Contenido

% ELIMINACION GAUSSIANA SIMPLE

```
format long %permite utilizar la maxima capacidad de la maquina

fprintf('ELIMINACION GAUSSIANA SIMPLE');

A=input('Ingrese la matriz A = \n');
b=input('\nIngrese el vector b, correspondite a los terminos independientes b=\n');
% Las matrices A y b deben ser ingresadas entre corchetes separando las
%columnas mediante coma ',' y las filas mediante punto y coma ';'.

%permite obtener el tamaño de la matriz A
[n,m]=size(A);
% la matriz C, representa la forma de la matriz aumentada [Ab]
C=[A,b];

fprintf('\nLa Matriz C, que corresponte a la matriz aumentada [Ab] es = \n');
disp(C); % la funcion disp nos permite imprimir una variable en el espacio de trabajo

if n==m
   for k=1:(n-1)%instruccion iterativa que permite repetir pasos un numero
                %determinado de veces
       fprintf('\n ETAPA %g=\n\n',k)
       fprintf('\nLa matriz correspondiente a esta etapa antes del proceso:\n')
       disp(C)
       fprintf('\nLos Multiplicadores correpondientes a esta etapa son:\n')
      for i=(k+1):n
         m(i,k)=C(i,k)/C(k,k); %formula para hallar los multiplicadores
         fprintf('\nm(%g,%g)=',i,k)
         disp(m(i,k));
         for j=k:(n+1)
            C(i,j)= C(i,j) - m(i,k)*C(k,j); %formula de la nueva fila
         end
      end
      fprintf('\nLa matriz correspondiente a esta etapa despues del proceso:\n')
       disp(C)
   end
   for i=n:-1:1
      suma=0;
      for p=(i+1):n
         suma = suma + C(i,p)*X(p);
      end
      X(i)=(C(i,n+1)-suma)/C(i,i);
      %formula de la susticion regresiva y solucion de las variables
   end
else %funcion asignada del if, en caso de que este sea falso
   fprintf('\nERROR: La matriz NO es cuadrada\n');
end
```

```
fprintf('\n\n\nSOLUCION:\n');
fprintf('\n\nLa matriz Ab final:\n');
disp(C)
fprintf('\n\nLa solucion de X1 hasta Xn es:\n');

%Se muestran los resultados de todo el proceso
for i=1:n
   Xi=X(1,i);
   fprintf('\nX%g=',i)
   disp(Xi);
end
```

ELIMINACION GAUSSIANA SIMPLE TEST

ELIMINACION GAUSSIANA SIMPLE
Ingrese la matriz A =
[4 3 -2 -7;3 12 8 -3; 2 3 -9 3;1 -2 -5 6]

Ingrese el vector b, correspondite a los terminos independientes b=
[20;18;31;12]

La Matriz C, que corresponte a la matriz aumentada [Ab] es =

4	3	-2	-7	20
3	12	8	-3	18
2	3	-9	3	31
1	-2	-5	6	12

ETAPA 1=

La matriz correspondiente a esta etapa antes del proceso:

4	3	-2	-7	20
3	12	8	-3	18
2	3	-9	3	31
1	-2	-5	6	12

Los Multiplicadores correpondientes a esta etapa son:

m(2,1)= 0.750000000000000

m(3,1)= 0.500000000000000

m(4,1)= 0.250000000000000

La matriz correspondiente a esta etapa despues del proceso:

4.000000000000000	3.000000000000000	-2.000000000000000	-7.000000000000000
20.000000000000000			
0	9.750000000000000	9.500000000000000	2.250000000000000
3.000000000000000			
0	1.500000000000000	-8.000000000000000	6.500000000000000
21.000000000000000			
0	-2.750000000000000	-4.500000000000000	7.750000000000000
7.000000000000000			

ETAPA 2=

La matriz correspondiente a esta etapa antes del proceso:

4.000000000000000 3.000000000000000 -2.000000000000000 -7.000000000000000 20.000000000000000
0 9.750000000000000 9.500000000000000 2.250000000000000 3.000000000000000
0 1.500000000000000 -8.000000000000000 6.500000000000000 21.000000000000000
0 -2.750000000000000 -4.500000000000000 7.750000000000000 7.000000000000000

Los Multiplicadores correpondientes a esta etapa son:

m(3,2)= 0.153846153846154

m(4,2)= -0.282051282051282

La matriz correspondiente a esta etapa despues del proceso:

