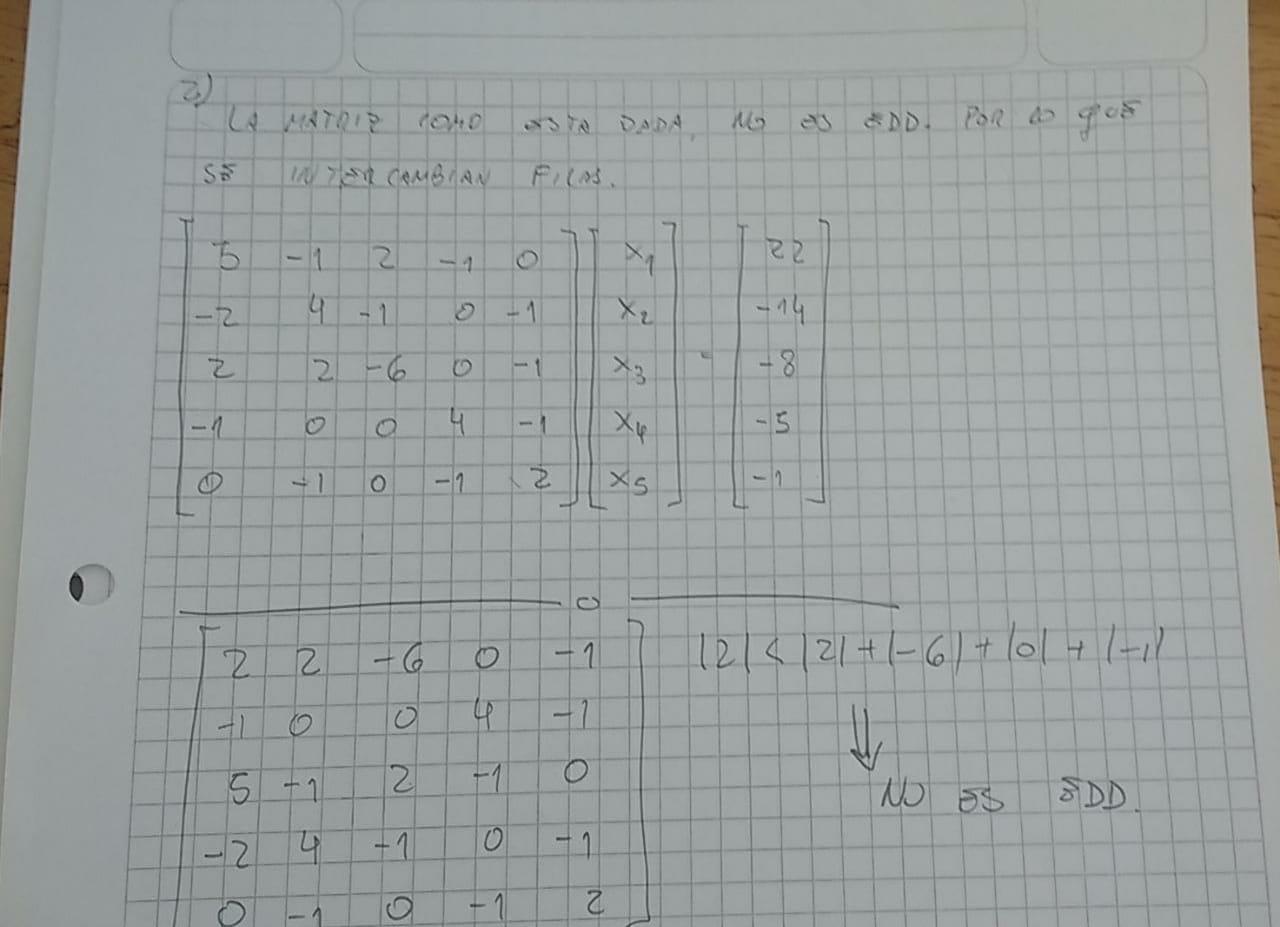
EJERCICIO 2 LEANDRO ZABALA

Primero se verifico que la matriz original sea EDD. Resulto no serlo.



Para evitar tener que calcular el radio espectral, decidí reacomodar las filas y encontré una forma en la que la matriz si quedo EDD.

