W4 NUMERICAL ANALYSIS 2019-2020 FALL

LINEAR SYSTEM OF EQUATIONS AX=B

CLASS EXERCISES

Class exercises will be completed and graded in class

EX 1 Following excel files(programs) are given investigate solutions and solve the following system of equation:

$$\begin{bmatrix} \mathbf{1} & \mathbf{2} & \mathbf{3} \\ \mathbf{2} & \mathbf{1} & \mathbf{2} \\ \mathbf{3} & \mathbf{2} & \mathbf{1} \end{bmatrix} \begin{Bmatrix} x_0 \\ x_1 \\ x_2 \end{Bmatrix} = \begin{Bmatrix} \mathbf{1} \\ \mathbf{1} \\ \mathbf{2} \end{Bmatrix}$$

gauss.xls

	A			B
	1	2	3	1
	2	1	2	1
	3	2	1	2
c	1	2	3	1
2	0	-3	-4	-1
3	0	-4	-8	-1
c	1	2	3	1
1.333333	0	-3	-4	-1
	0	0	-2.66667	0.333333

x0	0.375
x1	0.5
x2	-0.125

Gauss_jordan.xls

	A			B	
	1	2	3	1	
	2	1	2	1	
	3	2	1	2	
c	1	2	3	1	Normalisation
	2	1	2	1	1
	3	2	1	2	
	1	2	3	1	
2	0	-3	-4	-1	
3	0	-4	-8	-1	
	1	2	3	1	Normalisation
	0	1	1.333333	0.333333	-3

	0	-4	-8	-1
2	1	0	0.333333	0.333333
	0	1	1.333333	0.333333
-4	0	0	-2.66667	0.333333
	1	0	0.333333	0.333333
	0	1	1.333333	0.333333
	0	0	1	-0.125
0.333333	1	0	0	0.375
1.333333	0	1	0	0.5
	0	0	1	-0.125

Normalisation
-2.66667

Jacobi.xls

$$\begin{bmatrix} \mathbf{3} & \mathbf{-0.1} & \mathbf{-0.2} \\ \mathbf{0.1} & \mathbf{7} & \mathbf{-0.3} \\ \mathbf{0.3} & \mathbf{-0.2} & \mathbf{10} \end{bmatrix} \begin{Bmatrix} x_0 \\ x_1 \\ x_2 \end{Bmatrix} = \begin{Bmatrix} \mathbf{7.85} \\ \mathbf{-19.3} \\ \mathbf{71.4} \end{Bmatrix}$$

iterative method

3	-0.1	-0.2	x1	7.85
0.1	7	-0.3	x2	-19.3
0.3	-0.2	10	x3	71.4

Jacobi

Initial guess

x1	0
x2	0
x3	0
iter	1
x1	2.616667
x2	-2.75714
x3	7.14
iter	2
x1	3.000762
x2	-2.48852
x3	7.006357
iter	3
x1	3.000806
x2	-2.49974
x3	7.000207
iter	4
x1	3.000022
x2	-2.5
x3	6.999981

Gauss-Seidel

Initial guess

x1	0
x2	0
x3	0
iter	1
x1	2.616667
x2	-2.79452
x3	7.00561
iter	2
x1	2.990557
x2	-2.49962
x3	7.000291
iter	3
x1	3.000032
x2	-2.49999
x3	6.999999
iter	4
x1	3
x2	-2.5
x3	7