4.000000000000000 3.000000000000000 -2.000000000000000 -7.000000000000000 20.000000000000000
0 9.750000000000000 9.500000000000000 2.250000000000000 3.000000000000000
0 0 -9.461538461538462 6.153846153846154 20.538461538461540
0 0 -1.820512820512821 8.384615384615385 7.846153846153846

ETAPA 3=

La matriz correspondiente a esta etapa antes del proceso:

4.000000000000000 3.000000000000000 -2.000000000000000 -7.000000000000000 20.000000000000000
0 9.750000000000000 9.500000000000000 2.250000000000000 3.000000000000000
0 0 -9.461538461538462 6.153846153846154 20.538461538461540
0 0 -1.820512820512821 8.384615384615385 7.846153846153846

Los Multiplicadores correpondientes a esta etapa son:

m(4,3)= 0.192411924119241

La matriz correspondiente a esta etapa despues del proceso:

4.000000000000000 3.000000000000000 -2.000000000000000 -7.000000000000000 20.000000000000000

0 9.750000000000000 9.500000000000000 2.250000000000000 3.000000000000000

0 0 -9.461538461538462 6.153846153846154 20.538461538461540

0 0 0 7.200542005420054 3.894308943089430

SOLUCION:

La matriz Ab final:

4.000000000000000 3.000000000000000 -2.000000000000000 -7.000000000000000 20.000000000000000

0 9.750000000000000 9.500000000000000 2.250000000000000 3.000000000000000

0 0 -9.461538461538462 6.153846153846154 20.538461538461540

0 0 0 7.200542005420054 3.894308943089430

La solucion de X1 hasta Xn es:

X1= 3.570568310124200

X2= 1.955212645841174

X3= -1.818968761761385

X4= 0.540835528791870

% ELIMINACION GAUSSIANA CON PIVOTEO PARCIAL

```
format long %permite utilizar la maxima capacidad de la maquina

fprintf('ELIMINACION GAUSSIANA CON PIVOTEO PARCIAL (SOLUCION POR ETAPAS)\n\n\n');

A=input('Ingrese la matriz A = \n');
b=input('\nIngrese el vector b, correspondite a los terminos independientes b=\n');
% Las matrices A y b deben ser ingresadas entre corchetes separando las
%columnas mediante coma ',' y las filas mediante punto y coma ';'.

%permite obtener el tamaño de la matriz A
[n,m]=size(A);
% la matriz C, representa la forma de la matriz aumentada [Ab]
C=[A,b];


fprintf('\nLa Matriz C, que corresponte a la matriz aumentada [Ab] es = \n');
disp(C);  % la funcion disp nos permite imprimir una variable en el espacio de trabajo

if n==m %compara el numero de columnas y filas, para observar si son iguales
   for k=1:(n-1)
      fprintf('\n ETAPA %g=\n\n',k)
      mayor=0; %asigna como cero el numero mayor de la fila
      filam=k; %asigna la fila k como la columna que tiene el numero mayor
      for p=k:n
         if mayor<abs(C(p,k)) %se busca el numero mayor en la fila K;
            mayor=abs(C(p,k));%cambio de mayor
            filam=p; %cambio de fila
         end
      end
      if mayor ==0
         fprintf('\nEl sistema tiene infinitas soluciones\n')
         break %se interrumpe el programa con la instruccion break, ya que
               %si mayor=o, mas adelante se obtiene una division por
               %cero
      else
         if filam ~= k
         for j=1:(n+1)
            aux=C(k,j); %para poder intercambiar las filas, utilizamos una
                     %variable auxiliar
            C(k,j)=C(filam,j);
            C(filam,j)=aux;
         end
         end
      end
```

```
      fprintf('\nLa matriz correspondiente a esta etapa antes del proceso:\n')

       disp(C)
       fprintf('\nLos Multiplicadores correpondientes a esta etapa son:\n')
       for i=(k+1):n
         m(i,k)=C(i,k)/C(k,k); %formula multiplicadores
         fprintf('\nm(%g,%g)=',i,k)
         disp(m(i,k));
         for j=k:(n+1)
            C(i,j)= C(i,j) - m(i,k)*C(k,j);%formula nueva fila
         end
       end
       fprintf('\nLa matriz correspondiente a esta etapa despues del proceso:\n')

       disp(C)
   end
       for i=n:-1:1
         suma=0;
           for p=(i+1):n
            suma = suma + C(i,p)*X(p);
           end
         X(i)=(C(i,n+1)-suma)/C(i,i);
         %formula de la susticion regresiva y solucion de las variables
       end
else %funcion asignada del if, en caso de que este sea falso
    fprintf('\nERROR: La matriz NO es cuadrada\n');
end
fprintf('\n\n SOLUCION:\n');
fprintf('\n\nLa matriz Ab final:\n');
disp(C)
fprintf('\n\nLa solucion de X1 hasta Xn es:\n');

%Se muestran los resultados de todo el proceso
for i=1:n
   Xi=X(1,i);
   fprintf('\nX%g=',i)
   disp(Xi);
end
```