Una vez hecho esto, se ingreso al programa de octave la matriz A

A =[ 5 -1 2 -1 0;

-2 4 -1 0 -1; % Esta es la matriz de coeficientes. Se cumple que

2 2 -6 0 -1;

-1 0 0 4 -1;

0 -1 0 -1 2] % A\*x=b

la matriz b también se tiene que acomodar, de acuerdo al cambio de orden de las filas.

b=[22; -14; -8; -5; -1] %esta es la matriz A\*x=b

se tomo como condiciones iniciales x=[1;1;1;1;1]

elegí el método de gauss\_seidel, ya que es muy eficaz para resolver sistemas de ecuaciones lineales, además, es simple de verificar la convergencia del método. También tiene la ventaja de que se puede armar la matriz B de forma vectorial, de esa forma, se puede usar el mismo algoritmo para sistemas de ecuaciones de cualquier orden.

pR el calculo del residuo se usa la norma 2

clear all;

clc;

% ingresar datos pertenecientes al problema

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%A =[-1 -7 3;

% 2 -1 8; % Esta es la matriz de coeficientes. Se cumple que

% 5 -2 1] % A\*x=b

A =[ 5 -1 2 -1 0;

-2 4 -1 0 -1; % Esta es la matriz de coeficientes. Se cumple que

2 2 -6 0 -1;

-1 0 0 4 -1;

0 -1 0 -1 2] % A\*x=b

b=[22; -14; -8; -5; -1] %esta es la matriz A\*x=b

x=[1;1;1;1;1] %valores iniciales de x

ciclomax=25

tolerancia=10^(-4)

% Se calcula la matriz B y c tal que X=matB\*C+c

% esta es la matriz de iteracion del metodo de jacobi tiene que tener ceros en la primir columna

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[a1,a2]=size(A); % Encuentro el tamaño de A

D = diag(diag(A)); % Encuentro D, L, U de forma que A=D+L+U

L = tril(A,-1); % D es la diagonal de A, L es triangular estrictamente inferior

U = triu(A,1); % U es diagonal estrictamente superior de A

matB = inv(D+L)\*(-U);

c = inv(D+L)\*b;

fprintf("\n la siguiente es la matriz B:\n" )

disp(matB)

fprintf("\n la siguiente es la matriz B con fracciones:\n" )

format rat

disp(matB)

format short

fprintf("\n la siguiente es la matriz c:\n" )

disp(c)

fprintf("\n la siguiente es la matriz c con fracciones:\n" )

format rat

disp(c)

format short

% Se verifica que el metodo converja.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

R= sum(abs(A), 2); %abs(A) calcula el valor absoluto de cada elemento de la matriz

% sum(abs(A), 2) suma los elementos de cada fila.

D=abs(diag(A)); % extrae la diagonal de la matriz y le calcula el valor absoluto a los elementos

W=R-D;

check= D>=W;

DD = all(check); % devuelve 1 si todos los elementos del vector son distintos de 0

% DD=0 si no es EDD, DD=1 si es EDD

normaB=norm(matB,inf); % se calcula la norma infinito de la matriz B (la que tiene 0 en la primer columna)

fprintf('\n\n'); %deja un renglon vacio

if DD==1

disp("la matriz es EDD. El metodo converje")

fprintf('la norma infinito de la matriz B (la de 0 en la 1er columna) es: ')

disp(normaB)

else

disp("la matriz NO es EDD")

fprintf('la norma infinito de la matriz B (la de 0 en la primer columna) es: ')

disp(normaB)

disp("Se procede a calcular el radio espectral")

respec=max(abs(eig(matB))); %se calcula el radio espectral de la matriz de coeficientes

fprintf('el radio espectral de la matriz es: ')

disp(respec)

if respec<1

fprintf('\n\n'); %deja un renglon vacio

disp("la matriz es valida. El metodo converje")

else

fprintf('\n\n'); %deja un renglon vacio

disp("la matriz NO es valida. El metodo diverje")

disp('probar intercambiando filas')

endif

endif

% Ejecucion del metodo numerico.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

n=0;

error1= 999;

fprintf('\n la solucion del sistema en la iteracion %4.0f',n)

fprintf(' (condiciones iniciales)\n')

for i=1:a1

fprintf(' x(%1.0f)=%6.8f\n',i,x(i))

end

fprintf('\ncalculo de error |X(n)-X(n-1)|')

fprintf('norma euclidea(2) -------- \n')

fprintf('norma suma(1) --------- \n')

fprintf('norma del maximo (INF) --------\n')

residuo=norm( A\*x-b, 2); % Se puede cambiar la norma del residuo.

% reemplazar por 1\_ norma suma;

% reemplazar por 2\_ norma euclidea

% reemplazar por inf norma infinito

fprintf('residuo A\*x-b %8.8g \n',residuo)

disp("--------------------------------------------")

while (error1>tolerancia)&&(n<ciclomax)

n=n+1;

xant=x;

x=matB\*x+c;

error1=norm(x-xant); % aca se puede cambiar la norma

errorsuma=norm(x-xant, 1);

errorinf=norm(x-xant, inf);

fprintf('\n la solucion del sistema en la iteracion %4.0f\n',n)

for i=1:a1

fprintf(' x(%1.0f)=%6.8f\n',i,x(i))

end

fprintf('\ncalculo de error |X(n)-X(n-1)|\n')

fprintf('norma euclidea(2) %8.8g \n',error1)

fprintf('norma suma(1) %8.8g \n',errorsuma)

fprintf('norma del maximo (INF) %8.8g \n',errorinf)

residuo=norm( A\*x-b, 2); % Se puede cambiar la norma del residuo.

