segmentarias lineales.

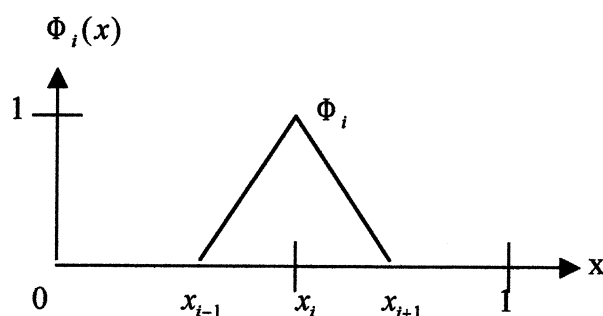
El primer paso consiste en formar una partición de $[0,1]$ escogiendo puntos x_0, x_1, \dots, x_{n+1} con

$$0 = x_0 < x_1 < \dots < x_n < x_{n+1} = 1.$$

Tomando $h_i = x_{i+1} - x_i$ para cada $i=0,1,\dots,n$, definimos las funciones base $\Phi_1(x), \Phi_2(x), \dots, \Phi_n(x)$ mediante

$$\Phi_i(x) = \begin{cases} 0, & 0 \leq x \leq x_{i-1}, \\ \frac{(x - x_{i-1})}{h_{i-1}}, & x_{i-1} < x \leq x_i, \\ \frac{(x_{i+1} - x)}{h_i}, & x_i < x \leq x_{i+1}, \\ 0, & x_{i+1} < x \leq 1, \end{cases}$$

para cada $i=1,2,\dots,n$.



Como las funciones Φ_i son lineales por pedazos, las derivadas Φ'_i , aunque no continuas, son constantes en el subintervalo abierto (x_j, x_{j+1}) para cada $j=0,1,\dots,n$. Por lo tanto, tenemos que

$$\Phi'_i(x) = \begin{cases} 0, & 0 \leq x \leq x_{i-1}, \\ \frac{1}{h_{i-1}}, & x_{i-1} < x \leq x_i, \\ -\frac{1}{h_i}, & x_i < x \leq x_{i+1}, \\ 0, & x_{i+1} < x \leq 1, \end{cases}$$

para cada $i=1,2,\dots,n$.

Debido a que Φ_i y Φ_j son diferentes de cero solamente en (x_{i-1}, x_{i+1}) ,

$$\Phi_i(x)\Phi_j(x) \equiv 0 \quad \text{y} \quad \Phi'_i(x)\Phi'_j(x) \equiv 0,$$

excepto cuando j es $i-1, i$ ó $i+1$. Como consecuencia, el sistema lineal descrito anteriormente, se reduce a un sistema lineal tridiagonal de $n \times n$. Las componentes distintas de cero de A están dadas por

$$a_{ii} = \int_0^1 \left\{ p(x) [\Phi'_i(x)]^2 + q(x) [\Phi_i(x)]^2 \right\} dx$$

$$\begin{aligned}
&= \int_{x-1}^{x_i} \left(\frac{1}{h_{i-1}}\right)^2 p(x) dx + \int_{x_i}^{x_{i+1}} \left(\frac{-1}{h_i}\right)^2 p(x) dx \\
&\quad + \int_{x-1}^{x_i} \left(\frac{1}{h_{i-1}}\right)^2 (x - x_{i-1})^2 q(x) dx + \int_{x_i}^{x_{i+1}} \left(\frac{1}{h_i}\right)^2 (x_{i+1} - x)^2 q(x) dx,
\end{aligned}$$

para cada $i=1,2,\dots,n$;

$$\begin{aligned}
a_{i,i+1} &= \int_0^1 \{p(x)\Phi'_i(x)\Phi'_{i+1}(x) + q(x)\Phi_i(x)\Phi_{i+1}(x)\} dx \\
&= \int_x^{x_{i+1}} -\left(\frac{1}{h_i}\right)^2 p(x) dx + \int_{x_i}^{x_{i+1}} \left(\frac{1}{h_i}\right)^2 (x_{i+1} - x)(x - x_i) q(x) dx,
\end{aligned}$$

para cada $i=1,2,\dots,n-1$; y

$$\begin{aligned}
a_{i,i-1} &= \int_0^1 \{p(x)\Phi'_i(x)\Phi'_{i-1}(x) + q(x)\Phi_i(x)\Phi_{i-1}(x)\} dx \\
&= \int_{x-1}^{x_i} -\left(\frac{1}{h_{i-1}}\right)^2 p(x) dx + \int_{x_{i-1}}^{x_i} \left(\frac{1}{h_{i-1}}\right)^2 (x_i - x)(x - x_{i-1}) q(x) dx,
\end{aligned}$$

para cada $i=2,\dots,n$. Las componentes en b están dadas por

$$\begin{aligned}
b_i &= \int_0^1 f(x)\Phi_i(x) dx \\
&= \int_{x_{i-1}}^{x_i} \frac{1}{h_{i-1}} (x - x_{i-1}) f(x) dx + \int_{x_i}^{x_{i+1}} \frac{1}{h_i} (x_{i+1} - x) f(x) dx,
\end{aligned}$$

para cada $i=1,\dots,n$. Las componentes en c son los coeficientes desconocidos c_1, c_2, \dots, c_n de los cuales se construye la aproximación de Rayleigh-Ritz Φ dada por

5.2.1.- Ejemplos resueltos.

EJERCICIO 1.

Consideramos el siguiente problema de valor de frontera:

$$-y'' + \pi^2 \cdot y = 2 \cdot \pi^2 \cdot \text{sen}(\pi \cdot x)$$

Y sus condiciones iniciales son las siguientes :

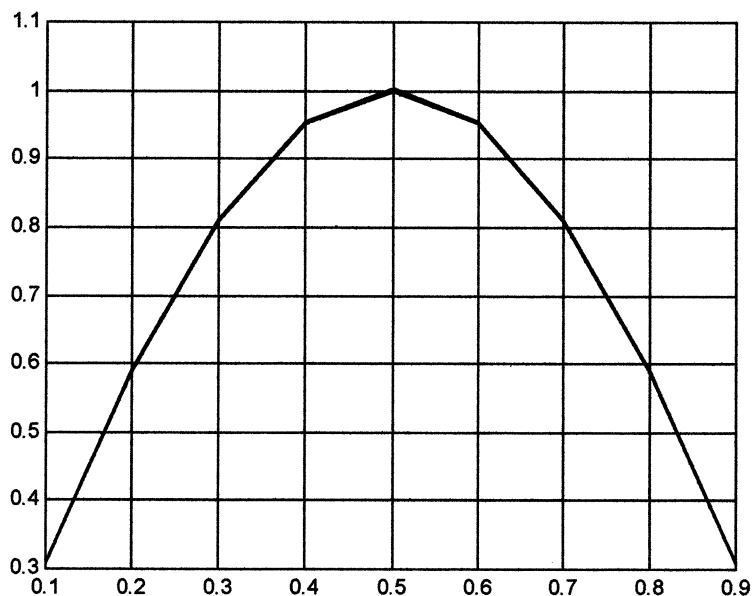
$$0 \leq x \leq 1, y(0) = y(1) = 0.$$

La función real es la siguiente: $y(x) = \text{sen}(\pi \cdot x)$

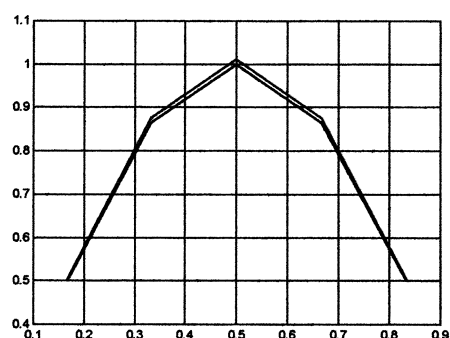
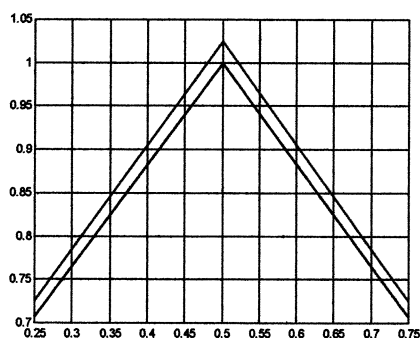
Las variables tomadas para nuestro algoritmo son las siguientes , $\text{tol}=0.001$ y $n=10$. A continuación podemos ver la tabla de valores obtenidos , así como su comparación con función real.

Valor de X(i)	Valor del método	Valor de f(x), función real
0.1	0.31028667561447	0.30901699437495
0.2	0.59020032952541	0.58778525229247
0.3	0.81234106301491	0.80901699437495
0.4	0.95496419334351	0.95105651629515
0.5	1.00410877480087	1.00000000000000
0.6	0.95496419334351	0.95105651629515
0.7	0.81234106301491	0.80901699437495
0.8	0.59020032952541	0.58778525229247
0.9	0.31028667561447	0.30901699437495

En esta primera aproximación a la función podemos ver como las gráficas obtenidas por el método y por la función original son casi idénticas. Como podemos ver apenas se nota diferencia en las siguientes calculadas con menos valores veremos como la diferencia es algo mayor. La línea roja representa el método de los elementos y la azul representa la función.



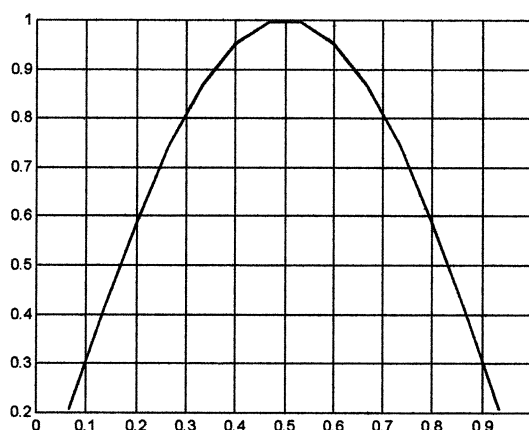
En estas dos gráficas podemos ver como es mayor la diferencia, la aproximación sigue siendo buena pero difiere de la original. En este caso hemos tomado el algoritmo para $n=4$ y $n=6$ respectivamente.



Por el contrario ahora vamos a aumentar n hasta 15 , y podremos ver como resulta muy difícil distinguir.

Valor de X(i)	Valor del método	Valor de f(i), función real
0.06666666666667	0.20829154981048	0.20791169081776
0.13333333333333	0.40747975940049	0.40673664307580
0.20000000000000	0.58885914819988	0.58778525229247
0.26666666666667	0.74450256656324	0.74314482547739
0.33333333333333	0.86760765024810	0.86602540378444
0.40000000000000	0.95279411637372	0.95105651629515
0.46666666666667	0.99633890760037	0.99452189536827
0.53333333333333	0.99633890760037	0.99452189536827
0.60000000000000	0.95279411637372	0.95105651629515
0.66666666666667	0.86760765024810	0.86602540378444
0.73333333333333	0.74450256656324	0.74314482547739
0.80000000000000	0.58885914819988	0.58778525229247
0.86666666666667	0.40747975940049	0.40673664307580
0.93333333333333	0.20829154981048	0.20791169081776

La gráfica queda de la siguiente manera , podemos observar mayor suavidad en los puntos de la gráfica:



EJERCICIO 2.

$$-x^2 \cdot y'' - 2 \cdot x \cdot y' + 2 \cdot y = -4 \cdot x^2$$

Consideramos el siguiente problema de valor de frontera:

Y sus condiciones iniciales son las siguientes :

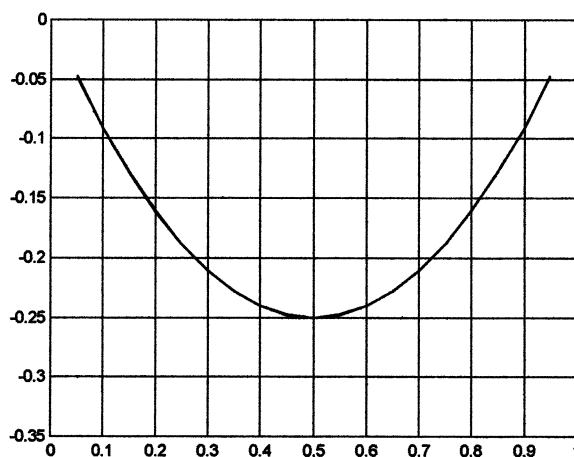
$$0 \leq x \leq 1, y(0) = y(1) = 0.$$

La función real es la siguiente: $y(x) = x^2 - x$

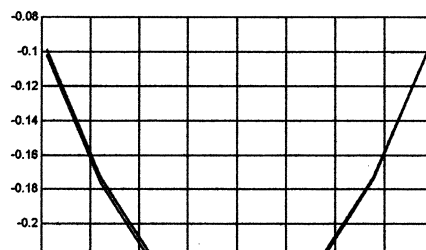
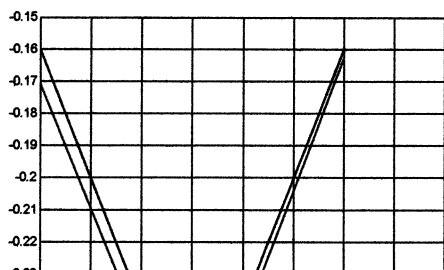
Las variables tomadas para nuestro algoritmo son las siguientes , $tol=0.001$ y $n=20$. A continuación podemos ver la tabla de valores obtenidos ,así como su comparación con función real.

Valor de X(i)	Valor del método	Valor de f(x),función real
0.05000000000000	-0.04829166666667	-0.04750000000000
0.10000000000000	-0.09075000000000	-0.09000000000000
0.15000000000000	-0.12820833333333	-0.12750000000000
0.20000000000000	-0.16066666666667	-0.16000000000000
0.25000000000000	-0.18812500000000	-0.18750000000000
0.30000000000000	-0.21058333333333	-0.21000000000000
0.35000000000000	-0.22804166666667	-0.22750000000000
0.40000000000000	-0.24050000000000	-0.24000000000000
0.45000000000000	-0.24795833333333	-0.24750000000000
0.50000000000000	-0.25041666666667	-0.25000000000000
0.55000000000000	-0.24787500000000	-0.24750000000000
0.60000000000000	-0.24033333333333	-0.24000000000000
0.65000000000000	-0.22779166666667	-0.22750000000000
0.70000000000000	-0.21025000000000	-0.21000000000000
0.75000000000000	-0.18770833333333	-0.18750000000000
0.80000000000000	-0.16016666666667	-0.16000000000000
0.85000000000000	-0.12762500000000	-0.12750000000000
0.90000000000000	-0.09008333333333	-0.09000000000000
0.95000000000000	-0.04754166666667	-0.04750000000000

Como podemos ver para $n=20$ las soluciones obtenidas son muy buenas solo difiere de la solución real a partir de la tercera cifra decimal y tenemos que tener en cuenta que la tolerancia utilizada para integrar es grande ya que su valor es de 0.001.



Ahora vamos a estudiar el comportamiento del algoritmo cuando reducimos n hasta llevarlo a un valor de 5 y a 9.

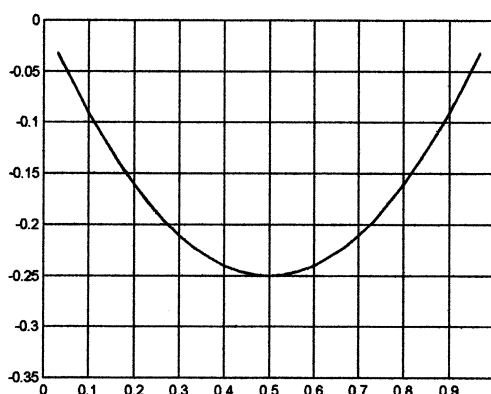


La diferencia de estas gráficas con la anterior es bastante considerable pero aun así los resultados siempre se acercan bastante a la solución real. Vamos a probar ahora a aumentar la tolerancia para la integral que como sabemos es uno de los cálculos importantes que realiza el algoritmo de los elementos finitos .Fijamos la tol en 0.0000000001.

Valor de X(i)	Valor del método con tol=0.001	Valor del método con tol=0.0000000001	Valor de f(x),función real
0.111111	-0.10242341106539	-0.10242341106539	-0.09876543209877
0.222222	-0.17604023776863	-0.17604023776863	-0.17283950617284
0.333333	-0.22496570644719	-0.22496570644719	-0.22222222222222
0.444444	-0.24919981710105	-0.24919981710105	-0.24691358024691
0.555555	-0.24874256973022	-0.24874256973022	-0.24691358024691
0.666666	-0.22359396433470	-0.22359396433470	-0.22222222222222
0.777777	-0.17375400091449	-0.17375400091449	-0.17283950617284
0.888888	-0.09922267946959	-0.09922267946959	-0.09876543209877

Por el contrario de lo que pensamos al principio la tolerancia no ha afectado al resultado final, si hubiéramos utilizado otro método para integrar en vez de quad8 ,creo que si notaríamos la diferencia ya que el quad8 es un método estándar que siempre se puede mejorar.

Si aumentamos n hasta 30 podemos observar la perfección del trazado de las líneas entre los diferentes puntos obtenidos ,es decir los valores obtenidos por uno u otro camino son idénticos .



EJERCICIO 3.

