Numerical Analysis

Project 2

Final Report

Abdalla Gamal Mubarak 6771

The Test cases

1]

LU -jordan -elimination

8*x+4*y-1*z=11

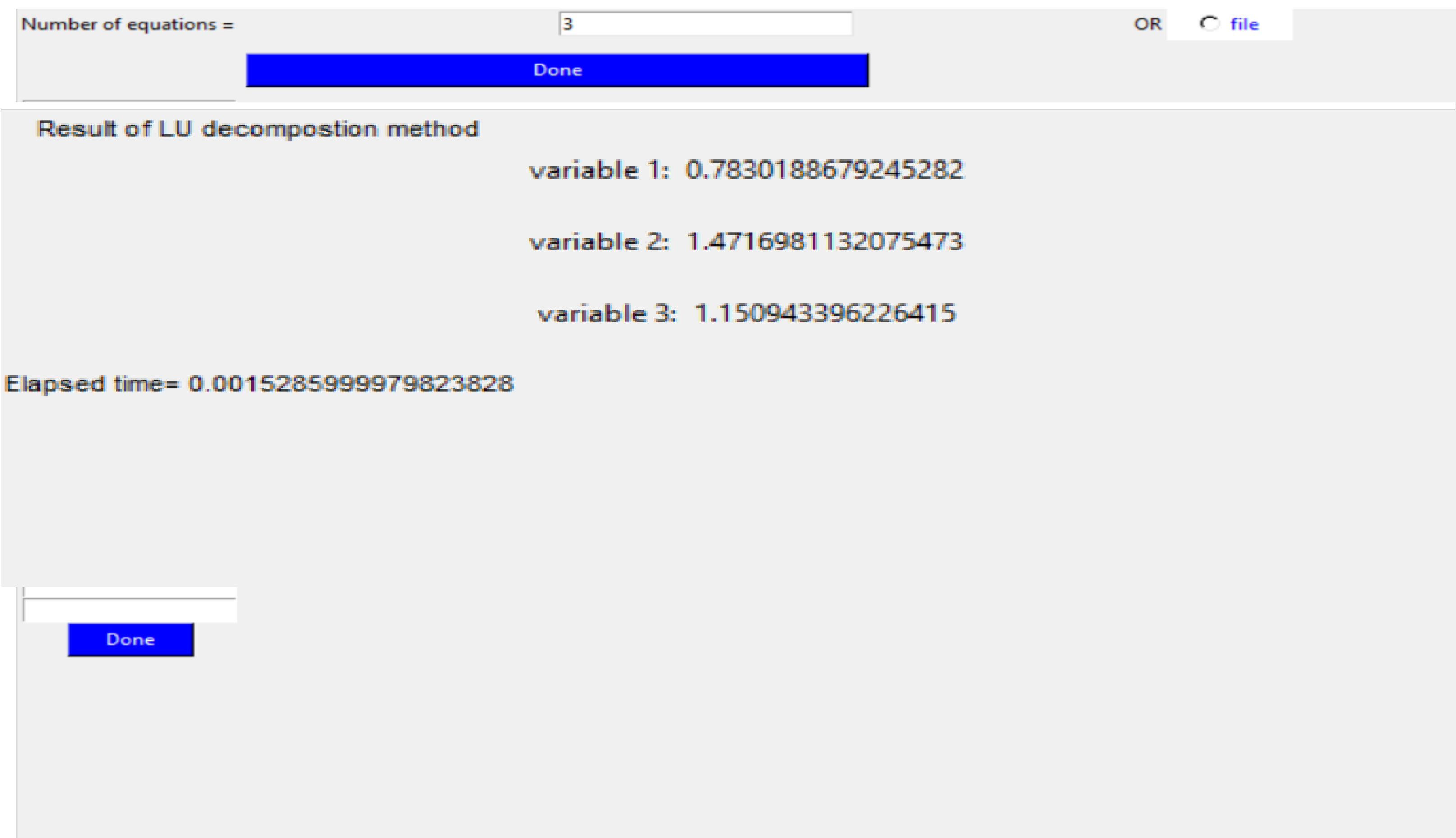
-2*x+3*y+1*z=4

2*x-1*y+6*z=7

Sol

x=0.783,y=1.4717andz=1.1509

solution for (LU -jordan -elimination) only



output

Output of all Method

Output

Result of Gauss Elmination method

variable 1: 0.7830188679245282

variable 2: 1.4716981132075473

variable 3: 1.150943396226415

Elapsed time= 0.0004631001502275467

```
##########################################################################################################
```