ELIMINACION GAUSSIANA CON PIVOTEO PARCIAL TEST

ELIMINACION GAUSSIANA CON PIVOTEO PARCIAL (SOLUCION POR ETAPAS)

Ingrese la matriz A =
[4 3 -2 -7;3 12 8 -3; 2 3 -9 3;1 -2 -5 6]

Ingrese el vector b, correspondite a los terminos independientes b=
[20;18;31;12]

La Matriz C, que corresponte a la matriz aumentada [Ab] es =

4	3	-2	-7	20
3	12	8	-3	18
2	3	-9	3	31
1	-2	-5	6	12

ETAPA 1=

La matriz correspondiente a esta etapa antes del proceso:

4	3	-2	-7	20
3	12	8	-3	18
2	3	-9	3	31
1	-2	-5	6	12

Los Multiplicadores correpondientes a esta etapa son:

m(2,1)= 0.750000000000000

m(3,1)= 0.500000000000000

m(4,1)= 0.250000000000000

La matriz correspondiente a esta etapa despues del proceso:

4.000000000000000	3.000000000000000	-2.000000000000000	-7.000000000000000
20.000000000000000			
0	9.750000000000000	9.500000000000000	2.250000000000000
3.000000000000000			
0	1.500000000000000	-8.000000000000000	6.500000000000000
21.000000000000000			

0 -2.750000000000000 -4.500000000000000 7.750000000000000
7.000000000000000

ETAPA 2=

La matriz correspondiente a esta etapa antes del proceso:
4.000000000000000 3.000000000000000 -2.000000000000000 -7.000000000000000
20.000000000000000
0 9.750000000000000 9.500000000000000 2.250000000000000
3.000000000000000
0 1.500000000000000 -8.000000000000000 6.500000000000000
21.000000000000000
0 -2.750000000000000 -4.500000000000000 7.750000000000000
7.000000000000000

Los Multiplicadores correpondientes a esta etapa son:

m(3,2)= 0.153846153846154

m(4,2)= -0.282051282051282

La matriz correspondiente a esta etapa despues del proceso:
4.000000000000000 3.000000000000000 -2.000000000000000 -7.000000000000000
20.000000000000000
0 9.750000000000000 9.500000000000000 2.250000000000000
3.000000000000000
0 0 -9.461538461538462 6.153846153846154 20.538461538461540
0 0 -1.820512820512821 8.384615384615385 7.846153846153846

ETAPA 3=

La matriz correspondiente a esta etapa antes del proceso:
4.000000000000000 3.000000000000000 -2.000000000000000 -7.000000000000000
20.000000000000000
0 9.750000000000000 9.500000000000000 2.250000000000000
3.000000000000000
0 0 -9.461538461538462 6.153846153846154 20.538461538461540
0 0 -1.820512820512821 8.384615384615385 7.846153846153846

Los Multiplicadores correpondientes a esta etapa son:

m(4,3)= 0.192411924119241

La matriz correspondiente a esta etapa despues del proceso:

4.000000000000000 3.000000000000000 -2.000000000000000 -7.000000000000000 20.000000000000000
0 9.750000000000000 9.500000000000000 2.250000000000000 3.000000000000000
0 0 -9.461538461538462 6.153846153846154 20.538461538461540
0 0 0 7.200542005420054 3.894308943089430

SOLUCION:

La matriz Ab final:

4.000000000000000 3.000000000000000 -2.000000000000000 -7.000000000000000 20.000000000000000
0 9.750000000000000 9.500000000000000 2.250000000000000 3.000000000000000
0 0 -9.461538461538462 6.153846153846154 20.538461538461540
0 0 0 7.200542005420054 3.894308943089430

La solucion de X1 hasta Xn es:

X1= 3.570568310124200

X2= 1.955212645841174

X3= -1.818968761761385

X4= 0.540835528791870

% ELIMINACION GAUSSIANA CON PIVOTEO TOTAL

```
format long %permite utilizar la maxima capacidad de la maquina

fprintf('METODO ITERATIVO DE JACOBI\n\n\n')

a = input('Ingrese la matriz de coeficientes:\n ');
b = input('\nIngrese los terminos independientes:\n ');
x = input('\nIngrese el vector con las aproximacimaciones Iniciales:\n ');
iter = input('\nIngrese el numero maximo de iteraciones:\n ');
tol = input('\nIngrese la tolerancia:\n ');


cond = norm(a)*norm(a^-1);%Se calcula el condicional de la matriz de coeficientes
disp('condicional=')
disp(cond)% la funcion disp nos permite imprimir una variable en el espacio de trabajo

determinante=det(a); %se calcula el determinante de la matriz de coeficiente
if determinante==0
   disp('El determinante es cero, el problema no tiene solucion unica')
   return
end

n = length(b);%numero de elementos del vector b
d = diag(diag(a)); %obtencion de la matriz diagonal
l = d-tril(a); %obtencion de la matriz diagonal superior L
u = d-triu(a);%obtencion de la matriz diagonal inferior u
fprintf('\nSOLUCION:\n')
fprintf('\nLa matriz de transicion de jacobi:\n')
T = inv(d)*(l+u); % matriz de transicion de jacobi
disp(T)
re = max(abs(eig(T))) %calculo del radio espectral

if re>1
  disp('Radio Espectral mayor que 1')
  disp('el metodo no converge')
  return
end
fprintf('\nEl vector constante es::\n')
C = (d^-1)*b; % vector constante C, para el metodo
disp(C)
i = 0;
err = tol + 1;
z = [i,x(1),x(2),x(3),err]; %vector que me permite graficar la tabla

while err>tol && i<iter

  xi = T*x+C;
```

```
 %disp(xi)
 err = norm(xi-x); %norma 2
 %err=max(abs(xi-x)); %norma 1
 %err=norm(xi-x)/norm(xi); %norma relativa
 x = xi;

 i = i+1;
 z(i,1) =  i;
 z(i,2) = x(1);
 z(i,3) = x(2);
 z(i,4) = x(3);
 z(i,5) = err;

end
fprintf('\nTABLA:\n\n    n              x1              x2              x3              Error\n\n   ');
disp(z)%impresion de la tabla.

----------------------------------------------------------------
function [  ] = jacobi (vectorini,nmax,delta,tolerancia)



format long



A=matriz();

b=vectorb();

[D,L,U]=partirmatriz();



T=(D^-1)*(L+U);

C=(D^-1)*b;

i=0;

m=delta+1;

error=tolerancia+1;

while i<nmax & m>delta & error>tolerancia

        i=i+1
```

```
        X=T*vectorini+C
        m=norm(A*X-b)
        error=norm(X-vectorini)
        vectorini=X;
endwhile

endfunction
```

ELIMINACION GAUSSIANA CON PIVOTEO TOTAL TEST

ELIMINACION GAUSSIANA CON PIVOTEO TOTAL (SOLUCION POR ETAPAS)

Ingrese la matriz A =
[4 3 -2 -7;3 12 8 -3; 2 3 -9 3;1 -2 -5 6]

Ingrese el vector b, correspondite a los terminos independientes b=
[20;18;31;12]

La Matriz C, que corresponte a la matriz aumentada [Ab] es =

4	3	-2	-7	20
3	12	8	-3	18
2	3	-9	3	31
1	-2	-5	6	12

ETAPA 1=

La matriz correspondiente a esta etapa antes del proceso:

12	3	8	-3	18
3	4	-2	-7	20
3	2	-9	3	31
-2	1	-5	6	12

Los Multiplicadores correpondientes a esta etapa son:

m(2,1)= 0.250000000000000

m(3,1)= 0.250000000000000

m(4,1)= -0.166666666666667

La matriz correspondiente a esta etapa despues del proceso:

12.000000000000000 3.000000000000000 8.000000000000000 -3.000000000000000 18.000000000000000
0 3.250000000000000 -4.000000000000000 -6.250000000000000 15.500000000000000
0 1.250000000000000 -11.000000000000000 3.750000000000000 26.500000000000000
0 1.500000000000000 -3.666666666666667 5.500000000000000 15.000000000000000

ETAPA 2=

La matriz correspondiente a esta etapa antes del proceso:

12.000000000000000 8.000000000000000 3.000000000000000 -3.000000000000000 18.000000000000000