Consideramos el siguiente problema de valor de frontera:

$$-(x \bullet y') + x^2 \bullet y = x^4 - x^3 - 4 \bullet x + 1$$

Y sus condiciones iniciales son las siguientes :

$$0 \leq x \leq 1, y(0) = y(1) = 0.$$

$$y(x) = x^2 - x$$

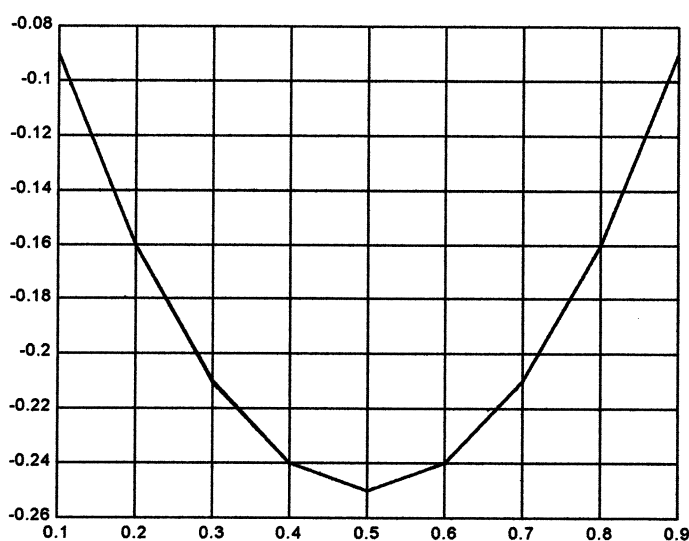
La función real es la siguiente:

Las variables tomadas para nuestro algoritmo son las siguientes , tol=0.001 y n=10. A continuación podemos ver la tabla de valores obtenidos ,así como su comparación con función real.

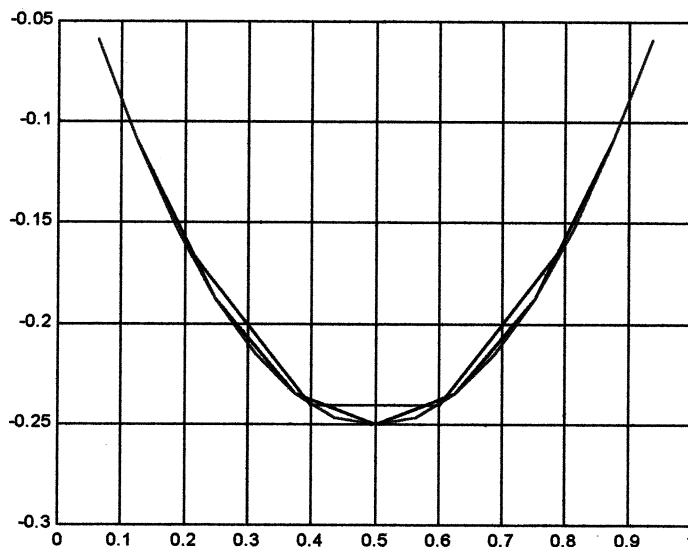
Valor de X(i)	Valor del método	Valor de f(x) real	Error cometido
0	0	0	0
0.1	-0.09008178092979	-0.09000000000000	0.00008178092979
0.2	-0.16010777543079	-0.16000000000000	0.07010777543079
0.3	-0.21012075322195	-0.21000000000000	0.12012075322195
0.4	-0.24012595753374	-0.24000000000000	0.15012595753374
0.5	-0.25012445434488	-0.25000000000000	0.16012445434488
0.6	-0.24011615087294	-0.24000000000000	0.15011615087294
0.7	-0.21010047853502	-0.21000000000000	0.12010047853502
0.8	-0.16007660606632	-0.16000000000000	0.07007660606632
0.9	-0.09004351162586	-0.09000000000000	0.00004351162586
1	0	0	0

La gráfica obtenida es la siguiente:

Como podemos observar en la tabla el error es mínimo ,por lo tanto como es prever en la gráfica no se distingue la real de la del método.



En esta gráfica podemos ver una vez más como trabaja este método desde el punto de la precisión numérica, en esta ocasión hemos tomado los siguientes valores de n; 5(rojo) ,8(azul) ,y 16(verde).



Estas comparaciones será mucho más interesantes hacerlas cuando realicemos estos mismos ejercicios con el método de las spline o cúbicas.

5.2.2 Código fuente del método con segmentarias lineales:

El fichero principal es el elemf.m, en el se hacen todas las operaciones principales y es donde metemos el valor de la n con la que queremos tratar.

%ALGORITMO DE LOS ELEMENTOS FINITOS CON FUNCIONES
SEGMENTARIAS LINEALES.

```
format long
clear;
nu=input('Escriba el valor de n :');
h=1/(nu);
n=nu-1;
tol=0.001;
for i=1:n
    q(1,i)=1/(h^2)*quad8('g',h*i,h*i+h,tol,0,i,1,h);
    q(2,i)=1/(h^2)*quad8('g',h*i-h,h*i,tol,0,i,2,h);
    q(3,i)=1/(h^2)*quad8('g',h*i,h*i+h,tol,0,i,3,h);
    q(4,i)=1/(h^2)*quad8('g',h*i-h,h*i,tol,0,i,4,h);
    q(5,i)=1/(h)*quad8('g',h*i-h,h*i,tol,0,i,5,h);
    q(6,i)=1/(h)*quad8('g',h*i,h*i+h,tol,0,i,6,h);
end
q(4,n+1)=1/h^2 *quad8('g', h*(n+1)-h, h*(n+1), tol, 0, n+1,
4, h );

for i=1:n
    a(i,i)=q(4,i)+q(4,i+1)+q(2,i)+q(3,i);
    b(i,1)=q(5,i)+q(6,i);
```

```

end
for i=1:n-1%diagonal superior
    a(i,i+1)=-q(4,i+1)+q(1,i);
end
for i=2:n %diagonal inferior
    a(i,i-1)=-q(4,i)+q(1,i-1);
end

c=a\b;
x=h:h:1-h;

y=c(1)*o(x,1,h);
for i=2:n
    y=y+c(i)*o(x,i,h);
end

j=1;
for k=h:h:1-h
    indice(j)=k;
    r(j)=freal(k);
    j=j+1;
end;
'valor de la x'
x'
'valores obtenidos por el método'
y'
'valores obtenidos por la función'
r'
'error cometido'
for l=1:j-1
    error(l)=abs(y(l)-r(l));
end;
plot(indice,y,'g');
grid;
hold on;
%plot(indice,r,'b');

```

A continuación tenemos una serie de ficheros auxiliares como son g.m ,freal.m los cuales son utilizados para las operaciones integración y para valores de la función real ,para posteriormente compararlos.

g.m

```

function y=g(x,i,j,h)
switch j
    case 1, y=(h*i+h - x).*(x-h*i).*q(x);
    case 2, y=((x - h*i+h).^2).*q(x);
    case 3, y=((h*i+h - x).^2).*q(x);
    case 4, y=p(x);
    case 5, y=(x - h*i+h).*f(x);
    case 6, y=(h*i+h - x).*f(x);
end

```

freal.m

```
function y=freal(x)
y=(x.^2)-x;
```

Por ultimo tenemos los ficheros donde metemos la funciones a estudiar ,de manera que tenemos que desglosarlas en las componentes p,q,f.

f.m

```
function y=f(x)
y=-4*(x.^2);
```

p.m

```
function y=p(x)
y=x;
```

q.m

```
function y=q(x)
y=x.^2;
```

5.3. Funciones segmentarias cúbicas.

Un trazador interpolante cúbico S en los cinco nodos x_0, x_1, x_2, x_3, x_4 . Se define una función f mediante :

- a) S es un polinomio cúbico, denotado por S_j en $[X_j, X_{j+1}]$ para $j= 1,2,3$.
- b) $S(X_j) = f(X_i)$ para $j=0,1,2,3,4$.
- c) $S_{j+1}(X_{j+1}) = S_j(X_{j+1})$. para $j=0,1,2$.
- d) $S'_{j+1}(X_{j+1}) = S'_j(X_{j+1})$. para $j=0,1,2$.
- e) $S''_{j+1}(X_{j+1}) = S''_j(X_{j+1})$. para $j=0,1,2$.
- f) Se satisface una de las siguientes condiciones de frontera .
 - i) Libre $S''(X_0) = S''(X_4) = 0$.
 - ii) Sujeta $S'(X_0) = f'(X_0)$ y $S'(X_4) = f'(X_4)$.

Como la unicidad de la solución requiere que el número de constantes en (a), 16 ,sea igual al número de condiciones en (b) hasta (f) ,solo puede especificarse una de las condiciones de frontera en (f) para los trazadores interpolantes cúbicos .

Las funciones de trazador cúbico que usaremos para nuestras funciones base se llaman trazadores -B o trazadores de forma campana. Estos trazadores difieren de los trazadores interpolantes en que se satisfacen ambos conjuntos de condiciones de frontera en (f) . Esto requiere de la relajación de dos de las condiciones de (b) hasta (e). Como el trazador debe tener dos derivadas continuas en $[X_0, X_4]$ la única posibilidad es la de modificar (b) a

$$b') S(X_j) = f(X_i) \text{ para } j=0,2,4.$$

El trazador -B básico S definido enseguida usa nodos igualmente espaciados

$X_0 = -2, X_1 = -1, X_2 = 0, X_3 = 1, X_4 = 2$. Satisface las condiciones interpolantes :

$$b') S(X_0) = 0, S(X_2) = 1, S(X_4) = 0.$$

así como ambos conjuntos de condiciones.

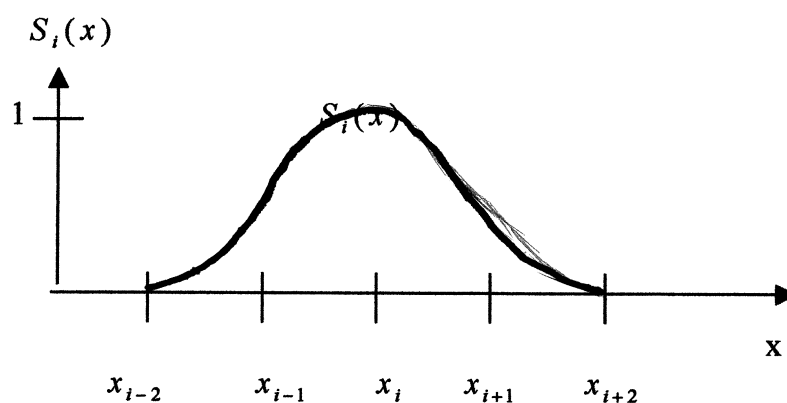
$$i) S''(X_0) = S''(X_4) = 0.$$

$$ii) S'(X_0) = S'(X_4) = 0.$$

Como consecuencia , $S \in C_0^2(-\infty, \infty)$.

$$S(x) = \begin{cases} 0 \rightarrow x \leq -2 \\ \frac{1}{4}[(2-x)^3 - 4(1-x)^3 - 6 \cdot x^3 + 4(1+x)^3] \rightarrow -2 < x \leq -1 \\ \frac{1}{4}[(2-x)^3 - 4(1-x)^3 - 6 \cdot x^3] \rightarrow -1 < x \leq 0 \\ \frac{1}{4}[(2-x)^3 - 4(1-x)^3] \rightarrow 0 < x \leq 1 \end{cases}$$

Para construir las funciones base $\phi_i \rightarrow C_0^2[0,1]$ dividimos primero $[0,1]$ escogiendo un entero positivo n y definiendo $h=1/(n+1)$. Esto produce los nodos igualmente espaciados $X_i = i \cdot h$ para cada $i=0,1,\dots,n+1$. Definimos entonces S_i mediante $S_i(x) = S((x-x_i)/h)$ para cada $i=0,1,\dots,n+1$. La gráfica de un S_i típico se muestra en la siguiente figura.



Es fácil demostrar que $\{ S_i \}_{i=0}^{n+1}$ es un conjunto linealmente independiente de trazadores. Para que el conjunto $\{ S_i \}_{i=0}^{n+1}$ satisfaga las condiciones de frontera $\phi_i(0) = \phi_i(1) = 0$, es necesario modificar S_0, S_1, S_n, S_{n+1} la base con esta modificación queda de la siguiente forma.

$$S(x) = \begin{cases} S_0(x) - 4S((x+h)/h). \rightarrow i = 0. \\ S_1(x) - S((x+h)/h). \rightarrow i = 1. \\ S_i(x). \rightarrow 2 \leq i \leq n-1. \\ S_n(x) - S((x-(n+2)h)/h). \rightarrow i = n. \\ S_{n+1}(x) - 4S((x-(n+2)h)/h) \rightarrow i = n+1. \end{cases}$$

La matriz en la aproximación de rayleigh-ritz será una matriz banda con espesor de banda de a lo más siete.

$$A = \begin{bmatrix} a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} & 0 \\ \cdot & \cdot & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \cdot & \cdot & \cdot & a_{n,n+1} \\ 0 & a_{n+1,n-2} & a_{n+1,n-1} & a_{n+1,n} & a_{n+1,n+1} \end{bmatrix}$$

Donde:

$$a_{i,j} = \int_0^1 \{p(x) \cdot \phi_i'(x) \cdot \phi_j'(x) + q(x) \cdot \phi(x) \cdot \phi'(x)\} dx$$

Para cada $i, j = 0, 1, \dots, n+1$. La matriz A es positiva definida, así que el sistema lineal, puede resolverse fácilmente con el algoritmo de choleski o con eliminación gaussiana.

5.3.1 Ejemplos resueltos del método con segmentarias cúbicas:

EJERCICIO 1.

Consideramos el siguiente problema de valor de frontera:

$$-y'' + \pi^2 \cdot y = 2 \cdot \pi^2 \cdot \text{sen}(\pi \cdot x)$$

Y sus condiciones iniciales son las siguientes :

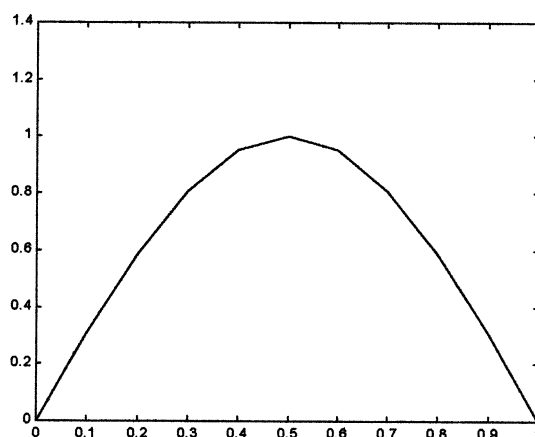
$$0 \leq x \leq 1, y(0) = y(1) = 0.$$

La función real es la siguiente: $y(x) = \text{sen}(\pi \cdot x)$

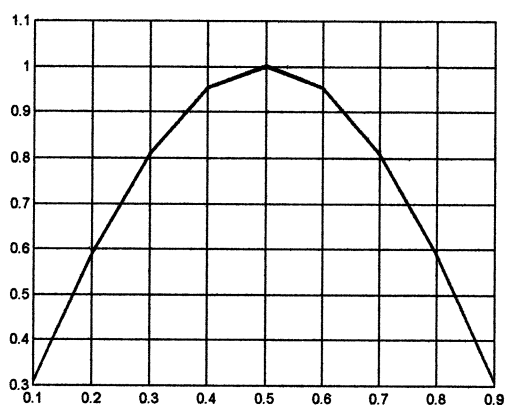
Las variables tomadas para nuestro algoritmo son las siguientes, $\text{tol}=0.001$ y $n=10$. A continuación podemos ver la tabla de valores obtenidos, así como su comparación con función real.

Valor de X(i)	Valor con cúbica	Valor con lineal	Valor de f(x) real
0	0	0	0
0.1	0.30902161223200	0.31028667561447	0.30901699437495
0.2	0.58779414435911	0.59020032952541	0.58778525229247
0.3	0.80902855373012	0.81234106301491	0.80901699437495
0.4	0.95107011624145	0.95496419334351	0.95105651629515
0.5	1.00001418110165	1.00410877480087	1.00000000000000
0.6	0.95107011624145	0.95496419334351	0.95105651629515
0.7	0.80902855373013	0.81234106301491	0.80901699437495
0.8	0.58779414435911	0.59020032952541	0.58778525229247
0.9	0.30902161223200	0.31028667561447	0.30901699437495
1	0.00000000000000	0	0

En esta primera aproximación a la función podemos ver como las gráficas obtenidas por el método y por la función original son idénticas. Como podemos ver apenas se nota diferencia en las siguientes calculadas con menos valores veremos como la diferencia es algo mayor. La línea roja representa el método de los elementos y la azul representa la función.



Esta gráfica representa los valores obtenidos por el método de los elementos finitos pero usando trazadores lineales, como podemos observar la gráfica anterior tiene mas suavidad en el trazado de los puntos que esta ,ya que los trazadores cúbicos son más precisos.



En estas tablas podemos observar el error cometido por cada método en cuestión con respecto a la función original.

Valor de X(i)	Valor con cúbica	Error con cúbica	Valor de f(x) real
0	0	0	0
0.1	0.30902161223200	0.00000461785704	0.30901699437495
0.2	0.58779414435911	0.00000889206663	0.58778525229247
0.3	0.80902855373012	0.00001155935517	0.80901699437495
0.4	0.95107011624145	0.00001359994629	0.95105651629515
0.5	1.00001418110165	0.00001418110164	1.00000000000000
0.6	0.95107011624145	0.00001359994629	0.95105651629515
0.7	0.80902855373013	0.00001155935517	0.80901699437495
0.8	0.58779414435911	0.00000889206663	0.58778525229247
0.9	0.30902161223200	0.00004617857049	0.30901699437495
1	0	0	0

Ahora representamos los valores obtenidos por los trazadores lineales.

Valor de X(i)	Valor con lineal	Error con lineal	Valor de f(x) real
0	0	0	0
0.1	0.31028667561447	0.00126968123953	0.30901699437495
0.2	0.59020032952541	0.00241507723294	0.58778525229247
0.3	0.81234106301491	0.00332406863996	0.80901699437495
0.4	0.95496419334351	0.00390767704835	0.95105651629515
0.5	1.00410877480087	0.00410877480087	1.00000000000000
0.6	0.95496419334351	0.00390767704835	0.95105651629515
0.7	0.81234106301491	0.00332406863996	0.80901699437495
0.8	0.59020032952541	0.00241507723294	0.58778525229247
0.9	0.31028667561447	0.00126968123953	0.30901699437495
1	0	0	0

Como podemos observar lo que ante comentábamos es cierto, es decir el error cometido por los trazadores lineales es mayor que el error cometido por los trazadores cúbicos, en cuestión sobre un 10 elevado -2 respecto uno de otro.

