Report

Homework nr.5

Student: Macrii Danu
Gr.: FAF-222
Teacher: Bostan Viorel

Chișinău, Spring 2023

Problem 1

My code:

```
C=[2 1 0 0 0;
-1 2 1 0 0;
0 -1 2 1 0;
0 0 -1 2 1;
0 0 0 -1 2]
b=[3 2 2 2 1];
b=b'
n=5
iC=inv(C)
X=iC*b
```

Explaining:

Here is a simple problem with a small matrix so i just used inverse method to solve this tridiagonal system

Results

```
X =

   1.000000000000000
   1.000000000000000
   1.000000000000000
   1.000000000000000
   1.000000000000000
```

Problem 2

My code:

```
C=[3 -1 3;1 2 -1;1 2 0]
b=[11 2 0];
b=b'
n=3

A=[C b];

[L,U]=lu(A);

x = zeros(n,1);
x(n) = U(n,n+1)/U(n,n);

for i=n-1:-1:1
    summ = 0;
    for j=i+1:n
        summ = summ + U(i,j)*x(j,:);
        x(i,:) = (U(i,n+1) - summ)/U(i,i);
    end
end
x
% strait forward method
A=[C b];

A(2,:) = A(2,:) - (A(2,1)/A(1,1))*A(1,:);
A(3,:) = A(3,:) - (A(3,1)/A(1,1))*A(1,:);
A(3,:) = A(3,:) - (A(3,2)/A(2,2))*A(2,:);
A
x3=A(3,4)/A(3,3)
x2=(A(2,4)-x3*A(2,3))/A(2,2)
x1=(A(1,4)-x2*A(1,2)-x3*A(1,3))/A(1,1)
```

Explaining:

Here i used the build in function for LU Factorization and from U matrix derived the results , also all of this operation by hand to exercise. To find L by hand is also easy by operation with the matrix or just find $L = U^{-1} * A$

Result

```
x =

   4.857142857142857
  -2.428571428571428
  -2.000000000000000
```

Problem 3

My code:

```
B=[-2/3;5;4]
A=[0 3 1 ; -2 0 -3 ; 1 4 0]
P1=[1;1;1]

P2=zeros(3,1);

for i=1:100
P2= B-1/10*A*P1
P1=P2;
endfor
```

Explaining:

Here is a simple program that just with a for to do the iteration .Also we see that it converges at the 41st iteration.

Result

```
1 iteration
P2 =

  -1.066666666666667
   5.500000000000000
   3.500000000000000
20 iteration
P2 =

  -2.434229545879187
   5.166233015320869
   2.176929748783412


41 iteration
P2 =

  -2.434229546111593
   5.166233015322346
   2.176929748482221
```

Problem 4

My code:

```
clc clear
close all

A=[0 1 5 -7 23 -1 7 8 1 -5;17 0 -24 -75 100 -18 10 -8 9 -50;
3 -2 15 0 -78 -90 -70 18 -75 1;5 5 -10 0 -72 -1 80 -3 10 -18;
100 -4 -75 -8 0 83 -10 -75 3 -8;70 85 -4 -9 2 0 3 -17 -1 -21;
1 15 100 -4 -23 13 0 7 -3 17;16 2 -7 89 -17 11 -73 0 -8 -23;
51 47 -3 5 -10 18 -99 -18 0 12;1 1 1 1 1 1 1 1 1 0];
b=[10 -40 -17 43 -53 12 -60 100 0 100];
b=b';
n=10;
C=[A b]

t=C(1,:);
C(1,:) = C(5,:);    %swap first and fifth  row
C(5,:) = t;

t=C(2,:);
C(2,:) = C(6,:);    %swap second and sixth  row
C(6,:) = t;

t=C(3,:);
C(3,:) = C(7,:);    %swap third and seven  row
C(7,:) = t;

t=C(4,:);
C(4,:) = C(6,:);    %swap fourth and sixth row
C(6,:) = t;

t=C(6,:);
C(6,:) = C(9,:);    %swap six and ninght  row
C(9,:) = t;

t=C(7,:);
C(7,:) = C(8,:);    %swap seven and eight  row
C(8,:) = t;

t=C(8,:);
C(8,:) = C(9,:);    %swap eight and ninght  row
C(9,:) = t;


C(10,:) = C(10,:)-1 ;
```

```
C(9,:) = C(9,:)+C(1,:)+C(3,:)


%Gauss elimination
[L,U]=lu(C);

x = zeros(n,1);
x(n) = U(n,n+1)/U(n,n);

for i=n-1:-1:1
    summ = 0;
    for j=i+1:n
        summ = summ + U(i,j)*x(j,:);
        x(i,:) = (U(i,n+1) - summ)/U(i,i);
    end
end
x

%Gauss-Jordan method

Aug=C
for j =1:N
  Aug(j,:) = Aug(j,:)/Aug(j,j);
  for i =1:N
    if i~=j
      m = Aug(i,j);
      Aug(i,:) = Aug(i,:)-m*Aug(j,:);
    endif
  endfor
endfor
Aug
```