% reemplazar por 1\_ norma suma;

% reemplazar por 2\_ norma euclidea

% reemplazar por inf norma infinito

fprintf('residuo (A\*x-b) %8.8g \n',residuo)

disp("--------------------------------------------")

end

A =

5 -1 2 -1 0

-2 4 -1 0 -1

2 2 -6 0 -1

-1 0 0 4 -1

0 -1 0 -1 2

b =

22

-14

-8

-5

-1

x =

1

1

1

1

1

ciclomax = 25

tolerancia = 1.0000e-04

la siguiente es la matriz B:

0 0.2000 -0.4000 0.2000 0

0 0.1000 0.0500 0.1000 0.2500

0 0.1000 -0.1167 0.1000 -0.0833

0 0.0500 -0.1000 0.0500 0.2500

0 0.0750 -0.0250 0.0750 0.2500

la siguiente es la matriz B con fracciones:

0 1/5 -2/5 1/5 0

0 1/10 1/20 1/10 1/4

0 1/10 -7/60 1/10 -1/12

0 1/20 -1/10 1/20 1/4

0 3/40 -1/40 3/40 1/4

la siguiente es la matriz c:

4.4000

-1.3000

2.3667

-0.1500

-1.2250

la siguiente es la matriz c con fracciones:

22/5

-13/10

71/30

-3/20

-49/40

la matriz es EDD. El metodo converje

la norma infinito de la matriz B (la de 0 en la 1er columna) es: 0.8000

la solucion del sistema en la iteracion 0 (condiciones iniciales)