EJERCICIO 2.

$$-x^2 \cdot y'' - 2 \cdot x \cdot y' + 2 \cdot y = -4 \cdot x^2$$

Consideramos el siguiente problema de valor de frontera:

Y sus condiciones iniciales son las siguientes :

$$0 \leq x \leq 1, y(0) = y(1) = 0.$$

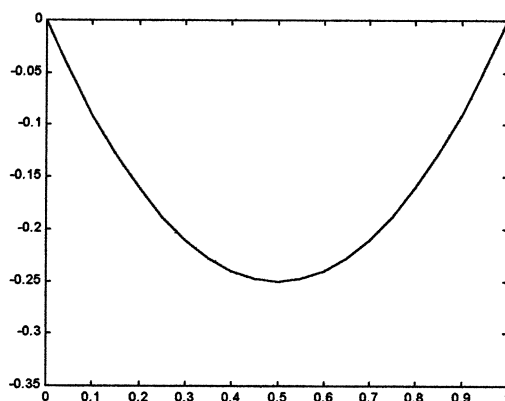
La función real es la siguiente: $y(x) = x^2 - x$

Las variables tomadas para nuestro algoritmo son las siguientes , $tol=0.001$ y $n=20$. A continuación podemos ver la tabla de valores obtenidos ,así como su comparación con función real.

Valor de X(i)	Valor del método	Valor de f(x) real	Error cometido
0.050000000000000	-0.04750018473849	-0.047500000000000	0.00000018473849
0.100000000000000	-0.09000005640023	-0.090000000000000	0.00000005640023
0.150000000000000	-0.12750004345291	-0.127500000000000	0.00000004345291
0.200000000000000	-0.16000003947603	-0.160000000000000	0.00000003947603
0.250000000000000	-0.18750004439053	-0.187500000000000	0.00000004439053
0.300000000000000	-0.21000004911064	-0.210000000000000	0.00000004911064
0.350000000000000	-0.22750005548291	-0.227500000000000	0.00000005548291
0.400000000000000	-0.24000006196522	-0.240000000000000	0.00000006196522
0.450000000000000	-0.24750006887210	-0.247500000000000	0.00000006887210
0.500000000000000	-0.25000007581795	-0.250000000000000	0.00000007581795
0.550000000000000	-0.24750008304083	-0.247500000000000	0.00000008304083
0.600000000000000	-0.24000009001432	-0.240000000000000	0.00000009001432
0.650000000000000	-0.22750009770479	-0.227500000000000	0.00000009770479
0.700000000000000	-0.21000010402285	-0.210000000000000	0.00000010402285
0.750000000000000	-0.18750011338547	-0.187500000000000	0.00000011338547
0.800000000000000	-0.16000011627107	-0.160000000000000	0.00000011627107
0.850000000000000	-0.12750013325074	-0.127500000000000	0.00000013325074
0.900000000000000	-0.09000011946043	-0.090000000000000	0.00000011946043
0.950000000000000	-0.04750017645829	-0.047500000000000	0.00000017645829

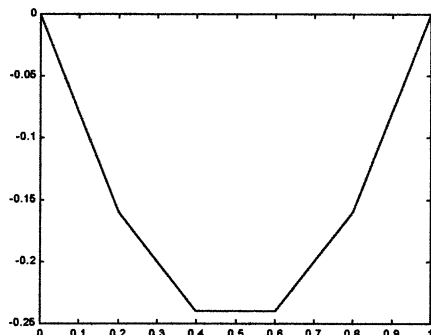
Como podemos ver para $n=20$ las soluciones obtenidas son muy buenas solo difiere de la solución real a partir de la sexta cifra decimal, mientras que con el método de los trazadores lineales se diferenciaban a partir de la tercera cifra decimal; tenemos que tener en cuenta que la tolerancia utilizada para integrar es grande ya que su valor es de 0.001.

La gráfica obtenida es la siguiente :

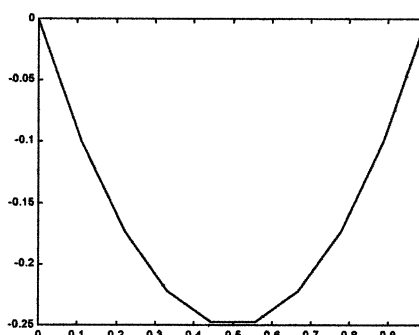


En las siguientes gráficas podremos observar la efectividad de nuestro algoritmo con valores de n más pequeños y por lo tanto gráficas y valores menos precisos.

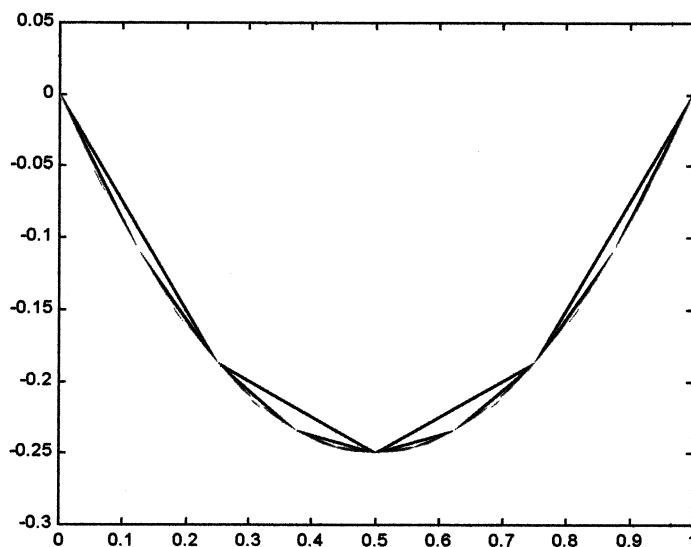
Para $n=5$.



para $n=9$.



Ahora vamos a ver la progresión que tiene la gráfica para distintos valores de n .



EJERCICIO 3.

Consideramos el siguiente problema de valor de frontera:

$$-(x \cdot y') + x^2 \cdot y = x^4 - x^3 - 4 \cdot x + 1$$

Y sus condiciones iniciales son las siguientes :

$$0 \leq x \leq 1, y(0) = y(1) = 0.$$

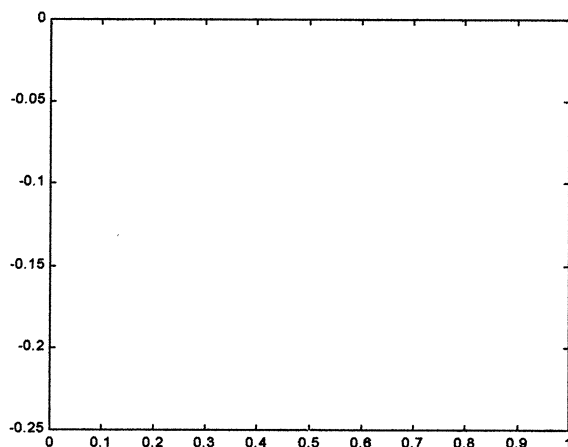
La función real es la siguiente: $y(x) = x^2 - x$

Las variables tomadas para nuestro algoritmo son las siguientes, $\text{tol}=0.001$ y $n=10$. A continuación podemos ver la tabla de valores obtenidos, así como su comparación con función real.

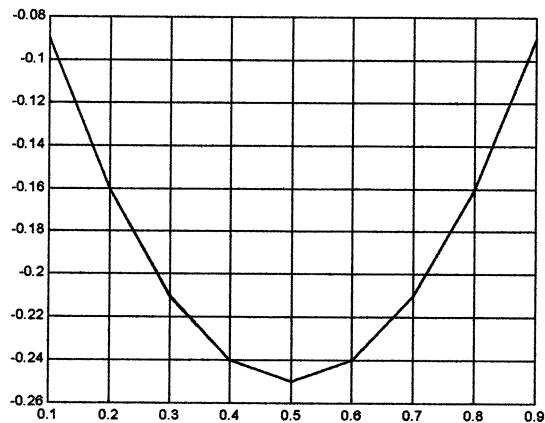
Valor de X(i)	Valor del método	Valor de f(x) real	Error cometido
0	0	0	0
0.1	-0.08998925212364	-0.09000000000000	0.00001074787636
0.2	-0.15999144724183	-0.16000000000000	0.00000855275817
0.3	-0.20999377575470	-0.21000000000000	0.00000622424530
0.4	-0.23999543746071	-0.24000000000000	0.00000456253929
0.5	-0.24999668590784	-0.25000000000000	0.00000331409216
0.6	-0.23999768617210	-0.24000000000000	0.00000231382790
0.7	-0.20999855503485	-0.21000000000000	0.00000144496515
0.8	-0.15999921001904	-0.16000000000000	0.00000078998096
0.9	-0.08999998791739	-0.09000000000000	0.0000001208261
1	0	0	0

Por lo que estamos viendo a medida que estamos haciendo los distintos ejercicios podemos concretar que el error para trazadores cúbicos oscila entre 10^{-4} y 10^{-6} , mientras que el error para trazadores lineales es de 10^{-3} , por lo tanto bastante mayor.

La gráfica obtenida es la siguiente:



Esta es la gráfica obtenida por la misma función pero para trazadores lineales.



5.3.2 Código fuente del método con segmentarias cúbicas.

El fichero principal es el ef.m, en el se hacen todas las operaciones principales y es donde metemos el valor de la n con la que queremos tratar.

```
clear;
format long;
ni=input('Escriba el valor de n=');
n=ni-1;
h=1/(n+1);
for i=1:n+2
    x(i)=(i-1)*h;
end

for i=0:n+1
    for j=i:min(i+3,n+1)
        if j-2<0
            li=0;
        else
            li=x(j-2+1);
        end

        if i+2>n+1
            ls=1;
        else
            ls=x(i+2+1);
        end

        a(i+1,j+1)=quad8('ga',li,ls,.001,[],i,j,h);
        if (i~=j)
            a(j+1,i+1)=a(i+1,j+1);
        end
    end
end

if (i>=4)
    for j=1:i-4
        a(i+1,j+1)=0;
    end
end
```

```

end
if i<=n-3
    for j=i+4:n+1
        a(i+1,j+1)=0;
    end
end
end
%xxxxxxxxxxxxxxxxx
    if i-2<0
        li=0;
    else
        li=x(i-2+1);
    end

    if i+2>n+1
        ls=1;
    else
        ls=x(i+2+1);
    end

    b(i+1)=quad8('gb',li,ls,0.001,0,i,h);
end
b=b';
c=a\b;
x=0:h:1;
y=c(1).*o(x,0,h);
for i=1:ni
    y=y+c(i+1).*o(x,i,h);
end
'valor de x          valor aprox. de y
valor real.          error'
M=[x',y',(feval('freal',x))',(abs(y-(feval('freal',x))))'];
M
plot(x,y,'b');
hold on
plot(x,feval('freal',x),'r');

```

A continuación tenemos una serie de ficheros auxiliares como son `ga.m`, `gb` y `freal.m` los cuales son utilizados para las operaciones integración y para valores de la función real, para posteriormente compararlos.

ga.m

```

function y=ga(x,i,j,h)
y=p(x).*op(x,i,h).*op(x,j,h) + q(x).*o(x,i,h).*o(x,j,h);

```

gb.m

```

function y=gb(x,i,h)
y=f(x).*o(x,i,h);

```

freal.m

```

function y=freal(x)
y=(x.^2)-x;

```

Seguidamente utilizaremos varios ficheros para definir los trazadores cúbicos ,así como sus derivadas .Para ello tenemos los ficheros o.m,op.m,s.m y sp.m.

o.m

```
function y=o(x,i,h)
n=1/h - 1;
%y=sin((i+1)*pi*x)
if (i==0)
    y=s(x/h) - 4*s((x+h)/h);
elseif (i==1)
    y=s((x-h)/h) - s((x+h)/h);
elseif ((1<i) & (i<n))
    y=s((x-i*h)/h);
elseif (i==n)
    y=s((x-n*h)/h) - s((x-(n+2)*h)/h);
elseif (i==n+1)
    y=s((x-(n+1)*h)/h) - 4*s((x-(n+2)*h)/h);
else
    error('error en o');
    y=i;
end
```

op.m

```
function y=op(x,i,h)
n=1/h - 1;

if (i==0)
    y=1/h*(sp(x/h) - 4*sp((x+h)/h));
elseif (i==1)
    y=1/h*(sp((x-h)/h) - sp((x+h)/h));
elseif ((1<i) & (i<n))
    y=1/h*sp((x-i*h)/h);
elseif (i==n)
    y=1/h*(sp((x-n*h)/h) - sp((x-(n+2)*h)/h));
elseif (i==n+1)
    y=1/h*(sp((x-(n+1)*h)/h) - 4*sp((x-(n+2)*h)/h));
else
    error('error en op');
end
```

s.m

```
function y=s(x)
for i=1:length(x)
    %y(i)=sin(pi*x(i));

    if (x(i)<=-2)
        y(i)=0;

    elseif ((-2<x(i)) & (x(i)<=-1))
        y(i)=1/4*(2+x(i))^3;
```

```

elseif ((-1<x(i)) & (x(i)<=0))
    y(i)=1/4*((2+x(i))^3 - 4*(1+x(i))^3);

elseif ((0<x(i)) & (x(i)<=1))
    y(i)=1/4*((2-x(i))^3 - 4*(1-x(i))^3);

elseif ((1<x(i)) & (x(i)<=2))
    y(i)=1/4*(2-x(i))^3;

elseif (2<x(i))
    y(i)=0;
else
    error('error en s');
end
end

```

sp.m

```

function y=sp(x)
for i=1:length(x)
    if x(i)<=-2
        y(i)=0;

    elseif ((-2<x(i)) & (x(i)<=-1))
        y(i)=3/4*(2+x(i))^2;

    elseif ((-1<x(i)) & (x(i)<=0))
        y(i)=3/4*((2+x(i))^2 - 4*(1+x(i))^2);

    elseif ((0<x(i)) & (x(i)<=1))
        y(i)=3/4*(-(2-x(i))^2 + 4*(1-x(i))^2);

    elseif ((1<x(i)) & (x(i)<=2))
        y(i)=-3/4*(2-x(i))^2;

    elseif (2<x(i))
        y(i)=0;
    else
        error('error en sp');
    end
end
end

```

Por ultimo tenemos los ficheros donde metemos la funciones a estudiar ,de manera que tenemos que desglosarlas en las componentes p,q,f.

f.m

```

function y=f(x)
y=-4*(x.^2);

```

p.m

```

function y=p(x)

```

**ECUACIONES EN
DERIVADAS
PARCIALES**

$$y=x;$$

q.m

function $y=q(x)$

$$y=x.^2;$$

6.-ECUACIONES EN DERIVADAS PARCIALES ELÍPTICAS.

6.1. Conceptos teóricos.

La ecuación diferencial parcial elíptica que consideraremos es la ecuación de Poisson:

$$\nabla^2 u(x, y) \equiv \frac{\partial^2 u}{\partial x^2}(x, y) + \frac{\partial^2 u}{\partial y^2}(x, y) = f(x, y)$$

para $(x, y) \in R$ y

$$u(x, y) = g(x, y) \text{ para } (x, y) \in S,$$

donde

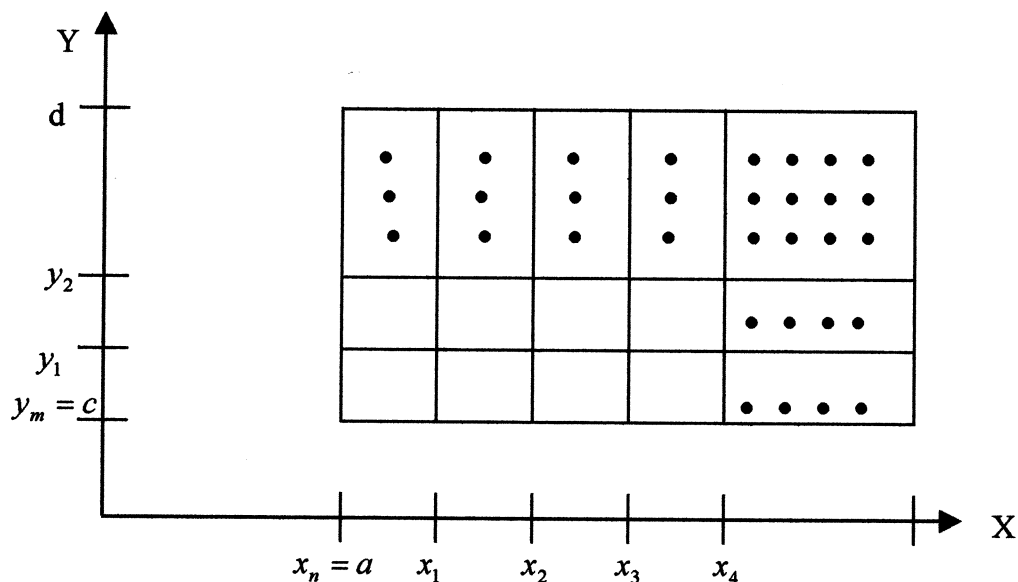
$$R = \{(x, y) \mid a < x < b, c < y < d\},$$

y S denota la frontera de R . Para esta discusión supondremos que ambas f y g son continuas en sus dominios con lo que se asegura una solución única de la ecuación.