EX2 Following Octave(Matlab) files are given investigate codes and solve the following system of equation:

$$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 1 & 2 \\ 3 & 2 & 1 \end{bmatrix} \begin{Bmatrix} x_0 \\ x_1 \\ x_2 \end{Bmatrix} = \begin{Bmatrix} 1 \\ 1 \\ 2 \end{Bmatrix}$$

gauss1.m

```
function x=gauss1(a,b)
% Gauss elimination
n=length(b);
carpan=0;
toplam=0;
 for k=1:n
  %pivoting
  p=k;
  buyuk=abs(a(k,k));
  for ii=k+1:n
   dummy=abs(a(ii,k));
    if dummy > buyuk buyuk=dummy;p=ii;end
  end
  if p~=k
   for jj=k:n
    dummy=a(p,jj);
    a(p,jj)=a(k,jj);
    a(k,jj)=dummy;
    end
   dummy=b(p);
   b(p)=b(k);
   b(k)=dummy;
  end
  %gauss eleme
  for i=k+1:n
   carpan=a(i,k)/a(k,k);
    a(i,k)=0;
    for j=k+1:n
      a(i,j)=a(i,j)-carpan*a(k,j);
    end
       b(i)   =b(i)   -carpan*b(k);
  end
 end
 %back substituting
 x(n)=b(n)/a(n,n);
 for i=n-1:-1:1
  toplam=0;
  for j=i+1:n
   toplam=toplam+a(i,j)*x(j);end
  x(i)=(b(i)-toplam)/a(i,i);
 end
end
```

```
>> a=[1 2 3;2 1 2;3 2 1]
a =
   1   2   3
   2   1   2
   3   2   1
>> b=[1 1 2]'
b =
   1
   1
   2
>> gauss1(a,b)
ans =
```

```
   0.37500   0.50000  -0.12500
>>
```

Gaussjordan.m

```
function x=gaussjordan(a,b)
 n=length(b);
 np1=n+1;
 carpan=0;
 C=a;
 for i=1:n
    C(i,n+1)=b(i);
 end
 for k=1:n
  %gauss jordan elimination
  for j=k+1:np1
   C(k,j)=C(k,j)/C(k,k);
  end
  for i=1:n
   if i~=k
      carpan=C(i,k);
       C(i,k)=0;
       for j=k+1:np1
         C(i,j)=C(i,j)-carpan*C(k,j); end
     end
  end
 end
 for i=1:n
  C(i,i)=1;
  x(i)=C(i,np1);
 end
 end
```

EX3: Following java programs are given investigate solutions and solve the following system of equation(s):

$$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 1 & 2 \\ 3 & 2 & 1 \end{bmatrix} \begin{Bmatrix} x_0 \\ x_1 \\ x_2 \end{Bmatrix} = \begin{Bmatrix} 1 \\ 1 \\ 2 \end{Bmatrix}$$