x(1)=1.00000000

x(2)=1.00000000

x(3)=1.00000000

x(4)=1.00000000

x(5)=1.00000000

calculo de error |X(n)-X(n-1)|norma euclidea(2) --------

norma suma(1) ---------

norma del maximo (INF) --------

residuo A\*x-b 23.664319

--------------------------------------------

la solucion del sistema en la iteracion 1

x(1)=4.40000000

x(2)=-0.80000000

x(3)=2.36666667

x(4)=0.10000000

x(5)=-0.85000000

calculo de error |X(n)-X(n-1)|

norma euclidea(2) 4.5716822

norma suma(1) 9.3166667

norma del maximo (INF) 3.4

residuo (A\*x-b) 6.0497704

--------------------------------------------

la solucion del sistema en la iteracion 2

x(1)=3.31333333

x(2)=-1.46416667

x(3)=2.09138889

x(4)=-0.63416667

x(5)=-1.54916667

calculo de error |X(n)-X(n-1)|

norma euclidea(2) 1.6509314

norma suma(1) 3.4594444

norma del maximo (INF) 1.0866667

residuo (A\*x-b) 1.6266337

--------------------------------------------

la solucion del sistema en la iteracion 3

x(1)=3.14377778

x(2)=-1.79255556

x(3)=2.04193519

x(4)=-0.85134722

x(5)=-1.82195139

calculo de error |X(n)-X(n-1)|

norma euclidea(2) 0.51050261

norma suma(1) 1.0373634

norma del maximo (INF) 0.32838889

residuo (A\*x-b) 0.67243408

--------------------------------------------

la solucion del sistema en la iteracion 4

x(1)=3.05444537

x(2)=-1.91778137

x(3)=2.01587990

x(4)=-0.94187650

x(5)=-1.92982894

calculo de error |X(n)-X(n-1)|

norma euclidea(2) 0.21017558

norma suma(1) 0.43902033

norma del maximo (INF) 0.12522581

residuo (A\*x-b) 0.26075403

--------------------------------------------

la solucion del sistema en la iteracion 5

x(1)=3.02171647

x(2)=-1.96762903

x(3)=2.00633397

x(4)=-0.97702812

x(5)=-1.97232857

calculo de error |X(n)-X(n-1)|

norma euclidea(2) 0.081786002

norma suma(1) 0.16977374

norma del maximo (INF) 0.04984766

residuo (A\*x-b) 0.10327132

--------------------------------------------

la solucion del sistema en la iteracion 6

x(1)=3.00853498

x(2)=-1.98723116

x(3)=2.00248937

x(4)=-0.99094840

x(5)=-1.98908978

calculo de error |X(n)-X(n-1)|

norma euclidea(2) 0.032364923

norma suma(1) 0.0673097

norma del maximo (INF) 0.019602133

residuo (A\*x-b) 0.040667312

--------------------------------------------

la solucion del sistema en la iteracion 7

x(1)=3.00336834

x(2)=-1.99496593

x(3)=2.00098243

x(4)=-0.99643036

x(5)=-1.99569815

calculo de error |X(n)-X(n-1)|

norma euclidea(2) 0.012748106

norma suma(1) 0.026498683

norma del maximo (INF) 0.0077347728

residuo (A\*x-b) 0.016040498

--------------------------------------------

la solucion del sistema en la iteracion 8

x(1)=3.00132777

x(2)=-1.99801504

x(3)=2.00038727

x(4)=-0.99859259

x(5)=-1.99830382

calculo de error |X(n)-X(n-1)|

norma euclidea(2) 0.0050279258

norma suma(1) 0.01045276

norma del maximo (INF) 0.0030491123

residuo (A\*x-b) 0.0063240018

--------------------------------------------

la solucion del sistema en la iteracion 9

x(1)=3.00052357

x(2)=-1.99921736

x(3)=2.00015271

x(4)=-0.99944506

x(5)=-1.99933121

calculo de error |X(n)-X(n-1)|

norma euclidea(2) 0.0019823085

norma suma(1) 0.0041209326

norma del maximo (INF) 0.0012023115

residuo (A\*x-b) 0.0024935723

--------------------------------------------

la solucion del sistema en la iteracion 10

x(1)=3.00020643

x(2)=-1.99969141

x(3)=2.00006021

x(4)=-0.99978119

x(5)=-1.99973630

calculo de error |X(n)-X(n-1)|

norma euclidea(2) 0.00078162583

norma suma(1) 0.0016249058

norma del maximo (INF) 0.00047405357

residuo (A\*x-b) 0.00098318722

--------------------------------------------

la solucion del sistema en la iteracion 11

x(1)=3.00008140

x(2)=-1.99987833

x(3)=2.00002374

x(4)=-0.99991373

x(5)=-1.99989603

calculo de error |X(n)-X(n-1)|

norma euclidea(2) 0.00030818665

norma suma(1) 0.00064068078

norma del maximo (INF) 0.00018691631

residuo (A\*x-b) 0.0003876635

--------------------------------------------

la solucion del sistema en la iteracion 12

x(1)=3.00003209

x(2)=-1.99995202

x(3)=2.00000936

x(4)=-0.99996598

x(5)=-1.99995900

calculo de error |X(n)-X(n-1)|

norma euclidea(2) 0.00012151568

norma suma(1) 0.00025261585

norma del maximo (INF) 7.3699464e-05

residuo (A\*x-b) 0.00015285243

--------------------------------------------

la solucion del sistema en la iteracion 13

x(1)=3.00001265

x(2)=-1.99998108

x(3)=2.00000369

x(4)=-0.99998659

x(5)=-1.99998384

calculo de error |X(n)-X(n-1)|

norma euclidea(2) 4.7912609e-05

norma suma(1) 9.9604275e-05

norma del maximo (INF) 2.9059103e-05

residuo (A\*x-b) 6.0268467e-05

--------------------------------------------

>>

b) como método directo, elegí la factorización LU, ya que tiene la ventaja de que permite reservar los parámetros de la eliminación de gauss, para ser aplicados en la resolución de sistemas con igual matriz A

>> A=[2 2 -6 0 -1; -1 0 0 4 -1; 5 -1 2 -1 0; -2 4 -1 0 -1; 0 -1 0 -1 2]

A =

2 2 -6 0 -1

-1 0 0 4 -1

5 -1 2 -1 0

-2 4 -1 0 -1

0 -1 0 -1 2

>> b=[-8; -5; 22; -14; -1]

b =

-8

-5

22

-14

-1

>> [L,U,P]=lu(A)

L =

1.0000 0 0 0 0

-0.4000 1.0000 0 0 0

0.4000 0.6667 1.0000 0 0

-0.2000 -0.0556 -0.0583 1.0000 0

0 -0.2778 0.0083 -0.2926 1.0000

U =

5.0000 -1.0000 2.0000 -1.0000 0

0 3.6000 -0.2000 -0.4000 -1.0000

0 0 -6.6667 0.6667 -0.3333

0 0 0 3.8167 -1.0750

0 0 0 0 1.4105

P =

Permutation Matrix

0 0 1 0 0

0 0 0 1 0

1 0 0 0 0

0 1 0 0 0

0 0 0 0 1

>>

>> X=U\(L\b)

X =

-1.6440

-4.6517

-4.8545

-5.2771

-5.4644

>> A\*X

ans =

22.0000

-14.0000

-8.0000

-5.0000

-1.0000

>>residuo=A\*X-b

residuo =

3.0000e+01

-9.0000e+00

-3.0000e+01

9.0000e+00

1.7764e-15

c) si se puede considerar el calentamiento como una función lineal, entonces lo mas inicado es buscar un polinomio de grado 1 (una recta) que se aproxime a los puntos.