0 -11.000000000000000 1.250000000000000 3.750000000000000 26.500000000000000

0 -4.000000000000000 3.250000000000000 -6.250000000000000 15.500000000000000

0 -3.666666666666667 1.500000000000000 5.500000000000000 15.000000000000000

Los Multiplicadores correpondientes a esta etapa son:

m(3,2)= 0.363636363636364

m(4,2)= 0.333333333333333

La matriz correspondiente a esta etapa despues del proceso:

12.000000000000000 8.000000000000000 3.000000000000000 -3.000000000000000 18.000000000000000

0 -11.000000000000000 1.250000000000000 3.750000000000000 26.500000000000000

0 0 2.795454545454545 -7.613636363636363 5.863636363636363

0 0 1.083333333333333 4.250000000000000 6.166666666666666

ETAPA 3=

La matriz correspondiente a esta etapa antes del proceso:

12.000000000000000 8.000000000000000 -3.000000000000000 3.000000000000000 18.000000000000000

0 -11.000000000000000 3.750000000000000 1.250000000000000 26.500000000000000

0 0 -7.613636363636363 2.795454545454545 5.863636363636363

0 0 4.250000000000000 1.083333333333333 6.166666666666666

Los Multiplicadores correpondientes a esta etapa son:

m(4,3)= -0.558208955223881

La matriz correspondiente a esta etapa despues del proceso:

12.000000000000000	8.000000000000000	-3.000000000000000	3.000000000000000
18.000000000000000			
0	-11.000000000000000	3.750000000000000	1.250000000000000
26.500000000000000			
0	0	-7.613636363636363	2.795454545454545
5.863636363636363			
0	0	0	2.643781094527363
9.439800995024875			

El vector de marcas final es:

marca =

2 3 4 1

SOLUCION:

La matriz Ab final:

La solucion de X1 hasta Xn es:

X1= 3.570568310124200

X2= 1.955212645841174

X3= -1.818968761761385

X4= 0.540835528791871

% METODO DE GAUSS – SEIDEL

```
% METODO DE GAUSS - SEIDEL

format long %permite utilizar la maxima capacidad de la maquina

fprintf('METODO ITERATIVO DE GAUSS SEIDEL\n\n\n')

a = input('Ingrese la matriz de coeficientes:\n ');
b = input('\nIngrese los terminos independientes:\n ');
x = input('\nIngrese el vector con las aproximacimaciones Iniciales:\n ');
iter = input('\nIngrese el numero maximo de iteraciones:\n ');
tol = input('\nIngrese la tolerancia:\n ');

k = norm(a)*norm(a^-1);%Se calcula el condicional de la matriz de coeficientes
disp('condicional=')
disp(k)% la funcion disp nos permite imprimir una variable en el espacio de trabajo

determinante=det(a);%se calcula el determinante de la matriz de coeficiente
if determinante == 0
  disp('El determinante es cero, el problema no tiene solucion unica')
end

n = length(b); %numero de elementos del vector b
d = diag(diag(a)); %obtencion de la matriz diagonal
l = d-tril(a); %obtencion de la matriz diagonal superior L
u = d-triu(a); %obtencion de la matriz diagonal inferior u

fprintf('\n     SOLUCION:\n')
fprintf('\nLa matriz de transicion de gauss seidel:\n')
T = ((d-l)^-1)*u; % matriz de transicion de gauss
disp(T)
re = max(abs(eig(T))) %calculo del radio espectral

if re > 1
  disp('Radio Espectral mayor que 1')
  disp('el metodo no converge')
  return
end
fprintf('\nEl vector constante es::\n')
C = ((d-l)^-1)*b; % vector constante C, para el metodo
disp(C)
i = 0;

err = tol+1;
z = [i,x(1),x(2),x(3),err]; %vector que me permite graficar la tabla

while err > tol & i < iter
```

```
  xi = T*x+C;
  disp(xi)
  i = i+1;
  err = norm(xi-x); %norma 2
  err = max(abs(xi-x)); %norma 1
  err = norm(xi-x)/norm(xi); %norma relativa

  x = xi;
  z(i,1) = i;
  z(i,2) = x(1);
  z(i,3) = x(2);
  z(i,4) = x(3);
  z(i,5) = err;
end
fprintf('\nTABLA:\n\n    n              x1             x2             x3             Error\n\n   ')
disp(z) %impresion de la tabla.
```

METODO DE GAUSS – SEIDEL TEST

METODO ITERATIVO DE GAUSS SEIDEL

Ingrese la matriz de coeficientes:

[4 3 -2 -7;3 12 8 -3; 2 3 -9 3;1 -2 -5 6]

Ingrese los terminos independientes:

[20;18;31;12]

Ingrese el vector con las aproximacimaciones Iniciales:

[1;1;1;1]

Ingrese el numero maximo de iteraciones:

100

Ingrese la tolerancia:

0.001
condicional=
9.242075201603759

SOLUCION:

La matriz de transicion de gauss seidel:

0	-0.750000000000000	0.500000000000000	1.750000000000000
0	0.187500000000000	-0.791666666666667	-0.187500000000000
0	-0.104166666666667	-0.152777777777778	0.659722222222222
0	0.100694444444444	-0.474537037037037	0.195601851851852

re =

0.512922535167055

El vector constante es::
5.000000000000000
0.250000000000000
-2.250000000000000
-0.625000000000000
6.500000000000000
-0.541666666666667
-1.847222222222222
-0.803240740740741

3.076967592592593
1.761429398148148
-2.441277649176955
0.039917159636488

2.528144156164266
2.505461683652692
-2.034175017533770
0.718650687245378

3.361454931173007
2.195415617373939
-1.726099024866226
0.733146863207290

3.773395785149190
1.890670452758676
-1.731306275755969
0.558568956931386

3.693839697182933
1.870386498774426
-1.813939359835189
0.496206083531662

3.558601092181995
1.943694154394209
-1.840344122428711
0.521177767410478

3.534118415958325
1.973660919482179
-1.827471900822820
0.545973986481989

3.561472836320436
1.964439888088935
-1.816201411516262
0.547733313712687

3.577102677172371
1.953458600145920
-1.815802100453271
0.541800670142186

3.575156172412750
1.952195858067873

-1.818633118949369
0.539344991496025

TABLA:

n	x1	x2	x3	Error
1.000000000000000	6.500000000000000	-0.541666666666667	-1.847222222222222	0.971529321368615
2.000000000000000	3.076967592592593	1.761429398148148	-2.441277649176955	0.987877674601911
3.000000000000000	2.528144156164266	2.505461683652692	-2.034175017533770	0.292410863023257
4.000000000000000	3.361454931173007	2.195415617373939	-1.726099024866226	0.212375513221245
5.000000000000000	3.773395785149190	1.890670452758676	-1.731306275755969	0.117790937468656
6.000000000000000	3.693839697182933	1.870386498774426	-1.813939359835189	0.029055469707431
7.000000000000000	3.558601092181995	1.943694154394209	-1.840344122428711	0.035256126028094
8.000000000000000	3.534118415958325	1.973660919482179	-1.827471900822820	0.010666138581935
9.000000000000000	3.561472836320436	1.964439888088935	-1.816201411516262	0.006916041389732
10.000000000000000	3.577102677172371	1.953458600145920	-1.815802100453271	0.004450997085683
11.000000000000000	3.575156172412750	1.952195858067873	-1.818633118949369	0.000980930979266

% METODO DE JACOBI

```
% METODO DE JACOBI

format long %permite utilizar la maxima capacidad de la maquina

fprintf('METODO ITERATIVO DE JACOBI\n\n\n')

a = input('Ingrese la matriz de coeficientes:\n ');
b = input('\nIngrese los terminos independientes:\n ');
x = input('\nIngrese el vector con las aproximacimaciones Iniciales:\n ');
iter = input('\nIngrese el numero maximo de iteraciones:\n ');
tol = input('\nIngrese la tolerancia:\n ');


cond = norm(a)*norm(a^-1);%Se calcula el condicional de la matriz de coeficientes
disp('condicional=')
disp(cond)% la funcion disp nos permite imprimir una variable en el espacio de trabajo

determinante=det(a); %se calcula el determinante de la matriz de coeficiente
if determinante==0
   disp('El determinante es cero, el problema no tiene solucion unica')
   return
end

n = length(b);%numero de elementos del vector b
d = diag(diag(a)); %obtencion de la matriz diagonal
l = d-tril(a); %obtencion de la matriz diagonal superior L
u = d-triu(a);%obtencion de la matriz diagonal inferior u
fprintf('\nSOLUCION:\n')
fprintf('\nLa matriz de transicion de jacobi:\n')
T = inv(d)*(l+u); % matriz de transicion de jacobi
disp(T)
re = max(abs(eig(T))) %calculo del radio espectral

if re>1
  disp('Radio Espectral mayor que 1')
  disp('el metodo no converge')
  return
end
fprintf('\nEl vector constante es::\n')
C = (d^-1)*b; % vector constante C, para el metodo
disp(C)
i = 0;
err = tol + 1;
z = [i,x(1),x(2),x(3),err]; %vector que me permite graficar la tabla

while err>tol && i<iter
```

```
 xi = T*x+C;
 %disp(xi)
 err = norm(xi-x); %norma 2
 %err=max(abs(xi-x)); %norma 1
 %err=norm(xi-x)/norm(xi); %norma relativa
 x = xi;

 i = i+1;
 z(i,1) =  i;
 z(i,2) = x(1);
 z(i,3) = x(2);
 z(i,4) = x(3);
 z(i,5) = err;

end
fprintf('\nTABLA:\n\n    n              x1              x2              x3              Error\n\n   ');
disp(z)%impresion de la tabla.

---------------------------------------------------------------------
function [  ] = jacobi (vectorini,nmax,delta,tolerancia)



format long



A=matriz();

b=vectorb();

[D,L,U]=partirmatriz();



T=(D^-1)*(L+U);

C=(D^-1)*b;

i=0;

m=delta+1;

error=tolerancia+1;

while i<nmax & m>delta & error>tolerancia
```

```
        i=i+1
        X=T*vectorini+C
        m=norm(A*X-b)
        error=norm(X-vectorini)
        vectorini=X;
endwhile


endfunction
```