El método que usaremos es una adaptación del método de diferencia finita para problemas de valor de frontera. El primer paso consiste en escoger dos enteros n y m , u en definir los tamaños de paso h y k por $h=(b-a)/n$ y $k=(d-c)/m$. Partiendo el intervalo $[a, b]$ en n partes iguales de longitud h y el intervalo $[c, d]$ en m partes iguales de longitud k da una manera de asociar una red al rectángulo R pasando rectas verticales y horizontales por unos puntos con coordenadas (x_i, y_j) donde

$$x_i = a + ih \text{ para cada } i=0, 1, \dots, n,$$

$$y_j = c + jk \text{ para cada } j=0, 1, \dots, m.$$



Las rectas $x = x_i$ e $y = y_j$ se llaman líneas de red y sus intersecciones se llaman puntos de red. Por cada punto de red en el interior de la malla (x_i, y_j) , $i = 1, 2, \dots, n-1$ y $j = 1, 2, \dots, m-1$, usamos la serie de Taylor en la variables x alrededor de x_i para generar la fórmula de diferencia centrada

$$\frac{\partial^2 u}{\partial x^2}(x_i, y_j) = \frac{u(x_{i+1}, y_j) - 2u(x_i, y_j) + u(x_{i-1}, y_j))}{h^2} - \frac{h^2}{12} \cdot \frac{\partial^4 u}{\partial x^4}(\xi_i, y_j),$$

donde $\xi_i \in (x_{i-1}, x_{i+1})$; y la serie de Taylor en la variables y alrededor de y_j para generar la fórmula de diferencia centrada

$$\frac{\partial^2 u}{\partial y^2}(x_i, y_j) = \frac{u(x_i, y_{j+1}) - 2u(x_i, y_j) + u(x_i, y_{j-1}))}{k^2} - \frac{k^2}{12} \cdot \frac{\partial^4 u}{\partial y^4}(x_i, \eta_j),$$

donde $\eta_j \in (y_{j-1}, y_{j+1})$.

Usando estas fórmulas podemos expresar la ecuación de Poisson en los puntos (x_i, y_j) como:

$$\frac{u(x_{i+1}, y_j) - 2u(x_i, y_j) + u(x_{i-1}, y_j))}{h^2} + \frac{u(x_i, y_{j+1}) - 2u(x_i, y_j) + u(x_i, y_{j-1}))}{k^2} =$$

$$f(x_i, y_j) + \frac{h^2}{12} \cdot \frac{\partial^4 u}{\partial x^4}(\xi_i, y_j) + \frac{k^2}{12} \cdot \frac{\partial^4 u}{\partial y^4}(x_i, \eta_j).$$

Para cada $i = 1, 2, \dots, (n-1)$ y $j = 1, 2, \dots, (m-1)$, y las condiciones de frontera como

$$\begin{aligned} u(x_0, y_j) &= g(x_0, y_j) \quad \text{para cada } j = 0, 1, \dots, m \\ u(x_n, y_j) &= g(x_n, y_j) \quad \text{para cada } j = 0, 1, \dots, m \\ u(x_i, y_0) &= g(x_i, y_0) \quad \text{para cada } i = 1, 2, \dots, n-1 \\ u(x_i, y_m) &= g(x_i, y_m) \quad \text{para cada } i = 1, 2, \dots, n-1. \end{aligned}$$

En la forma ecuación de diferencia, esto da lugar a un método llamado de diferencia centrada, con error de truncamiento local de orden $O(h^2 + k^2)$, que puede escribirse como:

$$2 \left[\left(\frac{h}{k} \right)^2 + 1 \right] w_{i,j} - (w_{i+1,j} + w_{i-1,j}) - \left(\frac{h}{k} \right)^2 (w_{i,j+1} + w_{i,j-1}) = h^2 f(x_i, y_j),$$

para cada $i = 1, 2, \dots, n-1$ y $j = 1, 2, \dots, m-1$, y

$$\begin{aligned}
 w_{0,j} &= g(x_0, y_j) \text{ para cada } j = 0, 1, \dots, m, \\
 w_{n,j} &= g(x_n, y_j) \text{ para cada } j = 0, 1, \dots, m, \\
 w_{i,0} &= g(x_i, y_0) \text{ para cada } i = 1, 2, \dots, n-1, \\
 w_{i,j} &= g(x_i, y_m) \text{ para cada } i = 1, 2, \dots, n-1,
 \end{aligned}$$

donde $w_{i,j}$ aproxima a $u(x_n, y_m)$.

La ecuación típica involucra aproximaciones en una región de forma de estrella alrededor de (x_i, y_j) .

Si usamos la información de las condiciones de frontera donde sea apropiado en el sistema dado anteriormente, tendremos un sistema de $(n-1)(m-1)$ ecuaciones lineales con $(n-1)(m-1)$ incógnitas, siendo las incógnitas las aproximaciones $w_{i,j}$ de $u(x_i, y_j)$ para los puntos interiores de la red.

El sistemas lineal que contiene a estas incógnitas puede expresarse más eficientemente para los cálculos de la matriz si se introduce un renombramiento de los puntos interiores de la red. Una manera de etiquetar estos puntos es :

$$P_l = (x_i, y_j) \quad y \quad w_l = w_{i,j},$$

donde $l = i + (m-1-j)(n-1)$ para cada $i = 1, 2, \dots, n-1$ y $j = 1, 2, \dots, m-1$.

Así reordenamos los puntos de red consecutivamente de izquierda a derecha y de arriba hacia abajo. Marcando los puntos de esta manera asegura que el sistema necesario para determinar $w_{i,j}$ será una matriz banda con espesor de la banda de a lo más $2n-1$.

6.2.- Aplicación del método.

6.2.1 Usar el algoritmo de Poisson para aproximar la solución de la siguiente ecuación diferencial elíptica:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0, \quad 0 < x < 1, \quad 0 < y < 1;$$

$$u(x, 0) = 0, u(x, 1) = x, 0 \leq x \leq 1;$$

$$u(0, y) = 0, u(1, y) = y, 0 \leq y \leq 1.$$

Usando $h = k = 0.1$, con lo obtendremos n y $m=10$, usaremos una tolerancia de 10^{-9} y un número máximo de 200 iteraciones.

Los resultados obtenidos, se encuentran representados en la siguiente tabla, donde podremos comparar con algunos de los valores conocidos de la función real, por lo que sólo pondremos los valores cuya solución real conozcamos :

x_i	y_j	$w_{i,j}$	REAL	ERROR
0.3	0.3	0.08999999420426	0.089941	$5.89 \cdot 10^{-5}$
0.3	0.7	0.20999999291592	0.209928	$7.19 \cdot 10^{-5}$
0.7	0.3	0.20999999525830	0.209951	$4.89 \cdot 10^{-5}$
0.7	0.7	0.48999999420426	0.489941	$5.89 \cdot 10^{-5}$

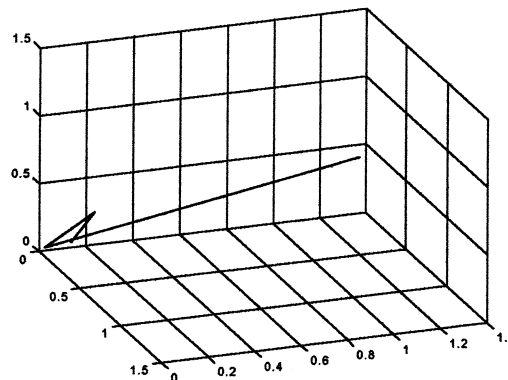
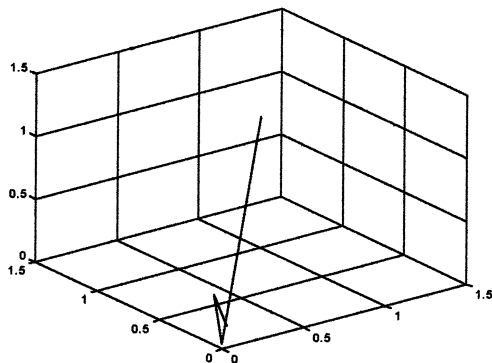
Como vemos el error se puede considerar pequeño, aunque si ponemos más tolerancia a nuestro algoritmo podremos comprobar que este error se incrementará.

A continuación mostramos todos los valores obtenidos y su representación gráfica:

x_i	y_j	$w_{i,j}$
0.10000000000000	0.10000000000000	0.00999999915441
0.10000000000000	0.20000000000000	0.01999999830883
0.10000000000000	0.30000000000000	0.02999999755251
0.10000000000000	0.40000000000000	0.03999999697474
0.10000000000000	0.50000000000000	0.04999999665535
0.10000000000000	0.60000000000000	0.05999999665535
0.10000000000000	0.70000000000000	0.06999999700846
0.10000000000000	0.80000000000000	0.07999999771466
0.10000000000000	0.90000000000000	0.08999999873670
0.20000000000000	0.10000000000000	0.01999999847032
0.20000000000000	0.20000000000000	0.03999999694064
0.20000000000000	0.30000000000000	0.05999999557245
0.20000000000000	0.40000000000000	0.07999999452725
0.20000000000000	0.50000000000000	0.09999999394948
0.20000000000000	0.60000000000000	0.11999999394948
0.20000000000000	0.70000000000000	0.13999999458825
0.20000000000000	0.80000000000000	0.15999999586579
0.20000000000000	0.90000000000000	0.17999999771466
0.30000000000000	0.10000000000000	0.02999999799762
0.30000000000000	0.20000000000000	0.05999999599525
0.30000000000000	0.30000000000000	0.08999999420426
0.30000000000000	0.40000000000000	0.11999999283608
0.30000000000000	0.50000000000000	0.14999999207976
0.30000000000000	0.60000000000000	0.17999999207976
0.30000000000000	0.70000000000000	0.20999999291592
0.30000000000000	0.80000000000000	0.23999999458825
0.30000000000000	0.90000000000000	0.26999999700846
0.40000000000000	0.10000000000000	0.03999999776127
0.40000000000000	0.20000000000000	0.07999999552255
0.40000000000000	0.30000000000000	0.11999999352017
0.40000000000000	0.40000000000000	0.15999999199049
0.40000000000000	0.50000000000000	0.19999999114490
0.40000000000000	0.60000000000000	0.23999999114490
0.40000000000000	0.70000000000000	0.27999999207976
0.40000000000000	0.80000000000000	0.31999999394948

0.40000000000000	0.90000000000000	0.35999999665535
0.50000000000000	0.10000000000000	0.04999999776127
0.50000000000000	0.20000000000000	0.09999999552255
0.50000000000000	0.30000000000000	0.14999999352017
0.50000000000000	0.40000000000000	0.19999999199049
0.50000000000000	0.50000000000000	0.24999999114490
0.50000000000000	0.60000000000000	0.29999999114490
0.50000000000000	0.70000000000000	0.34999999207976
0.50000000000000	0.80000000000000	0.39999999394948
0.50000000000000	0.90000000000000	0.44999999665535
0.60000000000000	0.10000000000000	0.05999999797505
0.60000000000000	0.20000000000000	0.11999999595011
0.60000000000000	0.30000000000000	0.17999999413894
0.60000000000000	0.40000000000000	0.23999999275533
0.60000000000000	0.50000000000000	0.29999999199049
0.60000000000000	0.60000000000000	0.35999999199049
0.60000000000000	0.70000000000000	0.41999999283608
0.60000000000000	0.80000000000000	0.47999999452725
0.60000000000000	0.90000000000000	0.53999999697474
0.70000000000000	0.10000000000000	0.06999999836178
0.70000000000000	0.20000000000000	0.13999999672357
0.70000000000000	0.30000000000000	0.20999999525830
0.70000000000000	0.40000000000000	0.27999999413894
0.70000000000000	0.50000000000000	0.34999999352017
0.70000000000000	0.60000000000000	0.41999999352017
0.70000000000000	0.70000000000000	0.48999999420426
0.70000000000000	0.80000000000000	0.55999999557245
0.70000000000000	0.90000000000000	0.62999999755251
0.80000000000000	0.10000000000000	0.07999999886802
0.80000000000000	0.20000000000000	0.15999999773604
0.80000000000000	0.30000000000000	0.23999999672357
0.80000000000000	0.40000000000000	0.31999999595011
0.80000000000000	0.50000000000000	0.39999999552255
0.80000000000000	0.60000000000000	0.47999999552255
0.80000000000000	0.70000000000000	0.55999999599525
0.80000000000000	0.80000000000000	0.63999999694064
0.80000000000000	0.90000000000000	0.71999999830883
0.90000000000000	0.10000000000000	0.08999999943401
0.90000000000000	0.20000000000000	0.17999999886802
0.90000000000000	0.30000000000000	0.26999999836178
0.90000000000000	0.40000000000000	0.35999999797505
0.90000000000000	0.50000000000000	0.44999999776127
0.90000000000000	0.60000000000000	0.53999999776127
0.90000000000000	0.70000000000000	0.62999999799762
0.90000000000000	0.80000000000000	0.71999999847032
0.90000000000000	0.90000000000000	0.80999999915441

La gráfica que obtenemos al representar los valores son :



6.2.2 Aproximar la solución mediante el algoritmo de Poisson, de la siguiente ecuación diferencial parcial elíptica:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = (x^2 + y^2)e^{2y}, \quad 0 < x < 2, \quad 0 < y < 1;$$

$$u(x,0) = 1, u(x,1) = e^x, 0 \leq x \leq 2;$$

$$u(0,y) = 1, u(2,y) = e^{2y}, 0 \leq y \leq 1.$$

Usando $h = 0.2$ y $k = 0.1$, con lo obtendremos n y $m=10$, usaremos una tolerancia de 10^{-9} y un número máximo de 200 iteraciones.

Compararemos los resultados con la función real dada:

$$u(x,y) = e^{xy}$$

Obtenemos los valores dados por el algoritmos y comparamos con los obtenidos de la función real:

x_i	y_j	$w_{i,j}$	REAL
0.2000000000000000	0.1000000000000000	1.02021405253680	1.02020134002676
0.2000000000000000	0.2000000000000000	1.04083571604545	1.04081077419239
0.2000000000000000	0.3000000000000000	1.06187285312401	1.06183654654536
0.2000000000000000	0.4000000000000000	1.08333353949260	1.08328706767496
0.2000000000000000	0.5000000000000000	1.10522592751711	1.10517091807565
0.2000000000000000	0.6000000000000000	1.12755798582649	1.12749685157938

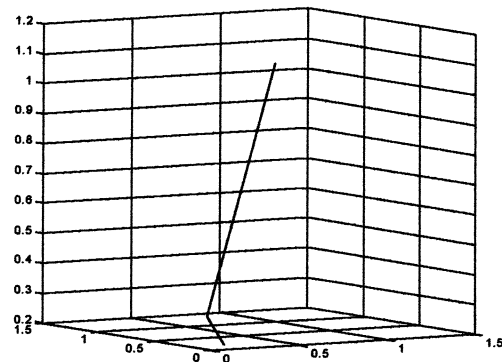
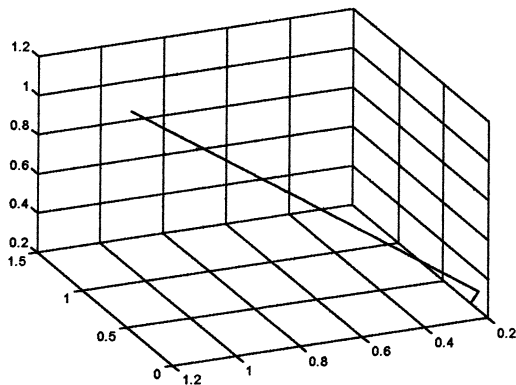
0.20000000000000	0.70000000000000	1.15033706530601	1.15027379885723
0.20000000000000	0.80000000000000	1.17356920230552	1.17351087099181
0.20000000000000	0.90000000000000	1.19725799676850	1.19721736312181
0.40000000000000	0.10000000000000	1.04083806402561	1.04081077419239
0.40000000000000	0.20000000000000	1.08334013295964	1.08328706767496
0.40000000000000	0.30000000000000	1.12757306029568	1.12749685157938
0.40000000000000	0.40000000000000	1.17360657049261	1.17351087099181
0.40000000000000	0.50000000000000	1.22151315843508	1.22140275816017
0.40000000000000	0.60000000000000	1.27136783862138	1.27124915032140
0.40000000000000	0.70000000000000	1.32324770702497	1.32312981233744
0.40000000000000	0.80000000000000	1.37723127212514	1.37712776433596
0.40000000000000	0.90000000000000	1.43339751735857	1.43332941456034
0.60000000000000	0.10000000000000	1.06188356935379	1.06183654654536
0.60000000000000	0.20000000000000	1.12758741386160	1.12749685157938
0.60000000000000	0.30000000000000	1.19734590641792	1.19721736312181
0.60000000000000	0.40000000000000	1.27140823225000	1.27124915032140
0.60000000000000	0.50000000000000	1.35003902871668	1.34985880757600
0.60000000000000	0.60000000000000	1.43351892421954	1.43332941456034
0.60000000000000	0.70000000000000	1.52214494033715	1.52196155561863
0.60000000000000	0.80000000000000	1.61623071268883	1.61607440219289
0.60000000000000	0.90000000000000	1.71610648107052	1.71600686218486
0.80000000000000	0.10000000000000	1.08336315518792	1.08328706767496
0.80000000000000	0.20000000000000	1.17365605349540	1.17351087099181
0.80000000000000	0.30000000000000	1.27145333431677	1.27124915032140
0.80000000000000	0.40000000000000	1.37737799704461	1.37712776433596
0.80000000000000	0.50000000000000	1.49210506166548	1.49182469764127
0.80000000000000	0.60000000000000	1.61636541872019	1.61607440219289
0.80000000000000	0.70000000000000	1.75094984572886	1.75067250029610
0.80000000000000	0.80000000000000	1.89671314880189	1.89648087930495
0.80000000000000	0.90000000000000	2.05457836478261	2.05443321064389
1.00000000000000	0.10000000000000	1.10528923253656	1.10517091807565
1.00000000000000	0.20000000000000	1.22162666016370	1.22140275816017
1.00000000000000	0.30000000000000	1.35017171231952	1.34985880757600
1.00000000000000	0.40000000000000	1.49220624625985	1.49182469764127
1.00000000000000	0.50000000000000	1.64914688837433	1.64872127070013
1.00000000000000	0.60000000000000	1.82255861410442	1.82211880039051
1.00000000000000	0.70000000000000	2.01416964839075	2.01375270747048
1.00000000000000	0.80000000000000	2.22588775823470	2.22554092849247
1.00000000000000	0.90000000000000	2.45981798666469	2.45960311115695
1.20000000000000	0.10000000000000	1.12767143482405	1.12749685157938
1.20000000000000	0.20000000000000	1.27157712481134	1.27124915032140
1.20000000000000	0.30000000000000	1.43378600967042	1.43332941456034
1.20000000000000	0.40000000000000	1.61663073220342	1.61607440219289
1.20000000000000	0.50000000000000	1.82274044832703	1.82211880039051
1.20000000000000	0.60000000000000	2.05507784252857	2.05443321064389
1.20000000000000	0.70000000000000	2.31698081462295	2.31636697678109
1.20000000000000	0.80000000000000	2.61220938210236	2.61169647342312
1.20000000000000	0.90000000000000	2.94499842317301	2.94467955106552
1.40000000000000	0.10000000000000	1.15051143402083	1.15027379885723
1.40000000000000	0.20000000000000	1.32357276085580	1.32312981233744