Result of Gauss jordan method

variable 1: 0.7830188679245282

variable 2: 1.4716981132075473

variable 3: 1.150943396226415

Elapsed time= 0.0006227998528629541

```
##########################################################################################################
```

Result of LU decompostion method

variable 1: 0.7830188679245282

variable 2: 1.4716981132075473

variable 3: 1.150943396226415

Elapsed time= 0.0012655998580157757

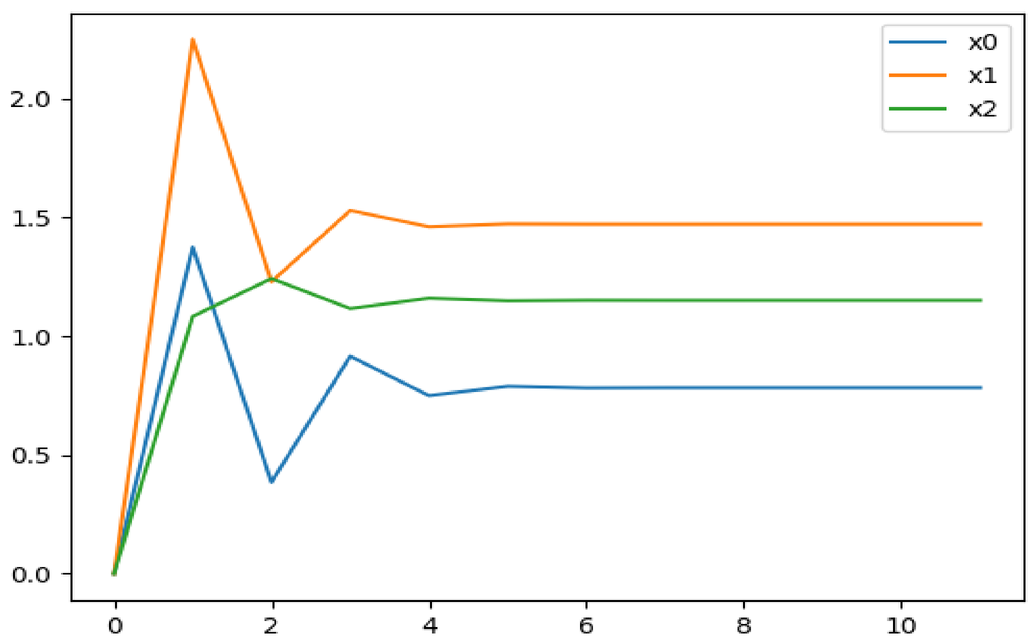
```
##########################################################################################################
```

Result of Gauss seidel method

Iteration	variable 1	variable 2	variable 3	error of variable 1	error of variable 2	error of variable 3
0	0	0	0			
1	1.375	2.25	1.0833333333333333	1.0	1.0	1.0
2	0.38541666666666663	1.2291666666666667	1.2430555555555556	2.567567567567568	0.8305084745762711	0.12849162011173193
3	0.915798611111111	1.529513888888889	1.1163194444444444	0.5791469194312796	0.19636776390465385	0.11353032659409025
4	0.749782986111111	1.4610821759259258	1.1602527006172838	0.22141823444283648	0.04683632042776532	0.037865247932166654
5	0.7894904996141976	1.4729094328703705	1.1489880722736625	0.05029511250925821	0.00802986027552011	0.009803955859463719
6	0.7821687925990226	1.471783170974794	1.1512409309627916	0.009360776196204549	0.0007652362914508469	0.0019568959272886284
7	0.783013530882952	1.4715953769343708	1.1509280525280776	0.0010788297399878092	0.00012761255122609923	0.00027184882150251965
8	0.7830683180988243	1.4717361945565237	1.1509332597264792	6.996479694808808e-5	9.568129306987142e-5	4.524326982102779e-6
9	0.782998560187548	1.471687953549539	1.1509484721957406	8.909072739491564e-5	3.277937205933536e-5	1.3217333033487029e-5
10	0.7830245822496981	1.471700230767885	1.1509418443780814	3.32327525086296e-5	8.342200462625162e-6	5.758603435541323e-6
11	0.7830176151633177	1.4716977953161845	1.1509437608315916	8.897739010531113e-6	1.6548585642271237e-6	1.6651148173602812e-6