Para calcular el polinomio aproximante de primer grado uso el siguiente programa:

clc

pkg load symbolic

X1=[0 2 5 8 10 14 16]

Y1=[25 100 150 200 250 270 350]

Xe=[12];

function []=Graficos(X1,Y1,Xe)

if length(X1)>3

[Rlagrange,polinomioL,XXlagrange,YYlagrange]=lagrange(X1,Y1,Xe);

[Rnewton,polinomioN,DD,XXnewton,YYnewton] = newton(X1, Y1,Xe);

[Rminimos,polinomioM,coeficientes,matvar,XXmin,YYmin] = mincuad(X1,Y1,Xe);

[Rspline,PP,polinomioS,XXspline,YYspline]=sspline(X1, Y1,Xe);

hold on;

plot(XXlagrange,YYlagrange);

plot(XXnewton,YYnewton);

plot(XXspline,YYspline);

plot(XXmin, YYmin);

plot(X1,Y1,'.r','markersize',20);

title('distintas aproximaciones e interpolaciones')

legend({'lagrange','newton','spline','mincuad 1'},'Location','southwest')

else

[Rlagrange,polinomioL,XXlagrange,YYlagrange]=lagrange(X1,Y1,Xe);

[Rnewton,polinomioN,DD,XXnewton,YYnewton] = newton(X1, Y1,Xe);

[Rminimos,polinomioM,coeficientes,matvar,XXmin,YYmin] = mincuad(X1,Y1,Xe);

hold on;

graficoL = plot(XXlagrange,YYlagrange);

set(graficoL,'Color','red','LineWidth',1)

legend(graficoL,'Lagrange')

graficoN = plot(XXnewton,YYnewton);

set(graficoN,'Color','blue','LineWidth',1)

legend(graficoN,'Newton')

for i=1:length(X1)-1

hold on;

subplot(2,2,4)

graficoM = plot(XXmin,YYmin(i,:));

set(graficoL,'Color','red','LineWidth',1)

legend(graficoM,'Min. Cuad.')

endfor

hold off;

clc

endif

endfunction

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function []=Tabla(X1,Y1,Xe)

Rlagrange=lagrange(X1, Y1,Xe);

Rspline=linspace(0,0,length(Xe));

Rnewton=newton(X1, Y1,Xe);

Rminimos=mincuad(X1,Y1,Xe);

if length(X1)>3

Rspline=sspline(X1,Y1,Xe);

endif

clc

disp("Tabla de aproximaciones: \n")

titulos=[" Valores" " Lagrange" " Spline" " Newton" " Min. Cuad."];

tabla(:,1)=Xe;

tabla(:,2)=Rlagrange;

tabla(:,3)=Rspline;

tabla(:,4)=Rnewton;

tabla(:,5)=Rminimos;

disp(titulos)

disp(tabla)

endfunction

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [Rlagrange,polinomioL,XXlagrange,YYlagrange]=lagrange(X1,Y1,Xe)

syms x

for i=1:length(X1)

T=1;

for k=1:length(X1)

if(k~=i)

T=T.\*(x-X1(k));

endif

end

b=char(T);

f=inline(b);

n=feval(f,X1(i));

T=T/n;

F(i)=T;

end

for i=1:length(Xe)

p(i)=0;

for k=1:length(X1)

b=char(F(k));

f=inline(b);

L(k)=feval(f,Xe(i));

p(i)=p(i)+(Y1(k)\*L(k));

end

end

polinomioL=0;

for i=1:length(X1)

G(i)=(F(i).\*Y1(i));

polinomioL=polinomioL+G(i);

end

clc

b=char(polinomioL);

f=inline(b);

XXlagrange=linspace(X1(1), X1(length(X1)));

for i=1:length(XXlagrange)

YYlagrange(i)=f(XXlagrange(i));

end

disp('Polinomio armado')

disp(polinomioL)

disp('Polinomio armado simplificado')

disp(simplify(polinomioL))

disp('Resultados de aproximaciones Xe')

Rlagrange=transpose(p);

endfunction

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [Rspline,PP,polinomioS,XXspline,YYspline]=sspline(X1, Y1,Xe)

syms x

syms Xk

n=length(X1)-1;

PP=spline(X1, Y1);

XXspline=linspace(X1(1), X1(length(X1)));

YYspline=ppval(PP, XXspline);

for i=1:n

for k=2:4

t=PP.coefs(i,k)\*((x-Xk)^(k-1));

polinomioS(i,k)=t;

endfor

polinomioS(i,1)=PP.coefs(i,1);

endfor

for i=1:length(Xe)