NA15.java (gauss elimination with pivoting)

```
 import java.io.*;
import java.util.Locale;
class NA15
{ public static String toString(double a[])
  {int n=a.length;
   String s="";
   for(int i=0;i<n;i++)
   {s+=String.format(Locale.US,"%10g",a[i])+" ";}
   return s;
  }
  public static String toString(double a[][])
  {int n=a.length;
   int m=a[0].length;
   String s="";
   for(int i=0;i<n;i++)
   { for(int j=0;j<m;j++)
              {s+=String.format(Locale.US,"%10g",a[i][j])+" ";}
              s+="\n";
   }
   return s;
  }

  public static double[] pivotlugauss(double a[][],double b[])
  { //gauss elimination with partial pivoting
```

```
int n=b.length;
double x[]=new double[n];
double carpan=0;
double toplam=0;
double buyuk;
double dummy=0;
//gauss eleme
int i,j,k,p,ii,jj;
System.out.println("Original matrix :\n"+toString(a));
for(k=0;k<(n-1);k++)
{ //pivoting
        p=k;
        buyuk=Math.abs(a[k][k]);
        for(ii=k+1;ii<n;ii++)
        {  dummy=Math.abs(a[ii][k]);
           if(dummy > buyuk) {buyuk=dummy;p=ii;}
        }
        if(p!=k)
        { for(jj=k;jj<n;jj++)
          { dummy=a[p][jj];
            a[p][jj]=a[k][jj];
            a[k][jj]=dummy;
          }
          dummy=b[p];
          b[p]=b[k];
          b[k]=dummy;
        }
        System.out.println("pivoted matrix : k="+k+"\n"+toString(a));
        // Solving gauss elimination
  for(i=k+1;i<n;i++)
  {  carpan=a[i][k]/a[k][k];
     a[i][k]=0;
     for(j=k+1;j<n;j++)
     {   a[i][j]-=carpan*a[k][j]; }
        b[i]   =b[i]   -carpan*b[k];
         }
}
System.out.println("final upper triangular matrix : \n"+toString(a));
// back substitution
x[n-1]=b[n-1]/a[n-1][n-1];
for(i=n-2;i>=0;i--)
{
  toplam=0;
  for(j=i+1;j<n;j++)
  {  toplam+=a[i][j]*x[j];}
  x[i]=(b[i]-toplam)/a[i][i];
}
return x;
}

public static void main(String args[]) throws IOException
{
double a[][]={{1,2,3},{2,1,2},{3,2,1}};
double b[]={1,1,2};
double x[]=pivotlugauss(a,b);
```

```
  System.out.println(toString(x));
  }
}
```

```
 ---------- Capture Output ----------
> "E:\co\java\bin\java.exe" NA15
Original matrix :
   1.00000    2.00000    3.00000
   2.00000    1.00000    2.00000
   3.00000    2.00000    1.00000

pivoted matrix : k=0
   3.00000    2.00000    1.00000
   2.00000    1.00000    2.00000
   1.00000    2.00000    3.00000

pivoted matrix : k=1
   3.00000    2.00000    1.00000
   0.00000    1.33333    2.66667
   0.00000  -0.333333    1.33333

final upper triangular matrix :
   3.00000    2.00000    1.00000
   0.00000    1.33333    2.66667
   0.00000    0.00000    2.00000

  0.375000   0.500000  -0.125000

> Terminated with exit code 0.
```

NA17.java Gauss-Jordan elimination with pivoting

```
 import java.io.*;

class NA17
{ public static double[][] C(double a[][],double b[])
  {int n=a.length;int m=a[0].length;double c[][]=new double[n][m+1];
  for(int i=0;i<n;i++){for(int j=0;j<m;j++){c[i][j]=a[i][j];};c[i][m]=b[i];}
  return c;
  }

  public static double[][] C(double a[][],double b[][])
  {int n=a.length;int o=b[0].length;int m=n+o;double c[][]=new double[n][m];
  for(int i=0;i<n;i++)
  {for(int j=0;j<n;j++){c[i][j]=a[i][j];};
   for(int j=n;j<m;j++){c[i][j]=b[i][j];}
  }
  return c;
  }

  public static double[][] gaussjordanwithpivot(double a[][])
  { //pivotlu
  int n=a.length;
  int m=a[0].length;
  int o=m-n;
  double x[][]=new double[n][o];
  double carpan=0;
  double toplam=0;
  double buyuk;
  double dummy=0;
  int i,j,k,p,ii,jj;
  i=0;
```

```
  for(k=0;k<n;k++)
  { //pivoting
                p=k;
                buyuk=Math.abs(a[k][k]);
                for(ii=0;ii<n;ii++)
                {  if(i!=k)
                   {
                   dummy=Math.abs(a[ii][k]);
                   if(dummy > buyuk) {buyuk=dummy;p=ii;}
      }
                }
                if(p!=k)
                { for(jj=k;jj<m;jj++)
                  { dummy=a[p][jj];
                    a[p][jj]=a[k][jj];
                    a[k][jj]=dummy;
                  }
                }
                //gauss jordan elimination
                for(j=k+1;j<m;j++)
                { a[k][j]/=a[k][k]; }
                a[k][k]=1.0;
    for(i=0;i<n;i++)
    {  if(i!=k)
       {
                     carpan=a[i][k];
         a[i][k]=0;
         for(j=k+1;j<m;j++)
           {a[i][j]-=carpan*a[k][j]; }
       }
                }
  }
  for(jj=0;jj<o;jj++)
  {for(i=0;i<n;i++) {x[i][jj]=a[i][n+jj];}}
  return x;
  }
  public static double[] gaussjordan(double a[][],double b[])
  {double c[][]=C(a,b);
   double d[][]=gaussjordanwithpivot(c);
   return d[0];
  }

  public static void main(String args[]) throws IOException
  { double a[][]={{1,2,3,1},{2,1,2,1},{3,2,1,2}};
    double x[][]=gaussjordanwithpivot(a);
    System.out.println(NA15.toString(x));
  }
}
```

```
---------- Capture Output ----------
> "E:\co\java\bin\java.exe" NA17
 0.375000
 0.500000
-0.125000
> Terminated with exit code 0.
```