0
0
0

Done


x0
x1
x2
2.0
1.5
1.0
0.5
0.0
0
2
4
6
8
10

2]

Seidel

10x+2*y-1*z=27

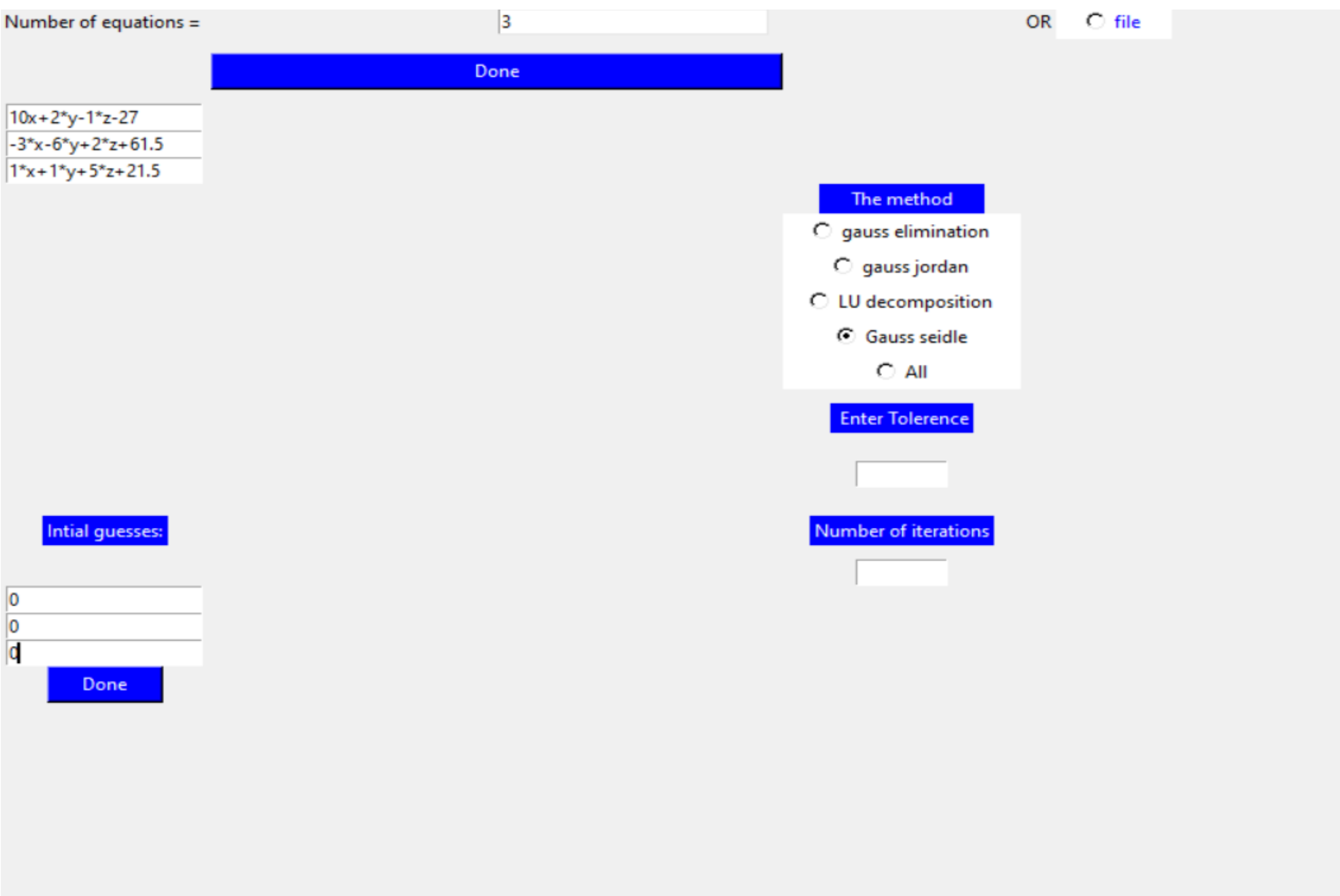
-3*x-6*y+2*z=-61.5

1*x+1*y+5*z=-21.5

Sol

x=0.5,y=8and z=-6

Seidel only



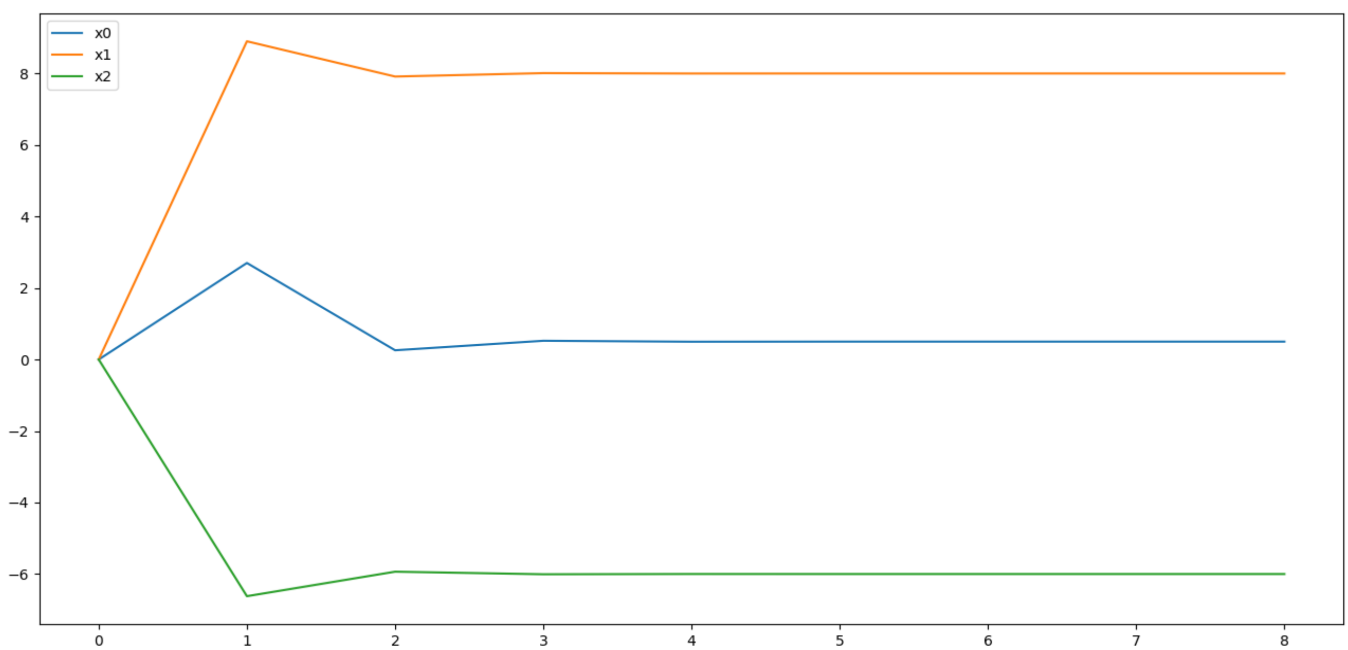
Output

Result of Gauss seidel method

Iteration	variable 1	variable 2	variable 3	error of variable 1	error of variable 2	error of variable 3
0	0	0	0			
1	2.7	8.9	-6.62	1.0	1.0	-1.0
2	0.2579999999999999	7.9143333333333326	-5.934466666666666	9.465116279069772	0.12454197026492034	-0.1155172607479471
3	0.5236866666666669	8.01000111111111	-6.0067375555555556	0.5073389940549696	0.011943541136975861	-0.01203163751045637
4	0.49732602222222233	7.999091137037037	-5.999283431851852	0.053004755968039184	0.00136390170922779	-0.001242502340217392
5	0.5002534294074075	8.00011214134568	-6.000073114150618	0.005851848309471661	0.00012762374959298956	-0.0001316121126763753
6	0.4999702603158022	7.999990498458558	-5.999992151754872	0.0005663718706516453	1.5205378949486495e-5	-1.3493750274683765e-5
7	0.5000026851328011	8.00000127351531	-6.000000791729622	6.484928573988812e-5	1.346881879446247e-6	-1.439995601630285e-6
8	0.49999966612397595	7.9999999030281375	-5.999999913830423	6.038021682212989e-6	1.7131089854503274e-7	-1.4631653536106985e-7