1.40000000000000	0.30000000000000	1.52257671867798	1.52196155561863
1.40000000000000	0.40000000000000	1.75142400857731	1.75067250029610
1.40000000000000	0.50000000000000	2.01459853056774	2.01375270747048
1.40000000000000	0.60000000000000	2.31725395455971	2.31636697678109
1.40000000000000	0.70000000000000	2.66531313386324	2.66445624192942
1.40000000000000	0.80000000000000	3.06558226163980	3.06485420329300
1.40000000000000	0.90000000000000	3.52588218784075	3.52542148736538
1.60000000000000	0.10000000000000	1.17379343742893	1.17351087099181
1.60000000000000	0.20000000000000	1.37764825885673	1.37712776433596
1.60000000000000	0.30000000000000	1.61679494864401	1.61607440219289
1.60000000000000	0.40000000000000	1.89736538700933	1.89648087930495
1.60000000000000	0.50000000000000	2.22654874705098	2.22554092849247
1.60000000000000	0.60000000000000	2.61277390395602	2.61169647342312
1.60000000000000	0.70000000000000	3.06592237403731	3.06485420329300
1.60000000000000	0.80000000000000	3.59757678735559	3.59663972556928
1.60000000000000	0.90000000000000	4.22131124411749	4.22069581699655
1.80000000000000	0.10000000000000	1.19746682243843	1.19721736312181
1.80000000000000	0.20000000000000	1.43377757150025	1.43332941456034
1.80000000000000	0.30000000000000	1.71662207204424	1.71600686218486
1.80000000000000	0.40000000000000	2.05519219998037	2.05443321064389
1.80000000000000	0.50000000000000	2.46048257836377	2.45960311115695
1.80000000000000	0.60000000000000	2.94564675084183	2.94467955106552
1.80000000000000	0.70000000000000	3.52642005613922	3.52542148736538
1.80000000000000	0.80000000000000	4.22162102599763	4.22069581699655
1.80000000000000	0.90000000000000	5.05374320382521	5.05309031656387

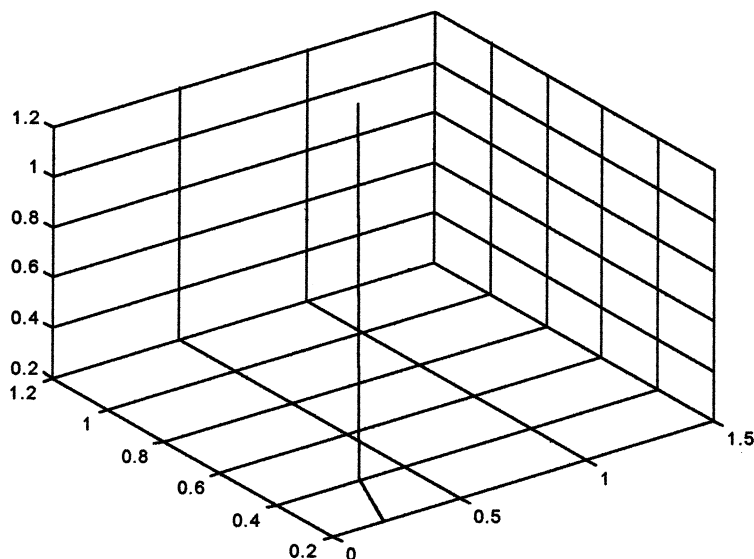
Según los resultados obtenidos podemos observar que aunque se ha utilizado una tolerancia de 10^{-9} , los errores son del orden de 10^{-3} , por lo que podemos deducir que los resultados obtenidos no son suficientemente buenos, con lo que habría que utilizar una tolerancia mayor para poder acercarnos a la solución real.

Probamos a aumentar la tolerancia y el número máximo de iteraciones y a pesar de ello no conseguimos acercarnos más a la solución dada por la ecuación $u(x, y) = e^{-xy}$

Las gráficas con los valores del método obtenidas son las que siguen:



La gráfica de la función real es:



Como era de prever gráficamente no se pueden apreciar las pequeñas diferencias que hay entre la solución real y la obtenida por el método de Poisson.

6.2.3 Por último resolveremos la siguiente ecuación y compararemos los resultados obtenidos por el método, con los resultados de la ecuación dada, utilizaremos una tolerancia de 10^{-9} y utilizaremos $h = \pi/5$ y $k = \pi/10$.

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = -(\cos(x+y) + \cos(x-y)), \quad 0 < x < \pi, \quad 0 < y < \frac{\pi}{2};$$

$$u(x,0) = \cos(x), u(x, \frac{\pi}{2}) = 0, 0 \leq x \leq \pi;$$

$$u(0,y) = \cos(y), u(\pi,y) = -\cos(y), 0 \leq y \leq \frac{\pi}{2}.$$

Comparar los resultados con los de la solución $u(x,y) = \cos x \cdot \cos y$.

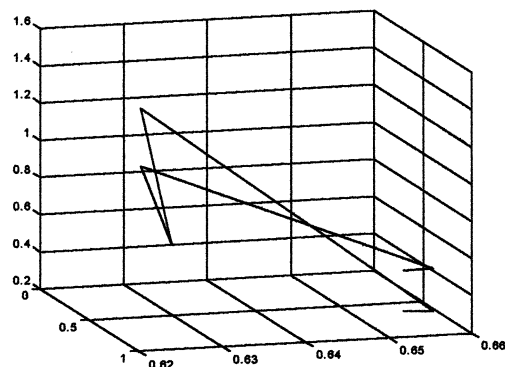
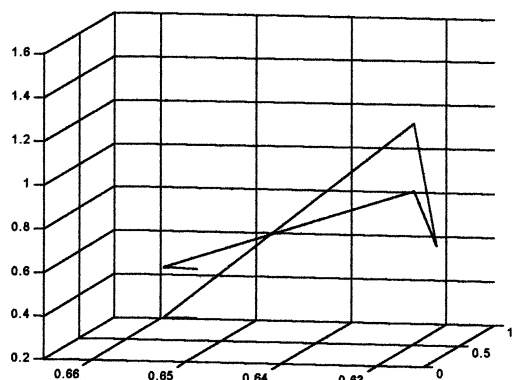
Obtenemos los valores y los comparamos midiendo el error cometido entre ambas soluciones:

x_i	y_j	$w_{i,j}$	REAL	ERROR
0.62831853071796	0.31415926535898	0.77219505406224	0.76942088429381	$2.77 \cdot 10^{-3}$
0.62831853071796	0.62831853071796	0.65803268074414	0.65450849718747	$3.52 \cdot 10^{-3}$
0.62831853071796	0.94247779607694	0.47851741035211	0.47552825814758	$2.98 \cdot 10^{-3}$
0.62831853071796	1.25663706143592	0.25167877854427	0.25000000000000	
1.25663706143592	0.31415926535898	0.29518494386555	0.29389262614624	$1.67 \cdot 10^{-3}$

1.25663706143592	0.62831853071796	0.25168075828005	0.25000000000000	1.29 10 ⁻³
1.25663706143592	0.94247779607694	0.18308094824005	0.18163563200134	1.68 10 ⁻³
1.25663706143592	1.25663706143592	0.09630905869099	0.09549150281253	1.44 10 ⁻³
1.88495559215388	0.31415926535898	-0.29518494279605	-0.29389262614624	8.17 10 ⁻³
1.88495559215388	0.62831853071796	-0.25168075614106	-0.25000000000000	1.29 10 ⁻³
1.88495559215388	0.94247779607694	-0.18308094559611	-0.18163563200134	1.68 10 ⁻³
1.88495559215388	1.25663706143592	-0.09630905667121	-0.09549150281253	1.68 10 ⁻³
2.51327412287183	0.31415926535898	-0.77219505337147	-0.76942088429381	-
2.51327412287183	0.62831853071796	-0.65803267936260	-0.65450849718747	-
2.51327412287183	0.94247779607694	-0.47851740864444	-0.47552825814758	-
2.51327412287183	1.25663706143592	-0.25167877723973	-0.25000000000000	-

Como podemos observar los errores cometidos son bastantes considerables y ocurre igual que en el ejemplo anterior, con lo que queda demostrado que el método no es lo suficiente bueno, a pesar de que se utiliza una tolerancia bastante buena.

Las gráficas que obtenemos son las siguientes, aplicamos las propiedad “rotate3d” para poder obtener varias perspectivas de la gráfica resultante, en esta gráfica queda representada de color azul los datos obtenidos del método y de color rojo la ecuación $u(x, y) = \cos x \cdot \cos y$:



Como podemos observar los datos obtenidos varían, lo que provoca que las gráficas se vean notablemente distintas y no coincidan una con otra que sería lo más normal si los valores fueran iguales.

6.3.-Código fuente.

Utilizamos tres ficheros:

- f.m donde meteremos la fuerza externa que se le aplica e la ecuación.
- g.m donde definiremos la función g a trozos, según el valor de x e y.
- Elíptico.m, fichero principal del método, hace las llamadas correspondientes a f y g, cuando le es necesario.

```

function eliptico
clear;
format long;
disp(' Introduzca los valores de entrada ');
a=input(' a: '); % extremo inferior del intervalo en x
b=input(' b: '); % extremo superior del intervalo en x
c=input(' c: '); % extremo inferior del intervalo en y
d=input(' d: '); % extremo superior del intervalo en y
n=input(' n: '); % divisiones del intervalo en x
m=input(' m: '); % divisiones del intervalo en y
tol=10.^input(' tolerancia (solo exponente): ');
iterac=input(' n° max iteraciones: ');
h=(b-a)/n; % paso en x
k=(d-c)/m; % paso en y
for i=1:1:n-1
    x(i)=a+i*h;
end
for j=1:1:m-1
    y(j)=c+j*k;
end
for i=1:1:n-1
    for j=1:1:m-1
        w(i,j)=0;
    end
end
landa=h.^2/k.^2;
mu=2*(1+landa);
l=1;
% realización de las iteraciones de Gauss-Seidel
while l<=iterac
    z=(-h.^2*f(x(1),y(m-1))+g(a,y(m-1))+landa*g(x(1),d)+landa*w(1,m-2)+w(2,m-1))/mu;
    norm=abs(z-w(1,m-1));
    w(1,m-1)=z;
    for i=2:1:n-2
        z=(-h.^2*f(x(i),y(m-1))+landa*g(x(i),d)+w(i-1,m-1)+w(i+1,m-1)+landa*w(i,m-2))/mu;
        if abs(w(i,m-1)-z)>norm
            norm=abs(w(i,m-1)-z);
        end
        w(i,m-1)=z;
    end
    z=(-h.^2*f(x(n-1),y(m-1))+g(b,y(m-1))+landa*g(x(n-1),d)+w(n-2,m-1)+landa*w(n-1,m-2))/mu;
    if abs(w(n-1,m-1)-z)>norm
        norm=abs(w(n-1,m-1)-z);
    end
    w(n-1,m-1)=z;
    for j=m-2:-1:2

```

```

        z=(-
h.^2*f(x(1),y(j))+g(a,y(j))+landa*w(1,j+1)+landa*w(1,j-
1)+w(2,j))/mu;
        if abs(w(1,j)-z)>norm
            norm=abs(w(1,j)-z);
        end
        w(1,j)=z;
        for i=2:1:n-2
            z=(-h.^2*f(x(i),y(j))+w(i-
1,j)+landa*w(i,j+1)+w(i+1,j)+landa*w(i,j-1))/mu;
            if abs(w(i,j)-z)>norm
                norm=abs(w(i,j)-z);
            end
            w(i,j)=z;
        end
        z=(-h.^2*f(x(n-1),y(j))+g(b,y(j))+w(n-2,j)+landa*w(n-
1,j+1)+landa*w(n-1,j-1))/mu;
        if abs(w(n-1,j)-z)>norm
            norm=abs(w(n-1,j)-z);
        end
        w(n-1,j)=z;
    end
    z=(-
h.^2*f(x(1),y(1))+g(a,y(1))+landa*g(x(1),c)+landa*w(1,2)+w(
2,1))/mu;
        if abs(w(1,1)-z)>norm
            norm=abs(w(1,1)-z);
        end
        w(1,1)=z;
        for i=2:1:n-2
            z=(-h.^2*f(x(i),y(1))+landa*g(x(i),c)+w(i-
1,1)+landa*w(i,2)+w(i+1,1))/mu;
            if abs(w(i,1)-z)>norm
                norm=abs(w(i,1)-z);
            end
            w(i,1)=z;
        end
        z=(-h.^2*f(x(n-1),y(1))+g(b,y(1))+landa*g(x(n-
1),c)+landa*w(n-1,2)+w(n-2,1))/mu;
        if abs(w(n-1,1)-z)>norm
            norm=abs(w(n-1,1)-z);
        end
        w(n-1,1)=z;
    if norm<=tol
        c=1;
        for i=1:1:n-1
            for j=1:1:m-1
                fin(c,1)=(x(i));
                fin(c,2)=(y(j));
                fin(c,3)=w(i,j);
                fin(c,4)=exp(x(i)*y(j));
            end
        end
    end
end

```

```
        c=c+1;
    end
end
fin
break;
else
    l=l+1;
end
end
if l>iterac
    disp('no se llega a una solución con ese número de
iteraciones y tolerancia');
end
```

7.-ECUACIONES DIFERENCIALES PARCIALES HIPERBÓLICAS.

7.1. Conceptos teóricos.

En esta sección consideraremos la solución numérica de la ecuación de onda, la cual es un ejemplo de una ecuación diferencial parcial hiperbólica. La ecuación de la onda está dada por la ecuación diferencial.

$$\frac{\partial^2 u}{\partial t^2}(x,t) - \alpha^2 \cdot \frac{\partial^2 u}{\partial x^2}(x,t) = 0 \rightarrow 0 < x < 1, t > 0.$$

Sujeta a las condiciones.

$$u(0,t) = u(1,t) = 0, \rightarrow t > 0.$$

$$u(x,0) = f(x), \rightarrow 0 \leq x \leq 1.$$

$$\frac{\partial u}{\partial t}(x,0) = g(x), \rightarrow 0 \leq x \leq 1.$$

Donde alfa es una constante. Para preparar el método de diferencia finita selección mas un entero $m > 0$ y un tamaño de paso de tiempo $k > 0$. Con $h=1/m$, los puntos de red

(x_i, t_j)

$$x_i = i \cdot h, \text{ para cada } i=0,1,\dots,m$$

$$t_j = j \cdot k, \text{ para cada } j=0,1,\dots,m$$

En cualquier punto interior de la red (x_i, t_j) la ecuación de onda será:

$$\frac{\partial^2 u}{\partial t^2}(x_i, t_j) - \alpha^2 \cdot \frac{\partial^2 u}{\partial x^2}(x_i, t_j) = 0.$$

El método de diferencia se obtiene usando el cociente de diferencia centrada para las segundas derivadas parciales, que está dado por:

$$\frac{\partial^2 u}{\partial t^2}(x_i, t_j) = \frac{u(x_i, t_{j+1}) - 2u(x_i, t_j) + u(x_i, t_{j-1}))}{k^2} - \frac{k^2}{12} \cdot \frac{\partial^4 u}{\partial x^4}(x_i, t_j)$$

Donde $t_{j-1} \leq t_j \leq t_{j+1}$.

y

$$\frac{\partial^2 u}{\partial x^2}(x_i, t_j) = \frac{u(x_{i+1}, t_j) - 2u(x_i, t_j) + u(x_{i-1}, t_j))}{h^2} - \frac{h^2}{12} \cdot \frac{\partial^4 u}{\partial x^4}(x_i, t_j)$$

Donde $x_{i-1} \leq x_i \leq x_{i+1}$.

sustituyendo se obtiene

$$\frac{u(x_i, t_{j+1}) - 2u(x_i, t_j) + u(x_i, t_{j-1}))}{k^2} - \alpha^2 \cdot \frac{u(x_{i+1}, t_j) - 2u(x_i, t_j) + u(x_{i-1}, t_j))}{h^2}$$

$$= \frac{1}{12} \left[k^2 \cdot \frac{\partial^4 u}{\partial t^4}(x_i, \mu_j) \alpha^2 \cdot h^2 \cdot \frac{\partial^4 u}{\partial x^4}(\varepsilon_i, t_j) \right]$$

Despreciando el error de truncamiento.

$$t_{i,j} = \frac{1}{12} \left[k^2 \cdot \frac{\partial^4 u}{\partial t^4}(x_i, \mu_j) \alpha^2 \cdot h^2 \cdot \frac{\partial^4 u}{\partial x^4}(\varepsilon_i, t_j) \right]$$

Nos lleva a la ecuación de diferencia :

$$\frac{w_{i,j+1} - 2w_{i,j} + w_{i,j-1}}{k^2} - \alpha^2 \cdot \frac{w_{i+1,j} - 2w_{i,j} + w_{i-1,j}}{h^2} = 0.$$

Si se usa λ para denotar $\alpha \cdot \frac{k}{h}$ podemos escribir la ecuación de diferencia como

$$\frac{w_{i,j+1} - 2w_{i,j} + w_{i,j-1}}{k^2} - \lambda^2 \cdot \frac{w_{i+1,j} - 2w_{i,j} + w_{i-1,j}}{h^2} = 0.$$

Y despejar $w_{i,j+1}$, la aproximación más avanzada en el tiempo, para obtener :

$$w_{i,j+1} = 2(1 - \lambda^2) \cdot w_{i,j} + \lambda^2 \cdot (w_{i+1,j} + w_{i-1,j}) - w_{i,j-1}$$