Explaining:

In this problem for iteration method we need to (of course) transform this matrix into an diagonally dominant one , it was quite confusing to do but eventually figured out how to make the 9th row diagonally dominant.I chose Gauss-Seidel method because for diagonally dominant matrices it converges more rapidly than Gauss-Jordan method.Also for direct method i chose of course Gauss elimination because inverse matrix method requires $2n^3$ and it is more computationally needy.

Result

```
%Result for Gauss Eliminaton
x =

  -1.593589558311444e+02
   1.028419963367540e+02
  -3.497232468802653e+01
  -5.732537476382350e+01
  -2.270804990883450e+01
   1.751231336404339e+02
  -1.718936175298308e+01
   2.320004993586198e+01
  -1.821245819157730e+02
  -9.900000000000000e+01


%Result for Gauss-Jordan method
x =

  -1.593589558311442e+02
   1.028419963367539e+02
  -3.497232468802638e+01
  -5.732537476382328e+01
  -2.270804990883446e+01
   1.751231336404337e+02
  -1.718936175298307e+01
   2.320004993586192e+01
  -1.821245819157727e+02
  -9.900000000000000e+01
```

Problem 5

My code:

```
A=[8 3 2;16 6 4.001;4 1.501 1]
B=[20 ;40.02 ;10.01]
N=length(B)
X=zeros(N,1);
C=[A B];
C(2,:) = C(2,:) -2*C(1,:); %second row minus double first row
C(3,:) = C(3,:) -0.5*C(1,:); %third row minus half first row
t = C(2,:);  % swap third and secon row
C(2,:) = C(3,:);
C(3,:) = t;
% Matrix is now diagonally dominant

%Gauss-Seidel method

e=10^-5;
N=size(A,1)
X=zeros(N,1);
Y=zeros(N,1);


C(2,:) = C(3,:);
C(3,:) = t;

A(:,1) = C(:,1);
A(:,2) = C(:,2);
A(:,3) = C(:,3);
B = C(:,4);

A
B

for j = 1:n
    for i = 1:N
        X(i) = (B(i)/A(i,i)) - (A(i,[1:i-1,i+1:N])*P([1:i-1,i+1:N]))/A(i,i);
        P(i) = X(i);
    end
    if abs(Y-X)<e
      break
    endif
    Y=X;
end

X
```

```
%Gauss-Jordan method
Aug=C
for j =1:N
  Aug(j,:) = Aug(j,:)/Aug(j,j);
  for i =1:N
    if i~=j
      m = Aug(i,j);
      Aug(i,:) = Aug(i,:)-m*Aug(j,:);
    endif
  endfor
endfor
Aug
```

Explaining:

For the methods to converge we need the matrix to be diagonally dominant . I adjusted the matrix and after used the methods , they both converged very quickly , around 4-th iteration. Also i put the condition for the program to stop when error is minimal.

Result

```
%Result for Gauss-Jordan method
X=
    -6.249999999999445e+00
    1.000000000000089e+01
    1.999999999999645e+01


%Result for Gauss-Seidel method
X=
    -6.249999999999446e+00
    1.000000000000089e+01
    1.999999999999645e+01
```

Appendix

1