METODO DE JACOBI TEST

METODO ITERATIVO DE JACOBI

Ingrese la matriz de coeficientes:

[4 3 -2 -7;3 12 8 -3; 2 3 -9 3;1 -2 -5 6]

Ingrese los terminos independientes:

[20;18;31;12]

Ingrese el vector con las aproximacimaciones Iniciales:

[1;1;1;1]

Ingrese el numero maximo de iteraciones:

100

Ingrese la tolerancia:

0.001
condicional=
9.242075201603759

SOLUCION:

La matriz de transicion de jacobi:

0	-0.750000000000000	0.500000000000000	1.750000000000000
-0.250000000000000	0	-0.666666666666667	0.250000000000000
0.222222222222222	0.333333333333333	0	0.333333333333333
-0.166666666666667	0.333333333333333	0.833333333333333	0

re =

0.869684907945416

El vector constante es::
5.000000000000000
1.500000000000000
-3.444444444444444
2.000000000000000

TABLA:

n	x1	x2	x3	Error
1.000000000000000	6.500000000000000	0.833333333333333	-2.555555555555555	6.849799492424560
2.000000000000000	8.347222222222221	2.328703703703703	-0.722222222222222	4.949258129827670
3.000000000000000	1.255787037037039	-0.339120370370370	-1.125000000000000	7.779507936207732
4.000000000000000	6.062403549382717	2.131847993827160	-3.017361111111110	5.726441127666427
5.000000000000000	3.187717013888889	2.181013695987653	-1.139906978737997	3.769254241919536
6.000000000000000	1.369344912765778	1.259445685799611	-2.280475715877915	3.114556134532379
7.000000000000000	5.095320727326961	2.989429894243683	-2.305065041652377	4.217535933101257
8.000000000000000	2.114985689995512	1.835678507580006	-1.218609435877010	3.376111275998900
9.000000000000000	3.410082108471893	1.840252111956567	-2.287098463424411	1.963239188434058
10.000000000000000	4.653064560172709	2.483183621239111	-1.658601930477421	1.890451757928035
11.000000000000000	2.551832510008386	1.477257221871736	-1.536317235197517	2.392460219008438
12.000000000000000	4.296483933056976	2.053765572209516	-2.161601866600980	1.944426311904802
13.000000000000000	3.755861150218728	2.063659251042555	-1.542798615375972	1.029216728931144
14.000000000000000	2.973408968550936	1.631360367627514	-1.866197889695144	1.129012132527768
15.000000000000000	4.201807859636194	2.194840695783467	-1.981152091391369	1.385599200756626
16.000000000000000	3.226137441949990	1.893579525944713	-1.614744559362558	1.090709481561732
17.000000000000000	3.438059389394239	1.865050057211453	-1.969547777864326	0.553795242066945
18.000000000000000	3.925234055321186	2.140487788441268	-1.809453515890426	0.674370208832216
19.000000000000000	3.202828769471722	1.826839739917781	-1.722879734020034	0.805284904244006
20.000000000000000	3.733401894500170	1.985732377899976	-1.939954030027622	0.601347918391485
21.000000000000000	3.659688879971445	2.019804381404358	-1.739752561907544	0.305756012182914