Esta ecuación se satisface para cada $i=1,2,\dots,(m-1)$, y $j=1,2,\dots$. Las condiciones de frontera dan :

$$w_{0,j} = w_{m,j} = 0 \quad \text{Para cada } i = 1, 2, \dots, m-1.$$

Escribiendo este conjunto de ecuaciones en forma matricial tenemos que .

$$\begin{bmatrix} w_{1,j+1} \\ w_{2,j+1} \\ \vdots \\ w_{m-1,j+1} \end{bmatrix} = \begin{bmatrix} 2(1-\lambda^2) & \lambda^2 & 0 & 0 \\ \cdot & \cdot & \cdot & 0 \\ 0 & \cdot & \cdot & \lambda^2 \\ 0 & 0 & \lambda^2 & 2(1-\lambda^2) \end{bmatrix} \cdot \begin{bmatrix} w_{1,j} \\ w_{2,j} \\ \vdots \\ w_{m-1,j} \end{bmatrix} - \begin{bmatrix} w_{1,j-1} \\ w_{2,j-1} \\ \vdots \\ w_{m-1,j-1} \end{bmatrix}$$

Las ecuaciones implican que el paso $(j+1)$ en el tiempo requiere de los valores de los pasos j y $(j-1)$ en el tiempo. Esto provoca un pequeño problema inicial, ya que los valores para $j = 1$, que son necesarios en la ecuación para calcular $w_{i,2}$ deben obtenerse de la condición de la velocidad inicial.

$$\frac{\partial u}{\partial t}(x,0) = g(x), \rightarrow 0 \leq x \leq 1.$$

Lo primero que se podría hacer sería reemplazar $\frac{\partial u}{\partial t}$ por la aproximación de diferencia progresiva.

$$\frac{\partial u}{\partial t}(x_i, 0) = \frac{u(x_i, t_j) - u(x_i, 0)}{k} - \frac{k}{2} \cdot \frac{\partial^2 u}{\partial t^2}(x_i, \bar{\mu}_j) \rightarrow 0 < \bar{\mu}_j < t_j.$$

Resolviendo para $u(x_i, t_j)$ da.

$$\begin{aligned} u(x_i, t_j) &= u(x_i, 0) + \frac{\partial u}{\partial t}(x_i, 0) \cdot k - \frac{k^2}{2} \cdot \frac{\partial^2 u}{\partial t^2}(x_i, \bar{\mu}_j) \\ &= u(x_i, 0) + k g(x_i) - \frac{k^2}{2} \cdot \frac{\partial^2 u}{\partial t^2}(x_i, \bar{\mu}_j) \end{aligned}$$

Como consecuencia

$$w_{i,1} = w_{i,0} + k \cdot g(x_i). \quad \text{para cada } i=1, \dots, m-1.$$

Sin embargo de la ecuación podemos ver que da una aproximación que tiene un error de truncamiento de sólo $O(k)$. Una mejor aproximación a $u(x_i, 0)$ puede obtenerse bastante fácilmente, particularmente cuando se puede determinar la segunda derivada de f en x_i , considerese:

$$\frac{u(x_i, t_j) - u(x_i, 0)}{k} = \frac{\partial u}{\partial t}(x_i, 0) - \frac{k}{2} \cdot \frac{\partial^2 u}{\partial t^2}(x_i, 0) + \frac{k^2}{6} \cdot \frac{\partial^3 u}{\partial t^3}(x_i, \bar{\mu}_j)$$

para alguna $0 < \bar{\mu}_j < t_j$ y supongamos que la ecuación de la onda se satisface también en la línea inicial, esto es:

$$\frac{\partial^2 u}{\partial t^2}(x_i, 0) + \alpha^2 \cdot \frac{\partial^2 u}{\partial x^2}(x_i, 0) = 0 \quad \text{para cada } i=0, 1, \dots, m.$$

Si f'' existe, entonces

$$\frac{\partial^2 u}{\partial t^2}(x_i, 0) = -\alpha^2 \cdot \frac{\partial^2 u}{\partial x^2}(x_i, 0) = -\alpha^2 \cdot \frac{d^2 f}{dx^2}(x_i) = -\alpha^2 \cdot f''(x_i)$$

Sustituyendo esto en la ecuación del párrafo anterior, nos queda.

$$u(x_i, t_j) = u(x_i, 0) - k \cdot g(x_i) + \frac{\alpha^2 \cdot k^2}{2} f''(x_i) + \frac{k^2}{6} \cdot \frac{\partial^3 u}{\partial t^3}(x_i, \bar{\mu}_j)$$

y

$$w_{i,1} = w_{i,0} + k \cdot g(x_i) + \frac{\alpha^2 \cdot k^2}{2} f''(x_i).$$

Esta es una aproximación con error de truncamiento local $O(k^2)$ para cada $i=1, 2, \dots, m-1$.

Si $f'' \in C^4[0,1]$ pero $f''(x_i)$ no puede conseguirse fácilmente, podemos usar la ecuación anterior para escribir:

$$f''(x_i) = \frac{f(x_{i+1}) - 2f(x_i) + f(x_{i-1}))}{h^2} - \frac{h^2}{12} f^{(4)}(\zeta_i)$$

para alguna ζ_i en (x_{i-1}, x_{i+1}) . Esto implica que la aproximación se convierte en

$$\frac{u(x_i, t_i) - u(x_i, 0)}{k} = g(x_i) + \frac{k \cdot \alpha^2}{2 \cdot h^2} [f(x_{i+1}) - 2f(x_i) + f(x_{i-1}))] + O(k^2 + h^2 \cdot k)$$

O haciendo $\lambda = \frac{k \cdot \alpha}{h}$

$$\begin{aligned} u(x_i, t_i) &= u(x_i, 0) + k \cdot g(x_i) + \frac{\lambda^2}{2} [f(x_{i+1}) - 2f(x_i) + f(x_{i-1}))] + O(k^3 + h^2 \cdot k^2) \\ &= (1 - \lambda^2) \cdot f(x_i) + \frac{\lambda^2}{2} f(x_{i+1}) + \frac{\lambda^2}{2} f(x_{i-1}) + k \cdot g(x_i) + O(k^3 + h^2 \cdot k^2) \end{aligned}$$

Entonces, se puede usar la ecuación de diferencia.

$$w_{i,1} = (1 - \lambda^2) \cdot f(x_i) + \frac{\lambda^2}{2} f(x_{i+1}) + \frac{\lambda^2}{2} f(x_{i-1}) + k \cdot g(x_i)$$

para encontrar a $w_{i,1}$ para cada $i=1, 2, \dots, m-1$.

7.2. Ejemplos resueltos del algoritmo de diferencia finita para la ecuación de la onda:

EJERCICIO 1.

Consideramos el siguiente problema hiperbólico:

$$\frac{\partial^2 u}{\partial t^2}(x, t) - 4 \cdot \frac{\partial^2 u}{\partial x^2}(x, t) = 0 \quad \text{para } 0 < x < 1, \quad 0 < t$$

Y sus condiciones de la frontera son las siguientes:

$$u(0, t) = u(1, t) = 0, \quad 0 < t$$

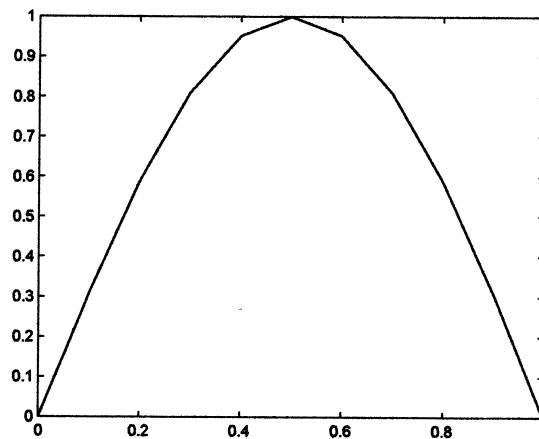
Y sus condiciones iniciales son:

$$u(x,0) = \text{sen}(\pi \cdot x) \quad \text{y} \quad \frac{\partial u}{\partial t}(x,0) = 0 \quad \text{para} \quad 0 \leq x \leq 1$$

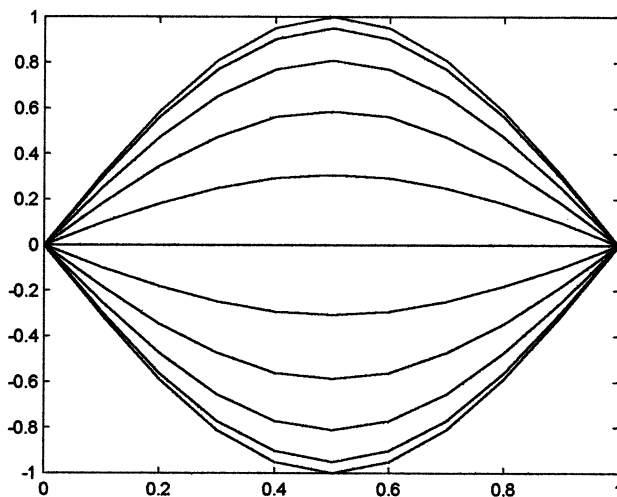
Las variables tomadas para nuestro algoritmo son las siguientes $m=20$ y $n=10$, para el tiempo de observación hemos utilizado $t=1$ y el valor de $\lambda = 1$. A continuación podemos ver la tabla de valores obtenidos.

Valor de X(i)	Valores Obtenidos
0	0
0.1	0.30901699437495
0.2	0.58778525229247
0.3	0.80901699437495
0.4	0.95105651629515
0.5	1.00000000000000
0.6	0.95105651629515
0.7	0.80901699437495
0.8	0.58778525229247
0.9	0.30901699437495
1	0

En esta gráfica podemos ver como queda representados los valores obtenidos para el tiempo $t=1$, que es el tiempo que se pedía para su estudio.



A continuación vamos hacer el estudio con los mismos valores que anteriormente pero ahora mostraremos las distintas representaciones para los distintos valores que va tomando el tiempo t , estos estados están representados en azul y el estado final, es decir para $t=1$ en rojo.



EJERCICIO 2.

Consideramos el siguiente problema hiperbólico:

$$\frac{\partial^2 u}{\partial t^2}(x,t) - \frac{\partial^2 u}{\partial x^2}(x,t) = 0 \quad \text{para } 0 < x < 1, \quad 0 < t$$

Y sus condiciones de la frontera son las siguientes :

$$u(0,t) = u(1,t) = 0, \quad 0 < t$$

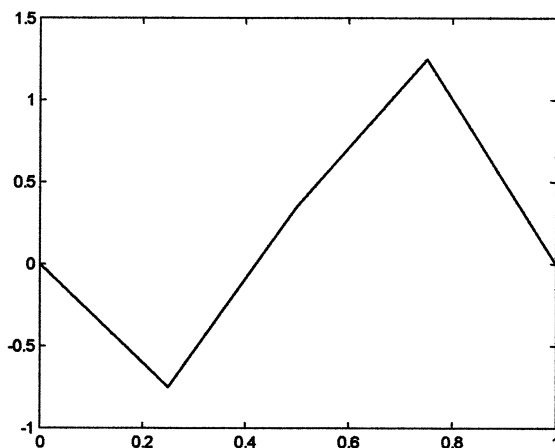
Y sus condiciones iniciales son :

$$u(x,0) = \sin(2 \cdot \pi \cdot x) \quad \text{y} \quad \frac{\partial u}{\partial t}(x,0) = \sin(\pi \cdot x) \quad \text{para } 0 \leq x \leq 1$$

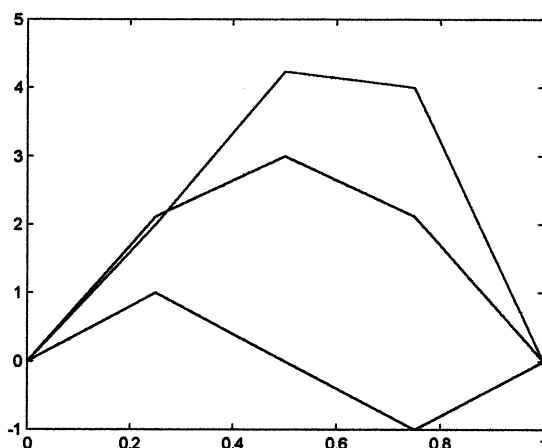
Las variables tomadas para nuestro algoritmo son las siguientes $m=4$ y $n=2$, para el tiempo de observación hemos utilizado $t=0.5$ y el valor de $\lambda=1$. A continuación podemos ver la tabla de valores obtenidos.

Valor de X(i)	Valores Obtenidos
0	0
0.25	-0.7500000000000000
0.50	0.35355339059327
0.75	1.2500000000000000
1.00	0

En esta gráfica podemos ver como queda representados los valores obtenidos para el tiempo $t=0.5$, que es el tiempo que se pedía para su estudio.



A continuación vamos hacer el estudio con los mismos valores que anteriormente pero ahora tomando el tiempo $t=6$ y además mostraremos las distintas representaciones para los distintos valores que va tomando el tiempo t , estos estados están representados en azul y el estado final, es decir para $t=6$ en rojo.



EJERCICIO 3.

Consideramos el siguiente problema hiperbólico:

$$\frac{\partial^2 u}{\partial t^2}(x, t) - \frac{\partial^2 u}{\partial x^2}(x, t) = 0 \quad \text{para } 0 < x < 1, \quad 0 < t$$

Y sus condiciones de la frontera son las siguientes :

$$u(0, t) = u(1, t) = 0, \quad 0 < t$$

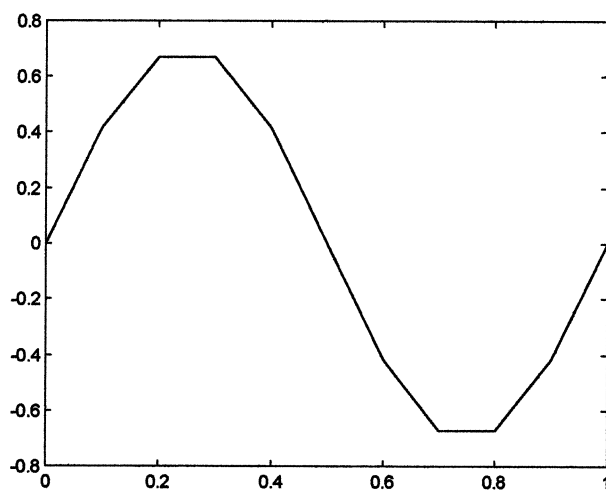
Y sus condiciones iniciales son :

$$u(x, 0) = \text{sen}(2 \cdot \pi \cdot x) \quad \text{y} \quad \frac{\partial u}{\partial t}(x, 0) = 2 \cdot \pi \text{sen}(2 \cdot \pi \cdot x) \quad \text{para } 0 \leq x \leq 1$$

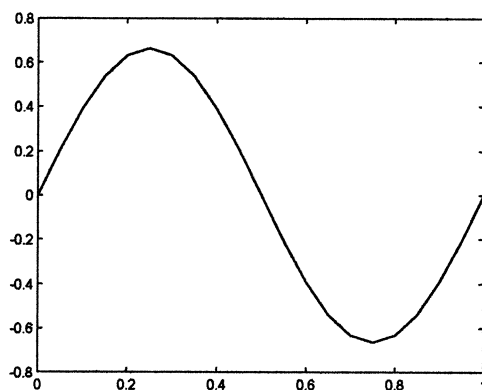
Las variables tomadas para nuestro algoritmo son las siguientes $m=10$ y $n=3$, para el tiempo de observación hemos utilizado $t=0.3$ y el valor de $\lambda=1$. A continuación podemos ver la tabla de valores obtenidos.

Valor de X(i)	Valores Obtenidos
0	0
0.1	0.41593080094697
0.2	0.67299017290017
0.3	0.67299017290017
0.4	0.41593080094697
0.5	0.00000000000000
0.6	-0.41593080094697
0.7	-0.67299017290017
0.8	-0.67299017290017
0.9	-0.41593080094697
1	0

En esta gráfica podemos ver como queda representados los valores obtenidos para el tiempo $t=0.3$, que es el tiempo que se pedía para su estudio.



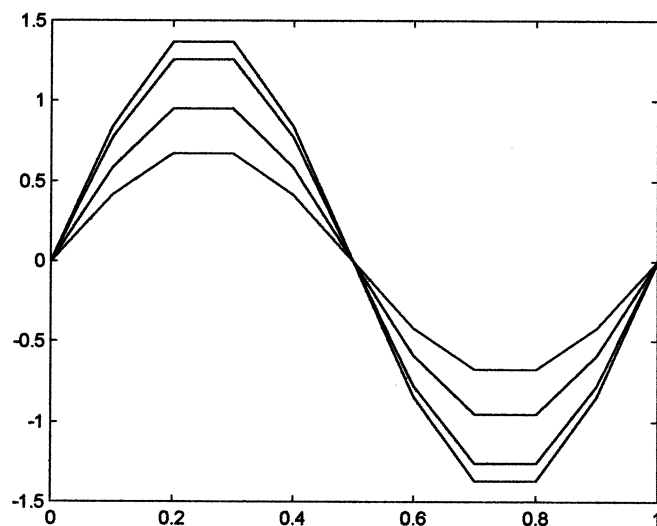
Ahora vamos a incrementar hasta 20 el valor de m , es decir el número de intervalos del eje x . Esta es la gráfica obtenida, como se puede ver la gráfica es más perfecta, con más suavidad:



A continuación podemos ver los valores obtenidos para $m=20$.

Valor de $X(i)$	Valores Obtenidos
0	0
0.05	0.20554745897380
0.10	0.39097450052993
0.15	0.53813023389457
0.20	0.63261003059194
0.25	0.66516554984156
0.30	0.63261003059193
0.35	0.53813023389460
0.40	0.39097450052987
0.45	0.20554745897388
0.50	-0.000000000000009
0.55	-0.20554745897375
0.60	-0.39097450052992
0.65	-0.53813023389462
0.70	-0.63261003059186
0.75	-0.66516554984170
0.80	-0.63261003059171
0.85	-0.53813023389483
0.90	-0.39097450052972
0.95	-0.20554745897391
1	0

A continuación vamos hacer el estudio con los mismos valores que anteriormente pero ahora mostraremos las distintas representaciones para los distintos valores que va tomando el tiempo t , estos estados están representados en azul y el estado final, es decir para $t=1$ en rojo.