Rspline(i,1)=ppval(PP, Xe(i));

endfor

clc

disp('Coeficientes como float de grado en orden ascendente ')

disp(PP.coefs)

disp('Polinomios armados')

disp(polinomioS)

disp('Resultados de aproximaciones Xe')

endfunction

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [Rnewton,polinomioN,DD,XXnewton,YYnewton] = newton(X1, Y1,Xe)

syms x;

n=length(X1)-1;

DD=zeros(n+1); %creo una matriz cuadrada de ceros

DD(:,1)=Y1; %lleno la primer columna de DD con los elementos de Y

% calculo la matriz de diferencias divididas

for k=2:n+1

for J=k:n+1

DD(J,k)=[DD(J,k-1)-DD(J-1,k-1)]/[X1(J)-X1(J-k+1)];

end

end

polnew=DD(1,1);

P=1;

for i=1:n

P=P\*(x-X1(i));

polnew=polnew+P\*DD(i+1,i+1);

end

polnew=expand(polnew);

pretty(polnew);

polinomioN=vpa(polnew, 5);

b=char(polinomioN);

f=inline(b);

XXnewton=linspace(X1(1), X1(n+1));

for i=1:length(XXnewton)

YYnewton(i)=f(XXnewton(i));

end

for i=1:length(Xe)

Rnewton(i)=f(Xe(i));

endfor

clc

disp('La matriz de diferencias divididas es:');

disp(DD);

disp('El polinomio de newton es');

disp(polinomioN)

disp('Resultados de aproximaciones Xe')

endfunction

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [Rminimos,polinomioM,coeficientes,matvar,XXmin,YYmin] = mincuad(X1,Y1,Xe)

syms x

m=length(X1);

n=m-1;

CM=zeros(n,m+1);

matvar=zeros(n,2);

XXmin=linspace(X1(1,1), X1(1,m));

m2=length(XXmin);

YYmin=zeros(n,m2);

for i=1:n

sum=0;

P=polyfit(X1,Y1,i);

for i2=1:i+1

zeros(i,i2)=P(1,i2);

endfor

for i3=1:m

diferencia=((Y1(1,i3)-polyval(P,X1(1,i3)))^2);

sum= sum+diferencia;

endfor

varianza=sum/(m-i-1);

matvar(i,1)=i;

matvar(i,2)=varianza;

YYmin(i,:)=polyval(P,XXmin);

endfor

polinomioM=0;

for i=1:n

polinomioM=polinomioM+P(i)\*x^(i);

endfor

polinomioM=polinomioM+P(length(P));

Rminimos(:,1)=polyval(P,Xe);

clc

disp('ordenados de forma descendente. ax+b en el segundo renglon ax^2+b\*x+c')

disp('el ultimo polinomio es de interpolacion')

coeficientes=zeros

for i=1:n

fprintf("\n varianzadel polinomio de orden %1.0g : %5.6g \n", matvar(i,1), matvar(i,2) )

endfor

matvar

disp('Polinomio interpolacion armado')

disp(polinomioM)

disp('Resultados de aproximaciones Xe')

endfunction

por consola llamo a la función de minimos cuadrados:

[Rminimos,polinomioM,coeficientes,matvar,XXmin,YYmin] = mincuad(X1,Y1,Xe)

Me devuelve los siguientes datos:

ordenados de forma descendente. ax+b en el segundo renglon ax^2+b\*x+c

el ultimo polinomio es de interpolacion

coeficientes =

18.1443 49.5805 0 0 0 0 0

-0.3327 23.4861 38.2693 0 0 0 0

0.1165 -3.1237 39.8123 25.9099 0 0 0

0.0124 -0.2755 0.7057 28.0702 29.9553 0 0

0.0069 -0.2655 3.6380 -21.4071 69.1592 24.4122 0

0.0011 -0.0460 0.6602 -3.7237 4.3300 39.1530 25.0000

varianzadel polinomio de orden 1 : 503.399

varianzadel polinomio de orden 2 : 507.481

varianzadel polinomio de orden 3 : 394.26

varianzadel polinomio de orden 4 : 496.313

varianzadel polinomio de orden 5 : 117.905

varianzadel polinomio de orden 6 : Inf

matvar =

1.0000 503.3993

2.0000 507.4807

3.0000 394.2605

4.0000 496.3132

5.0000 117.9051

6.0000 Inf

Polinomio interpolacion armado

6 5 4 3 2

3851\*pi\*x 1625\*pi\*x 435\*pi\*x 402\*pi\*x 442\*x 19\*x

---------- + ---------- - --------- + --------- - ------ + ----- + 25

309 1179 367 1913 9615 16974

Resultados de aproximaciones Xe

Rminimos = 276.90

polinomioM = (sym)