Number of iterations= 8

Elapsed time= 0.16877759993076324



All outputs

Result of Gauss Elmination method

variable 1: 0.5

variable 2: 8.0

variable 3: -6.0

Elapsed time= 0.0005415999330580235

##

Result of Gauss jordan method

variable 1: 0.5000000000000002

variable 2: 8.0

variable 3: -6.0

Elapsed time= 0.0005681999027729034

##

Result of LU decompostion method

variable 1: 0.5

variable 2: 8.0

variable 3: -6.0

Elapsed time= 0.0013411999680101871

##

Result of Gauss seidel method

Iteration	variable 1	variable 2	variable 3	error of variable 1	error of variable 2	error of variable 3
0	0	0	0			
1	2.7	8.9	-6.62	1.0	1.0	-1.0
2	0.2579999999999999	7.9143333333333326	-5.934466666666666	9.465116279069772	0.12454197026492034	-0.1155172607479471
3	0.5236866666666669	8.01000111111111	-6.0067375555555556	0.5073389940549696	0.011943541136975861	-0.01203163751045637
4	0.4973260222222233	7.999091137037037	-5.999283431851852	0.053004755968039184	0.0013639017092779	-0.001242502340217392
5	0.5002534294074075	8.00011214134568	-6.000073114150618	0.005851848309471661	0.00012762374959298956	-0.0001316121126763753
6	0.4999702603158022	7.999990498458558	-5.999992151754872	0.0005663718706516453	1.5205378949486495e-5	-1.3493750274683765e-5
7	0.5000026851328011	8.00000127351531	-6.000000791729622	6.484928573988812e-5	1.346881879446247e-6	-1.439995601630285e-6
8	0.4999996612397595	7.9999999030281375	-5.999999913830423	6.038021682212989e-6	1.7131089854503274e-7	-1.4631653536106985e-7

Number of iterations= 8

Elapsed time= 0.15326260006986558

3]