LU decomposition(Gauss-dolittle)

```
 ---------- Capture Output ----------
> "E:\co\java\bin\java.exe" NA18
d=
            0.375000000000000
           0.500000000000000
          -0.125000000000000


Solution of Gauss(Doolittle) decomposition and elimination
Matrix a=
```

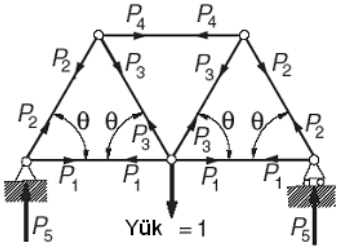
```
   1.00000   2.00000   3.00000
   2.00000   1.00000   2.00000
   3.00000   2.00000   1.00000
Matrix b =
   1.00000   1.00000   2.00000
Solution Matrix=
  0.375000  0.500000 -0.125000

> Terminated with exit code 0.
```

HOMEWORK EXERCISES

Homework exercises will be done at home and will bring to next Thursday class printed no late exercises will be excepted.

W3HW1: Solving method is not specified

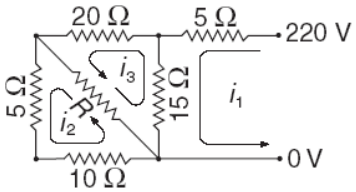


Load Matrix of the figure can be given with the following system of eqautions

$$\begin{bmatrix} c & 1 & 0 & 0 & 0 \\ 0 & s & 0 & 0 & 1 \\ 0 & 0 & 2s & 0 & 0 \\ 0 & -c & c & 1 & 0 \\ 0 & s & s & 0 & 0 \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

Where c=cos(θ),s=sin(θ) . For θ =60^o find the load distribution profile

W3HW2: **Solving method is not specified**



For the given electrical circuit current density-Voltage eqautions can be given as:

$$5i_1 + 15(i_1 - i_3) = 220\,\mathrm{V}$$

$$R(i_2 - i_3) + 5i_2 + 10i_2 = 0$$

$$20i_3 + R(i_3 - i_2) + 15(i_3 - i_1) = 0$$

Calculate current densities for R=10 Ω.

W3HW3 :

Solve system of equation by using

a) By using Gauss elimination
b) By using Gauss-Jordan elimination
c) LU decomposition (Dolittle)

$$
\begin{aligned}
x_1 + 3x_2 + x_3 + 5x_4 &= 4 \\
2x_1 + x_2 + 3x_4 &= 5 \\
4x_1 + 2x_2 + 2x_3 + x_4 &= 11 \\
-3x_1 + x_2 + 3x_3 + 2x_4 &= 3
\end{aligned}
$$

W3HW4:

Find the inverse matrix

$$
\begin{bmatrix}
1.1348 & 3.8326 & 1.1651 & 3.4017 \\
0.5301 & 1.7875 & 2.5330 & 1.5435 \\
3.4129 & 4.9317 & 8.7643 & 1.3142 \\
1.2371 & 4.9998 & 10.6721 & 0.0147
\end{bmatrix}
$$