6 5 4 3 2

3851\*pi\*x 1625\*pi\*x 435\*pi\*x 402\*pi\*x 442\*x 19\*x

---------- + ---------- - --------- + --------- - ------ + ----- + 25

309 1179 367 1913 9615 16974

coeficientes =

18.1443 49.5805 0 0 0 0 0

-0.3327 23.4861 38.2693 0 0 0 0

0.1165 -3.1237 39.8123 25.9099 0 0 0

0.0124 -0.2755 0.7057 28.0702 29.9553 0 0

0.0069 -0.2655 3.6380 -21.4071 69.1592 24.4122 0

0.0011 -0.0460 0.6602 -3.7237 4.3300 39.1530 25.0000

matvar =

1.0000 503.3993

2.0000 507.4807

3.0000 394.2605

4.0000 496.3132

5.0000 117.9051

6.0000 Inf

XXmin =

Columns 1 through 16:

0 0.1616 0.3232 0.4848 0.6465 0.8081 0.9697 1.1313 1.2929 1.4545 1.6162 1.7778 1.9394 2.1010 2.2626 2.4242

Columns 17 through 32:

2.5859 2.7475 2.9091 3.0707 3.2323 3.3939 3.5556 3.7172 3.8788 4.0404 4.2020 4.3636 4.5253 4.6869 4.8485 5.0101

Columns 33 through 48:

5.1717 5.3333 5.4949 5.6566 5.8182 5.9798 6.1414 6.3030 6.4646 6.6263 6.7879 6.9495 7.1111 7.2727 7.4343 7.5960

Columns 49 through 64:

7.7576 7.9192 8.0808 8.2424 8.4040 8.5657 8.7273 8.8889 9.0505 9.2121 9.3737 9.5354 9.6970 9.8586 10.0202 10.1818

Columns 65 through 80:

10.3434 10.5051 10.6667 10.8283 10.9899 11.1515 11.3131 11.4747 11.6364 11.7980 11.9596 12.1212 12.2828 12.4444 12.6061 12.7677

Columns 81 through 96:

12.9293 13.0909 13.2525 13.4141 13.5758 13.7374 13.8990 14.0606 14.2222 14.3838 14.5455 14.7071 14.8687 15.0303 15.1919 15.3535

Columns 97 through 100:

15.5152 15.6768 15.8384 16.0000

YYmin =

Columns 1 through 16:

49.581 52.513 55.445 58.378 61.310 64.243 67.175 70.107 73.040 75.972 78.905 81.837 84.769 87.702 90.634 93.567

38.269 42.056 45.826 49.578 53.313 57.031 60.731 64.414 68.079 71.727 75.358 78.971 82.567 86.145 89.706 93.250

25.910 32.263 38.456 44.492 50.373 56.103 61.685 67.121 72.414 77.568 82.586 87.470 92.223 96.848 101.348 105.727

29.955 34.509 39.093 43.700 48.324 52.959 57.598 62.236 66.867 71.485 76.086 80.664 85.215 89.733 94.214 98.655

24.412 35.045 44.650 53.312 61.112 68.129 74.435 80.100 85.190 89.768 93.891 97.615 100.991 104.067 106.888 109.497

25.000 31.426 37.989 44.612 51.225 57.767 64.188 70.443 76.496 82.318 87.885 93.182 98.195 102.918 107.350 111.491

Columns 17 through 32:

96.499 99.432 102.364 105.296 108.229 111.161 114.094 117.026 119.958 122.891 125.823 128.756 131.688 134.620 137.553 140.485

96.776 100.285 103.777 107.251 110.708 114.147 117.569 120.974 124.361 127.731 131.083 134.418 137.736 141.036 144.319 147.585

109.986 114.130 118.161 122.081 125.895 129.604 133.213 136.723 140.137 143.459 146.692 149.839 152.902 155.884 158.789 161.619

103.051 107.398 111.692 115.931 120.111 124.229 128.282 132.268 136.183 140.027 143.797 147.490 151.107 154.645 158.102 161.479

111.930 114.225 116.415 118.528 120.592 122.632 124.671 126.727 128.818 130.959 133.163 135.441 137.801 140.251 142.796 145.439

115.349 118.932 122.252 125.324 128.163 130.789 133.222 135.483 137.595 139.579 141.459 143.259 145.002 146.709 148.404 150.107

Columns 33 through 48:

143.418 146.350 149.283 152.215 155.147 158.080 161.012 163.945 166.877 169.809 172.742 175.674 178.607 181.539 184.471 187.404

150.833 154.064 157.277 160.473 163.652 166.813 169.957 173.084 176.193 179.285 182.359 185.416 188.455 191.478 194.482 197.470

164.378 167.067 169.691 172.253 174.754 177.198 179.589 181.928 184.219 186.465 188.669 190.834 192.963 195.058 197.123 199.160

164.774 167.987 171.117 174.164 177.128 180.010 182.809 185.528 188.165 190.723 193.203 195.605 197.933 200.187 202.370 204.485