jordan -elimination

x+y+1m=2

2x+1y-1z+1m=1

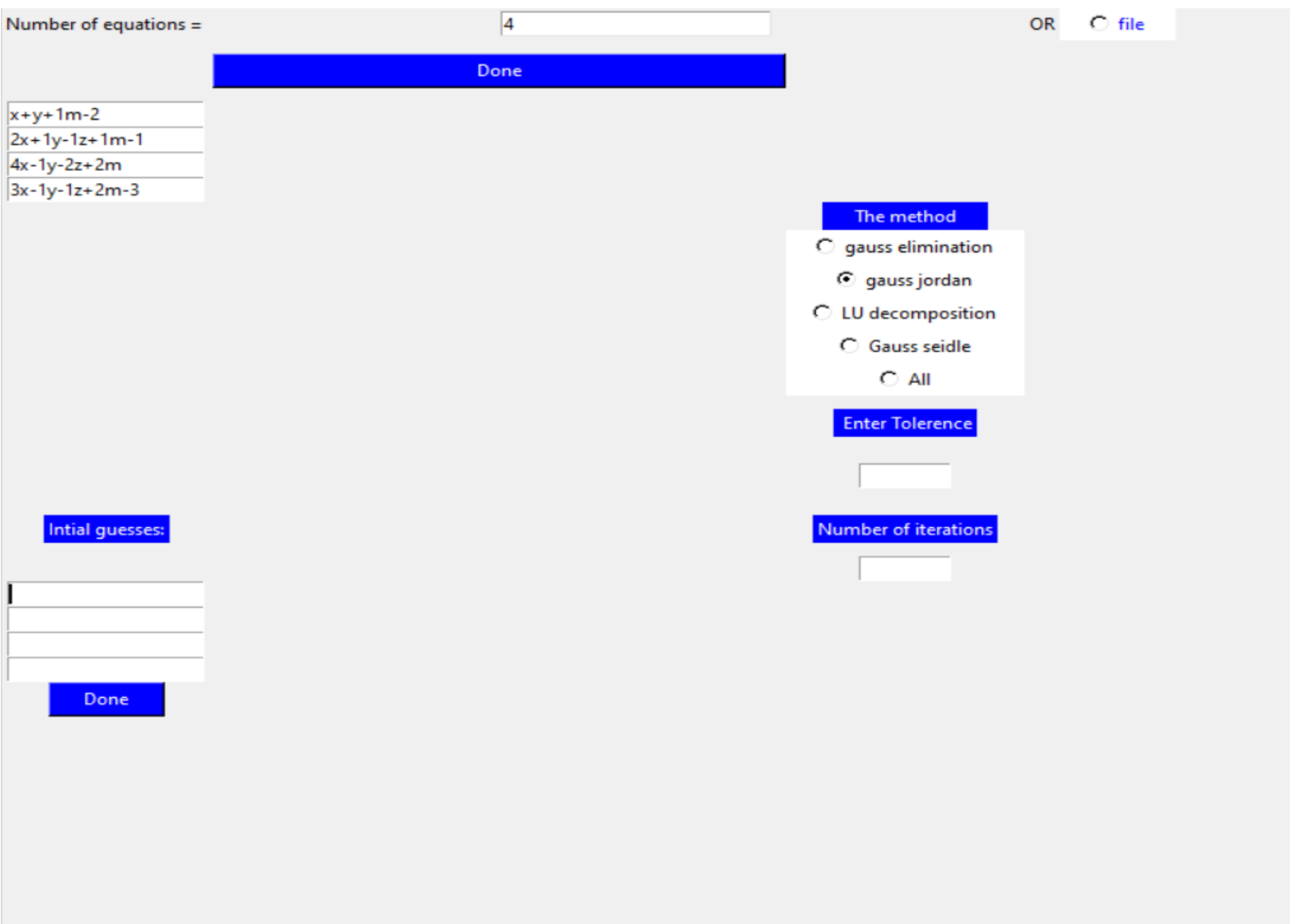
4x-1y-2z+2m=0

3x-1y-1z+2m=3

sol

no solution

Jordan-elimination only



Output

```
Result of Gauss jordan method

                    No solutions

Elapsed time= 0.0011785998940467834
```

Output of all Methods

```
Result of Gauss Elmination method
                                        No solutions
Elapsed time= 0.0009729000739753246
##########################################################################################
Result of Gauss jordan method
                                        No solutions
Elapsed time= 0.00024569989182054996
##########################################################################################
Result of LU decompostion method
                                        No solutions
Elapsed time= 0.0007048000115901232
##########################################################################################
Result of Gauss seidel method
                                        No solutions
Elapsed time= 0.00016930000856518745
```

Conclusion

System of equations has unique solutions. In non iterative methods: Gauss elimination, Gauss Jordan ,and Lu decomposition the result was the exact solution of the variables ,however Gauss seidel the error tolerance was satisfied after three iterations.

Pseudo code

1]

Gauss Elimination

Pseudo code:

1. Start

2. Input the equations from user and parse it to form augmented Matrix(A)

3. Apply Gauss Elimination on Matrix A:

```
For i = 1 to n-1
    If Ai,i = 0
        Print("Divided by zero detected")
    Stop

    End If
    For j = i+1 to n
    Ratio = Aj,i/Ai,i
        For k = 1 to n+1
        Aj,k = Aj,k - Ratio * Ai,k
        Next k
   Next j
```

```
    Next i          //End of Elimination
    If A[n-1][n-1]==0 and A[n-1][n]==0
      Print("infinite number of solutions")
      Break
    ElseIf A[n-1][n-1]==0 and A[n-1][n]!=0
      Print("No solutions")
      Break
```