7.3 Código fuente del algoritmo ecuación de onda.

El fichero principal es el edha .m ,en el se hacen todas las operaciones principales y es donde se pide el valor de la n ,m alfa ,tiempo de observación ..etc con la que queremos tratar.

Edha.m

```
%ALGORITMO DE LA ONDA.
```

```
%function edha(l,t,m,n,a)
%l=longitud max
%t=tiempo max
%v=lambda=k*alfa/h
%m=n° intervalos de x
%n=n° intervalos de t
%a es el valor de alfa.
%probar con (1,1,1,10,20)
clear;
l=input('introduzca el valor de l(longitud max o p.extremo):');
t=input('introduzca el valor de t(tiempo maximo):');
m=input('introduzca el valor de m(intervalos de x):');
n=input('introduzca el valor de n(intervalos de y):');
a=input('introduzca el valor de a(valor de alfa):');

format long;

h=l/m;
k=t/n;
x=0:h:l;

v=k*a/h;

for j=2:n+1
    w(1,j)=0;
    w(m+1,j)=0;
end

w(1,1)=f(0);
w(m+1,1)=f(1);

for i=1:m-1
    w(i+1,1)=f(i*h);
    w(i+1,2)=(1-v^2)*f(i*h) + (v^2)/2*( f((i+1)*h) + f((i-1)*h) )+k*g(i*h);
end

for j=2:n
    for i=2:m
        w(i,j+1)=2*(1-v^2)*w(i,j) + (v^2)*( w(i+1,j) + w(i-1,j) ) - w(i,j-1);
```

```
end  
end  
'valor de x'  
x'  
'valor obtenido por el metodo'  
w(:,n+1)
```

```
for j=1:n+1  
    y=w(:,j);
```

```
end
```

```
plot(x,y);
```

Por ultimo tenemos los ficheros donde metemos la funciones a estudiar ,de manera que tenemos que desglosarlas en las componentes f,g,freal (para comparación de resultados).

```
f.m  
function y=f(x)  
y=-4*(x.^2);
```

```
p.m  
function y=g(x)  
y=0;
```

```
freal.m  
function y=q(x)  
y=sen(pi*x);
```

8.-ECUACIONES DIFERENCIALES PARCIALES PARABÓLICAS.

8.1.-Conceptos teóricos.

La ecuación diferencial parcial parabólica que vamos a estudiar es la ecuación del calor ó de difusión:

$$(0) \quad \frac{\partial u}{\partial t}(x,t) = \alpha^2 \frac{\partial^2 u}{\partial x^2}(x,t), \quad 0 < x < 1, \quad t > 0,$$

sujeta a las condiciones

$$y \quad \begin{aligned} u(0,t) &= 0, & u(1,t) &= 0, & t > 0, \\ u(x,0) &= f(x), & 0 &\leq x \leq 1 \end{aligned}$$

El enfoque que usaremos para aproximar la solución a este problema consiste en el uso de diferencias finitas.

Primero seleccionamos dos constantes de malla h y k con la estipulación que $m=l/h$ es un entero. Los puntos de red para esta situación son (x_i, t_j) , donde $x_i = ih$ para $i=0,1,\dots,m$, y $t_j = jk$, para $j=0,1,\dots$.

Obtenemos el método de diferencia usando la serie de Taylor en t para formar el cociente diferencial

$$(1) \quad \frac{\partial u}{\partial t}(x_i, t_j) = \frac{u(x_i, t_j + k) - u(x_i, t_j)}{k} - \frac{k}{2} \frac{\partial^2 u}{\partial t^2}(x_i, \mu_j).$$

para alguna $\mu_j \in (t_j, t_{j+1})$, y la serie de Taylor en x para formar el cociente diferencial

$$(2) \quad \frac{\partial^2 u}{\partial x^2}(x_i, t_j) = \frac{u(x_i + h, t_j) - 2u(x_i, t_j) + u(x_i - h, t_j)}{h^2} - \frac{h^2}{12} \frac{\partial^4 u}{\partial x^4}(\xi_i, t_j)$$

donde $\xi_i \in (x_{i-1}, x_{i+1})$

La ecuación parcial (0) implica que en los puntos de red interiores (x_i, t_j) para cada $i=1,2,\dots$, tenemos que

$$\frac{\partial u}{\partial t}(x_i, t_j) - \alpha^2 \frac{\partial^2 u}{\partial x^2}(x_i, t_j) = 0;$$

así, el método de diferencia, utilizando los coeficientes diferenciales (1) y (2)

$$(3) \quad \frac{w_{i,j+1} - w_{i,j}}{k} - \alpha^2 \frac{w_{i+1,j} - 2w_{i,j} + w_{i-1,j}}{h^2} = 0$$

donde $w_{i,j}$ aproxima a $u(x_i, t_j)$.

El error de truncamiento local para esta ecuación de diferencia es

$$(4) \quad \tau_{i,j} = \frac{k}{2} \frac{\partial^2 u}{\partial t^2}(x_i, \mu_j) - \alpha^2 \frac{\partial^4 u}{\partial x^4}(\xi_i, t_j).$$

Si la ecuación (3) se resuelve para $w_{i,j+1}$

$$(5) \quad w_{i,j+1} = \left(1 - \frac{2\alpha^2 k}{l\tau^2}\right) w_{i,j} + \alpha^2 \frac{k}{h^2} (w_{i+1,j} + w_{i-1,j}).$$

para cada $i=1,2,\dots,(m-1)$ y $j=1,2,\dots$. Como la condición inicial $u(x,0) = f(x)$, para $0 \leq x \leq 1$, implica que $w_{i,0} = f(x_i)$ para cada $i=0,1,\dots,m$, estos valores pueden usarse en la ecuación (5) para encontrar el valor de $w_{j,l}$ para cada $i=1,2,\dots,(m-1)$. Las condiciones adicionales $u(0, t)=0$ y $u(l, t)=0$, implican que $w_{0,l} = w_{m,l} = 0$, así que todas las componentes de la forma $w_{i,l}$ pueden ser determinadas. Si el procedimiento se aplica nuevamente, una vez que se conocen todas las aproximaciones $w_{i,1}$, los valores de $w_{i,2}, w_{i,3}, \dots, w_{i,m-1}$ se pueden obtener de manera similar.

La naturaleza explícita del método de diferencia expresado en la ecuación (5) implica que la matriz de $(m-1) \times (m-1)$ asociada con este sistema puede escribirse en forma tridiagonal

$$A = \begin{bmatrix} (1-2\lambda) & \lambda & 0 & K & K & K & K & K & 0 \\ \lambda & (1-2\lambda) & \lambda & 0 & & & & & M \\ 0 & 0 & 0 & 0 & 0 & 0 & & & M \\ M & 0 & 0 & 0 & 0 & 0 & & & 0 \\ M & & 0 & 0 & 0 & 0 & 0 & & \lambda \\ 0 & K & K & 0 & \lambda & (1-2\lambda) & & & \end{bmatrix}$$

donde $\lambda = \alpha^2 (k/h^2)$. Si tomamos

$$w^{(0)} = (f(x_1), f(x_2), \dots, f(x_{m-1}))^t$$

y

$$w^{(j)} = (w_{1,j}, w_{2,j}, \dots, w_{m-1,j})^t \quad \text{para cada } j=1,2,\dots,$$

entonces la solución aproximada está dada por

$$w^{(j)} = Aw^{(j-1)} \quad \text{para cada } j=1,2,\dots,$$

Este método de diferencia se conoce como el método de diferencia progresiva. Si la solución de la ecuación diferencial parcial tiene cuatro derivadas parciales continuas en x y dos en t , entonces la ecuación (4) implica que el método es de orden $O(k+h^2)$.

Se puede demostrar que este método es **condicionalmente estable**, debido a que cualquier error de redondeo, puede hacer que este se propague hasta la solución final, de forma que el resultado no sea fiable. Esta condición de estabilidad viene determinada por las características de la matriz A , es decir, depende de sus autovalores. En concreto, el método converge a la solución de la ecuación (0) con una razón de convergencia de $O(k+h^2)$, siempre que

$$\alpha^2 (k/h^2) \leq \frac{1}{2}$$

y que se cumplan las condiciones de continuidad requeridas para la solución.

Para obtener un método que sea **incondicionalmente estable**, consideremos un método de diferencia implícito que resulta de usar el cociente de diferencia regresiva para $\frac{\partial u}{\partial t}(x_i, t_j)$ en la forma

$$\frac{\partial u}{\partial t}(x_i, t_j) = \frac{u(x_i, t_j) - u(x_i, t_{j-1})}{k} + \frac{k}{2} \frac{\partial^2 u}{\partial t^2}(x_i, t_j),$$

donde $\mu_j \in (t_{j-1}, t_j)$. Sustituyendo esta ecuación, junto con la ecuación (2) para $\frac{\partial^2 u}{\partial t^2}$, en la ecuación diferencial parcial se obtiene

$$\frac{u(x_i, t_j) - u(x_i, t_{j-1})}{k} - \alpha^2 \frac{u(x_{i+1}, t_j) - 2u(x_i, t_j) + u(x_{i-1}, t_j))}{h^2} = -\frac{k}{2} \frac{\partial^2 u}{\partial t^2}(x_i, \mu_j) - \frac{h^2}{12} \frac{\partial^4 u}{\partial t^4}(\xi_i, t_j),$$

para alguna $\xi_i \in (x_{i-1}, x_{i+1})$. El método de **diferencia regresiva** que resulta es

$$(6) \quad \frac{w_{i,j} - w_{i,j-1}}{k} - \alpha^2 \frac{w_{i+1,j} - 2w_{i,j} + w_{i-1,j}}{h^2} = 0 \quad \text{para cada } i=1,2,\dots,m-1 \text{ y } j=1,2,\dots$$

Este método involucra, en un paso típico, los puntos de red

$$(x_i, t_j), (x_i, t_{j-1}), (x_{i-1}, t_j), \quad \text{y} \quad (x_{i+1}, t_j),$$

Si usamos λ para denotar a la cantidad $\alpha^2(k/h^2)$, el método de diferencia regresiva como

$$(7) \quad (1+2\lambda)w_{i,j} - \lambda w_{i+1,j} - \lambda w_{i-1,j} = w_{i,j-1} \quad \text{para } i=1,2,\dots,m-1 \text{ y } j=1,2,\dots$$

Usando el hecho de que $w_{i,0} = f(x_i)$ para cada $i = 1,2,\dots,m-1$ y $w_{m,j} = w_{0,j} = 0$ para cada $j=1,2,\dots$, este método de diferencia tiene representación matricial

$$\begin{bmatrix} (1+2\lambda) & -\lambda & 0 & K & K & K & K & 0 \\ -\lambda & (1+2\lambda) & -\lambda & 0 & & & & M \\ 0 & 0 & 0 & 0 & 0 & & & M \\ M & 0 & 0 & 0 & 0 & & & 0 \\ M & & 0 & 0 & 0 & -\lambda & & \\ 0 & K & K & 0 & -\lambda & (1+2\lambda) & & \end{bmatrix} \begin{bmatrix} w_{1,j} \\ w_{2,j} \\ M \\ w_{m-1,j} \end{bmatrix} = \begin{bmatrix} w_{1,j-1} \\ w_{2,j-1} \\ M \\ w_{m-1,j-1} \end{bmatrix}$$

o $Aw^{(j)} = w^{(j-1)}$ para cada $j = 1, 2, \dots$. Como $\lambda > 0$, la matriz A es positiva definida y estrictamente dominante diagonalmente, además de ser tridiagonal y simétrica. Podemos usar, ya sea la reducción de Crout para sistemas lineales tridiagonales o el método de SOR para resolver el sistema.

El método de diferencia regresiva no tiene los problemas de estabilidad del método de diferencia progresiva. El método es estable independientemente de la elección de $\lambda = \alpha^2(k/h^2)$. El error de truncamiento local para este método es de orden $O(k+h^2)$, siempre que la solución de la ecuación diferencial satisfaga las condiciones usuales de diferenciabilidad.

La debilidad del método de diferencia regresiva resulta del hecho de que el error de truncamiento local tiene una porción con orden $O(k)$, lo cual requiere que los intervalos en el tiempo sean mucho más pequeños que los intervalos en el espacio. Claramente, sería muy deseable idear un procedimiento con error de truncamiento local $O(k^2+h^2)$. El primer paso en esta dirección consiste en usar una ecuación de diferencia que tenga un error de $O(k^2)$ para $u_i(x,t)$ en lugar de la que se usó previamente, cuyo error era de $O(k)$. Esto puede hacerse usando la serie de Taylor para la función $u(x,t)$ en el punto (x_i, t_j) y evaluándola en (x_i, t_{j+1}) y (x_i, t_{j-1}) para obtener la fórmula de diferencia centrada

$$\frac{\partial u}{\partial t}(x_i, t_j) = \frac{u(x_i, t_{j+1}) - u(x_i, t_{j-1})}{2k} - \frac{k^2}{6} \frac{\partial^3 u}{\partial t^3}(x_i, \mu_j).$$

donde $\mu_j \in (t_{j-1}, t_{j+1})$. El método de diferencia que resulta de sustituir ésta y el cociente diferencial usual para $(\partial^2 u / \partial x^2)$, ecuación (2), en la ecuación diferencial se llama **método de Richardson** y está dado por

$$\frac{w_{i,j+1} - w_{i,j-1}}{2k} - \alpha^2 \frac{w_{i+1,j} - 2w_{i,j} + w_{i-1,j}}{h^2} = 0$$

Este método sí tiene un error de truncamiento local de orden $O(k^2 + h^2)$, pero desafortunadamente, tiene también problemas de estabilidad serios.

Un método más efectivo puede derivarse promediando el método diferencia progresiva en el j -ésimo paso en t ,

$$\frac{w_{i,j+1} - w_{i,j}}{k} - \alpha^2 \frac{w_{i+1,j} - 2w_{i,j} + w_{i-1,j}}{h^2} = 0$$

que tiene un error de truncamiento local

$$\tau_F = \frac{k}{2} \frac{\partial^2 u}{\partial t^2}(x_i, \mu_j) + O(h^2),$$

con la fórmula de diferencia regresiva en el paso $(j + 1)$ en t ,

$$\frac{w_{i,j+1} - w_{i,j}}{k} - \alpha^2 \frac{w_{i+1,j+1} - 2w_{i,j+1} + w_{i-1,j+1}}{h^2} = 0$$

que tiene un error de truncamiento local

$$\tau_H = -\frac{k}{2} \frac{\partial^2 u}{\partial t^2}(x_i, \hat{\mu}_j) + O(h^2).$$

Si suponemos que $\mu_j \approx \hat{\mu}_j$, entonces el método de diferencia promediada,

$$\frac{w_{i,j+1} - w_{i,j}}{k} - \frac{\alpha^2}{2} \left[\frac{w_{i+1,j} - 2w_{i,j} + w_{i-1,j}}{h^2} + \frac{w_{i+1,j+1} - 2w_{i,j+1} + w_{i-1,j+1}}{h^2} \right] = 0,$$

tiene un error de truncamiento local de orden $O(k^2 + h^2)$, que desde luego, se satisfagan las condiciones usuales de diferenciabilidad.

Este método se conoce como el método **Crank-Nicolson** y puede representarse en la forma matricial $Aw^{(j+1)} = Bw^{(j)}$ para cada $j = 0, 1, 2, \dots$, donde

$$\lambda = \alpha^2 (k/h^2)$$

A y B están dadas por:

$$w^{(j)} = (w_{1,j}, w_{2,j}, \dots, w_{m-1,j})^t, \text{ y las matrices}$$

$$A = \begin{bmatrix} (1+\lambda) & -\frac{\lambda}{2} & 0 & K & K & K & K & K & 0 \\ -\frac{\lambda}{2} & (1+\lambda) & -\frac{\lambda}{2} & 0 & & & & & M \\ 0 & 0 & 0 & 0 & 0 & & & & M \\ M & 0 & 0 & 0 & 0 & & & & 0 \\ M & 0 & 0 & 0 & -\frac{\lambda}{2} & & & & \\ 0 & K & K & 0 & -\frac{\lambda}{2} & (1+\lambda) & & & \end{bmatrix} \quad \text{y} \quad B = \begin{bmatrix} (1-\lambda) & \frac{\lambda}{2} & 0 & K & K & K & K & K & 0 \\ \frac{\lambda}{2} & (1-\lambda) & \frac{\lambda}{2} & 0 & & & & & M \\ 0 & 0 & 0 & 0 & 0 & & & & M \\ M & 0 & 0 & 0 & 0 & & & & 0 \\ M & 0 & 0 & 0 & \frac{\lambda}{2} & & & & \\ 0 & K & K & 0 & \frac{\lambda}{2} & (1-\lambda) & & & \end{bmatrix}$$

Como A es una matriz tridiagonal, positiva definida, simétrica y estrictamente dominante diagonalmente, será no singular. Se pueden usar entonces, ya sea la reducción de Crout para un sistema lineal tridiagonal o el método SOR para obtener w^{j+1} de w^j , para cada $j = 0, 1, 2, \dots$.

8.2.-Ejemplos de aplicación.