148.182 151.026 153.968 157.007 160.138 163.358 166.659 170.035 173.477 176.978 180.528 184.117 187.733 191.367 195.006 198.639

151.839 153.620 155.467 157.396 159.423 161.561 163.820 166.212 168.743 171.420 174.245 177.220 180.346 183.618 187.034 190.586

Columns 49 through 64:

190.336 193.269 196.201 199.134 202.066 204.998 207.931 210.863 213.796 216.728 219.660 222.593 225.525 228.458 231.390 234.322

200.440 203.393 206.328 209.246 212.146 215.030 217.895 220.744 223.575 226.388 229.185 231.963 234.725 237.469 240.196 242.905

201.173 203.165 205.137 207.095 209.039 210.974 212.902 214.827 216.750 218.676 220.607 222.545 224.495 226.458 228.439 230.439

206.532 208.516 210.439 212.304 214.114 215.873 217.585 219.254 220.883 222.477 224.041 225.579 227.096 228.597 230.087 231.573

202.255 205.841 209.386 212.877 216.304 219.655 222.919 226.085 229.144 232.087 234.905 237.590 240.135 242.536 244.787 246.885

194.265 198.062 201.963 205.955 210.022 214.146 218.310 222.493 226.674 230.831 234.942 238.983 242.931 246.762 250.453 253.981

Columns 65 through 80:

237.255 240.187 243.120 246.052 248.985 251.917 254.849 257.782 260.714 263.647 266.579 269.511 272.444 275.376 278.309 281.241

245.597 248.272 250.929 253.569 256.191 258.796 261.384 263.954 266.507 269.042 271.560 274.061 276.544 279.010 281.459 283.890

232.462 234.511 236.588 238.697 240.841 243.022 245.243 247.508 249.819 252.180 254.593 257.061 259.588 262.175 264.827 267.545

233.060 234.553 236.059 237.584 239.136 240.719 242.342 244.012 245.736 247.521 249.375 251.307 253.324 255.435 257.649 259.974

248.827 250.614 252.245 253.723 255.052 256.238 257.287 258.210 259.018 259.724 260.344 260.896 261.401 261.880 262.360 262.869

257.324 260.460 263.369 266.034 268.437 270.565 272.406 273.951 275.196 276.140 276.785 277.139 277.215 277.032 276.614 275.992

Columns 81 through 96:

284.173 287.106 290.038 292.971 295.903 298.836 301.768 304.700 307.633 310.565 313.498 316.430 319.362 322.295 325.227 328.160

286.304 288.700 291.080 293.441 295.785 298.112 300.422 302.714 304.989 307.246 309.486 311.709 313.914 316.102 318.272 320.425

270.334 273.195 276.133 279.149 282.247 285.430 288.700 292.062 295.516 299.068 302.718 306.472 310.330 314.297 318.376 322.568

262.418 264.992 267.705 270.567 273.586 276.773 280.139 283.694 287.448 291.412 295.597 300.015 304.676 309.593 314.778 320.241

263.438 264.099 264.890 265.850 267.022 268.451 270.187 272.283 274.793 277.778 281.301 285.429 290.232 295.785 302.167 309.461

275.205 274.297 273.323 272.346 271.436 270.676 270.157 269.982 270.266 271.135 272.729 275.201 278.719 283.465 289.637 297.449

Columns 97 through 100:

331.092 334.024 336.957 339.889

322.561 324.679 326.780 328.864

326.878 331.308 335.861 340.540

325.997 332.056 338.432 345.137

317.753 327.135 337.702 349.555

307.132 318.937 333.131 350.000

>>

La matriz coeficientes tiene los coeficientes de los polinomios de aproximación de distintos grados. Me quedo con la primera fila, siendo el polinomio d aproximación lineal:

P1(x)=18.1443\*x+49.5805

La velocidad promedio de calentamiento seria la pendiente de esta recta, por lo tanto, seria 18.1443 grados por minuto.

Si especializamos este polinomio en tiempo=0min obtenemos que la temperatura es de 49.5805 grados.

La diferencia con el valor experimental es de 24.5808 grados, esta diferencia si produjo porque se aproximaron los datos tomados a una recta.

d) El programa antes mencionado, también calcula el polinomio de interpolación.

Se toma la ultima fila de la matriz de coeficientes:

0.0011 -0.0460 0.6602 -3.7237 4.3300 39.1530 25.0000

Esos son los coeficientes del polinomio de interpolación, ordenados de mayor a menor, por lo que quedaría:

P(x)= 0.0011\*x^6 -0.0460\*x^5 + 0.6602\*x^4 -3.7237\*x^3+ 4.3300\*x^2+ 39.1530\*x^+ 25.0000

e) si, seria el mismo polinomio, ya que hay un único polinomio de interpolación, pero varios métodos para calcularlo.