22.000000000000000 3.355605807081780 1.850675017577238 -1.816895860029244
0.404108875030581
23.000000000000000 3.777216619848152 2.025743999636533 -1.877353866272772
0.465786947171030
24.000000000000000 3.493217337781437 1.943151127883413 -1.748633588603429
0.328521846669173
25.000000000000000 3.510507957136866 1.912763976626137 -1.860040101861818
0.172569770519151
26.000000000000000 3.699966084298560 2.014479619247339 -1.823970407595970
0.242374938102205
27.000000000000000 3.456477432085021 1.916606229272937 -1.783246581139745
0.267531807513277
28.000000000000000 3.606921529705169 1.958425909247586 -1.859183987767002
0.178616536952565
29.000000000000000 3.610901692736431 1.981913197783464 -1.797847572496204
0.099334215700094
30.000000000000000 3.493727667484507 1.921423389728153 -1.813939094022139
0.145066179234624
31.000000000000000 3.633038239128080 1.976014430824861 -1.840714563963532
0.152649183142717
32.000000000000000 3.554130457109077 1.955526134604547 -1.796240934259856
0.096805353144632
33.000000000000000 3.543898457705323 1.938770417008634 -1.829716864512912
0.058208159475514
34.000000000000000 3.615647921053950 1.974491509448975 -1.823114807018120
0.086471474911938
35.000000000000000 3.536549230080529 1.944208643321400 -1.805855622519053
0.086609712566171
36.000000000000000 3.577429349552498 1.953839818440628 -1.831710564277560
0.052322789834008
37.000000000000000 3.587853587687792 1.965224286462207 -1.813591874843290
0.034626827778587
38.000000000000000 3.544364275297812 1.944251914054452 -1.815863470251924
0.051289974121500
39.000000000000000 3.588981382547416 1.960927728469337 -1.826799766858993
0.048898330367033
40.000000000000000 3.568144129004348 1.956655424000453 -1.811871438746824
0.028215320384402
41.000000000000000 3.559577990987668 1.949164728584142 -1.821589761104644
0.020829486936814
42.000000000000000 3.585692684544449 1.961407406542114 -1.821160563293447
0.030267034016850
43.000000000000000 3.560681606138432 1.952300648586910 -1.814332342328451
0.027479430483893
44.000000000000000 3.571076475083418 1.954022798020715 -1.822897242880681
0.015196429424283
45.000000000000000 3.577442857367120 1.958839792290980 -1.817738851786050
0.012616033765828

46.000000000000000 3.561891577633833 1.951735307955362 -1.817483717235057
0.017771839040392
47.000000000000000 3.575823213665622 1.956663853260105 -1.821693298524883
0.015371792361428
48.000000000000000 3.570785592655449 1.956096414437750 -1.816809087426404
0.008190648935300
49.000000000000000 3.566325895525619 1.953052910129997 -1.819513396806752
0.007664821162710
50.000000000000000 3.575517477066469 1.957150865618431 -1.819945397668306
0.010384589511897
51.000000000000000 3.567809592989703 1.954509768374779 -1.817378442073496
0.008558782133816
52.000000000000000 3.570153488548920 1.954593949211107 -1.820146986220203
0.004431107123334
53.000000000000000 3.573157050554345 1.956489524234254 -1.818750256508072
0.004657551929294
54.000000000000000 3.567761743212189 1.954140053315640 -1.818340842978045
0.006039414971207
55.000000000000000 3.571995163841608 1.955539739322891 -1.819891221771311
0.004742753727681
56.000000000000000 3.570970377943096 1.955629279742508 -1.818331486675964
0.002415555606807
57.000000000000000 3.569003856923193 1.954462905212686 -1.819039700212497
0.002824972949539
58.000000000000000 3.572150272531516 1.955801783750954 -1.819365355159389
0.003496092144576
59.000000000000000 3.569843658235335 1.955069479158368 -1.818436930829362
0.002615502578836
60.000000000000000 3.570245493308798 1.954939810959635 -1.819310109607500
0.001332726928880
61.000000000000000 3.571505692040780 1.955649976658409 -1.818959361948467
0.001708052409246
62.000000000000000 3.569682214199076 1.954891634145990 -1.818721877295192
0.002014522535280
63.000000000000000 3.570927925773574 1.955268925127597 -1.819273549266336
0.001435446519935
64.000000000000000 3.570804934576076 1.955387537551536 -1.818787948806843
0.000747204224750