8.2.1.- Consideremos la siguiente ecuación diferencial parcial parabólica:

$$\frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} = 0 \quad 0 < x < 2, \quad 0 < t;$$

$$u(0, t) = u(2, t) = 0, \quad 0 < t;$$

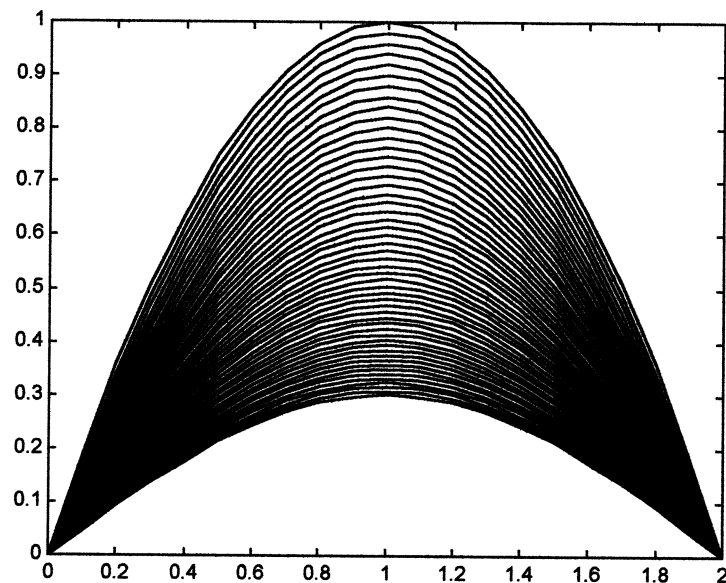
$$u(x, 0) = x(2 - x), \quad 0 \leq x \leq 2.$$

Aplicaremos el método de Crank-Nicolson. Tomaremos $h = 0.1$, con lo que el número de pasos m será 20; $T = 0.5$ (valor del tiempo de observación) y $N = 50$ (Divisiones del tiempo).

El resultado que se obtiene es el siguiente:

x_i	$w_{i,j}$
0	0
0.100000000000000	0.04713233249356
0.200000000000000	0.09310405120957
0.300000000000000	0.13678313346636
0.400000000000000	0.17709403194709
0.500000000000000	0.21304416333409
0.600000000000000	0.24374834796075
0.700000000000000	0.26845059892333
0.800000000000000	0.28654272564709
0.900000000000000	0.29757929629750
1.000000000000000	0.30128859350236
1.100000000000000	0.29757929629750
1.200000000000000	0.28654272564709
1.300000000000000	0.26845059892333
1.400000000000000	0.24374834796075
1.500000000000000	0.21304416333409
1.600000000000000	0.17709403194709
1.700000000000000	0.13678313346636
1.800000000000000	0.09310405120957
1.900000000000000	0.04713233249356
2.000000000000000	0

Y la gráfica que se obtiene con el tiempo es la siguiente:



Si aplicamos el método regresivo se obtiene:

x_i	$w_{i,j}$
0	0
0.1000000000000000	0.04784407329081
0.2000000000000000	0.09450989151093
0.3000000000000000	0.13884825033154
0.4000000000000000	0.17976732325653
0.5000000000000000	0.21625956104487
0.6000000000000000	0.24742649783497
0.7000000000000000	0.27250085383386
0.8000000000000000	0.29086539480319
0.9000000000000000	0.30206809125921
1.0000000000000000	0.30583321265208
1.1000000000000000	0.30206809125921
1.2000000000000000	0.29086539480319
1.3000000000000000	0.27250085383386
1.4000000000000000	0.24742649783497
1.5000000000000000	0.21625956104487
1.6000000000000000	0.17976732325653
1.7000000000000000	0.13884825033154
1.8000000000000000	0.09450989151093
1.9000000000000000	0.04784407329081
2.0000000000000000	0

Y como puede observarse, no da el mismo resultado. Esto se deberá a que el método de Crank-Nicolson tiene mayor precisión, como hemos visto en el apartado de teoría.

8.2.2.- Consideremos la siguiente ecuación diferencial parcial parabólica:

$$\frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} = 0 \quad 0 < x < 1, \quad 0 < t;$$

$$u(0,t) = u(1,t) = 0, \quad 0 < t;$$

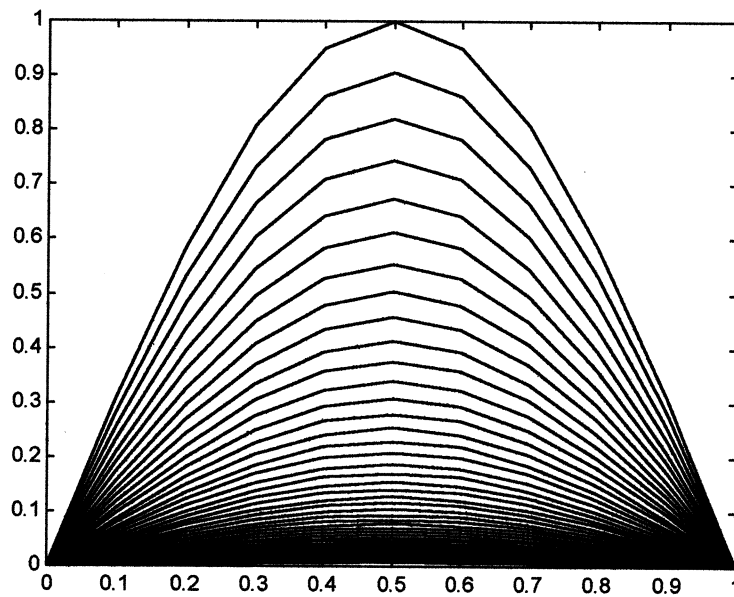
$$u(x,0) = \text{sen}(\pi x), \quad 0 \leq x \leq 1.$$

Aplicaremos el método de Crank-Nicolson. Tomaremos $h = 0.1$, con lo que el número de pasos m será 10; $T = 0.5$ y $N = 50$.

El resultado que se obtiene es el siguiente:

x_i	$w_{i,j}$
0	0
0.1000000000000000	0.00230512336779
0.2000000000000000	0.00438460519960
0.3000000000000000	0.00603489132513
0.4000000000000000	0.00709444024020
0.5000000000000000	0.00745953591469
0.6000000000000000	0.00709444024020
0.7000000000000000	0.00603489132513
0.8000000000000000	0.00438460519960
0.9000000000000000	0.00230512336779
1.0000000000000000	0

Y la gráfica que se obtiene es:



8.2.3.- Consideremos ahora la siguiente ecuación diferencial parcial parabólica:

$$\frac{\partial u}{\partial t} - \frac{1}{16} \frac{\partial^2 u}{\partial x^2} = 0 \quad 0 < x < 1, \quad 0 < t;$$

$$u(0,t) = u(1,t) = 0, \quad 0 < t;$$

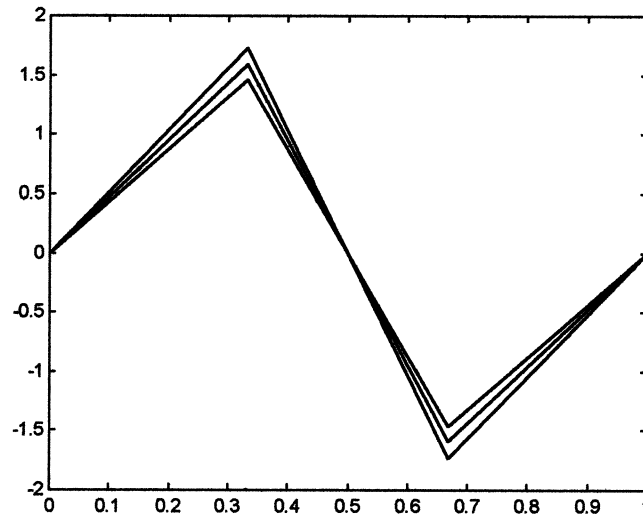
$$u(x,0) = 2 \operatorname{sen}(2\pi x), \quad 0 \leq x \leq 1.$$

Aplicaremos el método de Crank-Nicolson. Tomaremos $m=3$, $T=0.1$ y $N=2$. Como puede observarse, hay que tomar $\alpha=1/4$.

El resultado que se obtiene es:

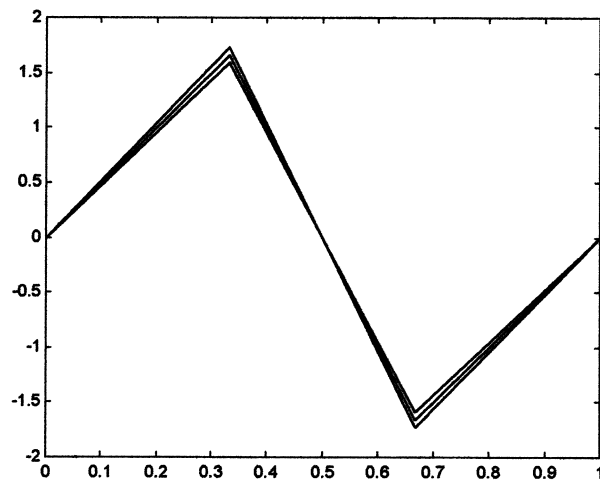
x_i	$w_{i,j}$
0	0
0.333333333333333	1.46295143262555
0.666666666666667	-1.46295143262555
1.000000000000000	0

Y el gráfico es:



Vamos a considerar ahora como tiempo de observación $T = 0.05$:

x_i	$w_{i,j}$
0	0
0.333333333333333	1.59188464699257
0.666666666666667	-1.59188464699257
1.000000000000000	0



8.2.4.- Veamos un último ejemplo de aplicación:

$$\frac{\partial u}{\partial t} - \frac{4}{\pi^2} \frac{\partial^2 u}{\partial x^2} = 0 \quad 0 < x < 4, \quad 0 < t;$$

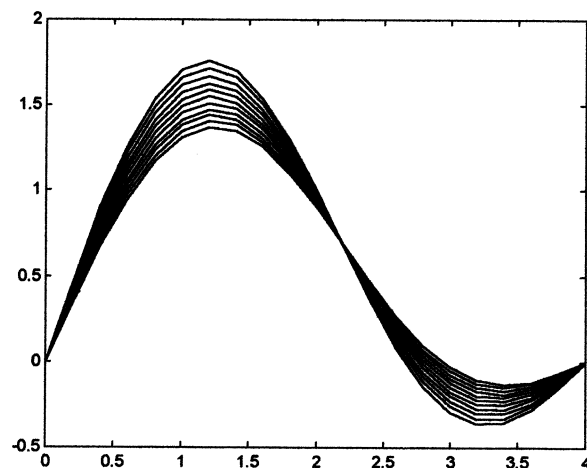
$$u(0,t) = u(4,t) = 0, \quad 0 < t;$$

$$u(x,0) = \operatorname{sen}\left(\frac{\pi}{4}x\right) \left(1 + 2 \cos\left(\frac{\pi}{4}x\right)\right), \quad 0 \leq x \leq 4.$$

Tomaremos $h=0.2$, o $n=20$; $T=0.4$, $N=10$ y $\alpha=2/\pi$.

x_i	$w_{i,j}$
0	0
0.2000000000000000	0.34938653512094
0.4000000000000000	0.67494510920999
0.6000000000000000	0.95492480449489
0.8000000000000000	1.17153176314066
1.0000000000000000	1.31243428224479
1.2000000000000000	1.37175148610249
1.4000000000000000	1.35043417634807
1.6000000000000000	1.25600537659915
1.8000000000000000	1.10169006583449
2.0000000000000000	0.90502258435950
2.2000000000000000	0.68607044325982
2.4000000000000000	0.46544987589961
2.6000000000000000	0.26232787805619
2.8000000000000000	0.09260581597745
3.0000000000000000	-0.03253906918964
3.2000000000000000	-0.10761390698439
3.4000000000000000	-0.13318149379700
3.6000000000000000	-0.11561039148955
3.8000000000000000	-0.06623308745373
4.0000000000000000	0

Y la gráfica que se obtiene:



8.3.- Código fuente del método Regresivo y de Crank-Nicolson.

- Fichero *regresivo.m*: Contiene el programa principal del algoritmo.

```

% ALGORITMO DEL METODO REGRESIVO.
% UTILIZAMOS PARA RESOLVER EL SISTEMA EL METODO DE CROUT.
format long;
clear;
clc;
p = input(' Escriba:   EL VALOR FRONTERA EN EL PUNTO
INICIAL: ');
X = input('           EL PUNTO FINAL Tn: ');
P = input('           EL VALOR FRONTERA EN ESE PUNTO
FINAL: ');
n = input('           EL NÚMERO DE PASOS: ');
T = input('           EL VALOR FINAL DEL TIEMPO DE
OBSERVACIÓN: ');
N = input('           EL NÚMERO DE DIVISIONES PARA EL
TIEMPO: ');
alfa = input('           EL VALOR DE ALFA: ');
F = 'f';

h = X/n;      % VALOR DEL PASO O INCREMENTO DE "x".
k = T/N;      % VALOR DEL PASO O INCREMENTO DEL TIEMPO.
w(1) = p;
w(n+1) = P;
landa = (alfa^2*k)/(h^2);
% PRIMEROS VALORES O DE PARTIDA PARA "w".
for i=2:n
    w(i) = feval(F,h*(i-1));
end
y = [0:h:X]; % PARA DIBUJAR LAS GRÁFICAS.

plot(y,w)    % PARA DIBUJAR LA GRÁFICA EN EL INSTANTE CERO.
pause;

% CREAMOS LA DIAGONAL INFERIOR Y PRINCIPAL DE LA MATRIZ
"A". ESTO ES NECESARIO PARA CROUT.
DP(1) = (1 + 2*landa);
for i=1:n-2
    DI(i) = -landa;
    DP(i+1) = (1 + 2*landa);
end
DP(n-1) = (1 + 2*landa);

% CÁLCULO DE LOS SUCESIVOS VALORES DE "w" EN CADA
INCREMENTO DEL TIEMPO.
hold on;
for j=1:N

```

```

    for i=1:n-1
        b(i) = w(i+1);
    end
    w = crout_cn(DI,DP,DI,b,[p P],n-1);
    plot(y,w)
    pause;
end
hold off;

j = 0;
for i=0:h:X
    j = j+1;
    M(j,:) = [i,w(j)];
end
M

```

- Fichero *Crank_n.m*: Contiene el programa principal del algoritmo.

```

% ALGORITMO DEL METODO CRANK-NICOLSON.
% UTILIZAMOS PARA RESOLVER EL SISTEMA EL METODO DE CROUT.
format long;
clear;
clc;
p = input('Escriba: EL VALOR FRONTERA EN EL PUNTO INICIAL: ');
X = input(' EL PUNTO FINAL Tn: ');
P = input(' EL VALOR FRONTERA EN ESE PUNTO FINAL: ');
n = input(' EL NÚMERO DE PASOS: ');
T = input(' EL VALOR FINAL DEL TIEMPO DE OBSERVACIÓN: ');
N = input(' EL NÚMERO DE DIVISIONES PARA EL TIEMPO: ');
alfa = input(' EL VALOR DE ALFA: ');
F = 'f';

h = X/n; % VALOR DEL PASO O INCREMENTO DE "x".
k = T/N; % VALOR DEL PASO O INCREMENTO DEL TIEMPO.
w(1) = p;
w(n+1) = P;
landa = ((alfa^2)*k)/(h^2);

% PRIMEROS VALORES O DE PARTIDA PARA "w".
for i=2:n
    w(i) = feval(F,h*(i-1));
end
y = [0:h:X]; %PARA DIBUJAR LAS GRÁFICAS.

plot(y,w); %DIBUJO DE LA PRIMERA GRÁFICA, O SEA, EN EL INSTANTE CERO.
pause

```

```

% CREAMOS LA DIAGONAL INFERIOR Y PRINCIPAL DE LA MATRIZ
"A". ESTO ES NECESARIO PARA CROUT.
% TENER EN CUENTA QUE EL ORDEN DE LA MATRIZ CUADRADA "A"
SIEMPRE ES n-1.
DP(1) = (1 + landa);
for i=1:n-2
    DI(i) = -(landa/2);
    DP(i+1) = (1 + landa);
end
DP(n-1) = (1 + landa);

% CÁLCULO DE LOS SUCESIVOS VALORES DE "w" EN CADA
INCREMENTO DEL TIEMPO.
hold on;
for j=1:N
    % PARA CADA ITERACIÓN EN EL TIEMPO, CALCULAMOS EL VECTOR
    "b".
    b(1) = (1 - landa)*w(2) + (landa/2)*w(3);
    for i=2:n-2
        b(i)=(landa/2)*w(i)+(1 -
landa)*w(i+1)+(landa/2)*w(i+2);
    end
    b(n-1)=(landa/2)*w(n-1) + (1 - landa)*w(n);
    w=croust_cn(DI,DP,DI,b,[p P],n-1);
    plot(y,w);
end
hold off;
j = 0;
for i=0:h:X
    j = j+1;
    M(j,:) = [i,w(j)];
end
M

```

♣ Los dos siguientes ficheros son comunes a los dos algoritmos:

- Fichero *croust_cn.m*: Resuelve el sistema con matriz tridiagonal que se obtiene.

%ALGORITMO DE CROUT PARA UTILIZARLO CON DIFERENCIAS FINITAS.

%PARÁMETROS:

```

% DI:  DIAGONAL INFERIOR.
% DP:  "      PRINCIPAL.
% DS:  "      SUPERIOR.
% lim: VALORES LÍMITES CONOCIDOS.
% n:   NÚMERO DE DIVISIONES.

```

```

function W = crout_cn(DI,DP,DS,b,lim,n);
L=0;
U=0;
Z=0;

L(1) = DP(1);
U(1) = DS(1)/DP(1);
for i=2:n-1
    L(i)=DP(i)-(DI(i-1)*U(i-1));
    U(i)=DS(i)/L(i);
end;

L(n)=DP(n)-(DI(n-1)*U(n-1));
Z(1)=b(1)/L(1);
for i=2:n
    Z(i)=(b(i)-DI(i-1)*Z(i-1))/L(i);
end;
W(1) = lim(1);
W(n+1) = Z(n);
W(n+2) = lim(2);
for i=n:-1:2
    W(i)=Z(i-1)-(U(i-1)*W(i+1));
end;

```

- Fichero *f.m*: Contiene la función $u(x, 0)$.

```

function y=f(x);
%y=2*sin(2*pi*x);
%y=x*(2-x);
y=sin(pi/4*x)*(1+2*cos(pi/4*x));

```