4. Obtaining Solution by Back Substitution:

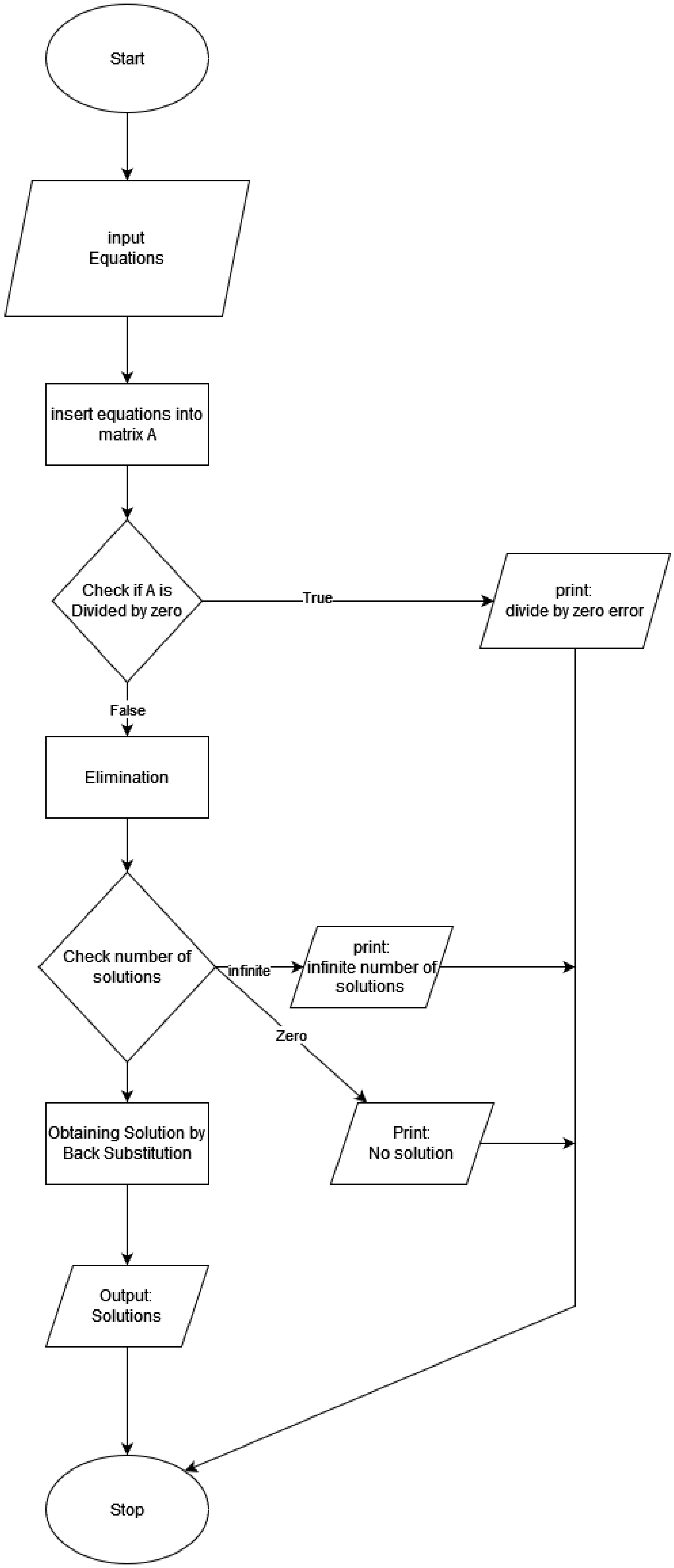
```
    Xn-1= A[n-1][n]/A[n-1][n-1]
    For i = n-2 to -1 (Step: -1)
        X[i] = A[i][n]
        For j = i+1 to n
         X[i] = X[i] – A[i][j] * X[j]
        Next j
        X[i] = X[i]/A[i][[i]
    Next i
```

5. Display Solution:

```
    For i = 1 to n
        Print X[i]
    Next i
```

6. Stop

Flowchart:


Start
input
Equations
insert equations into
matrix A
Check if A is
Divided by zero
True
print:
divide by zero error
False
Elimination
Check number of
solutions
infinite
print:
infinite number of
solutions
Zero
Print:
No solution
Obtaining Solution by
Back Substitution
Output:
Solutions
Stop

2] Gauss Seidel

Pseudo code:

1.Start

2. Input the equations from user and parse it to form augmented Matrix(A)

3. check if the input matrix is diagonally dominant matrix

```
    For i=0 to n
     Sum=0
     For j=0 to n
       Sum+=A[i][j]
     Sum-=A[i][i]
     If A[i][i]<Sum
       Print("the system of equations don't imply gauss seidel condition.(input matrix is not diagonal dominant"))
      Break
```

4. copy input matrix in a temp and perform forward elimination to check that the system of equations has a unique solutions

```
      For i = 1 to n
           If temp[i][i] = 0
               Print("Divided by zero detected")
           Stop
```

```
        End If
        For j = i+1 to n
                If i!=j and temp[n-1][n-1]!=0
          Ratio = temp[j][i]/temp[i][i]
          For k = 1 to n+1
            temp[j][k] = temp[j][k] - Ratio * temp[i][k]
          Next k
      Next j
    Next i          //End of Elimination
    If temp[n-1][n-1]==0 and temp[n-1][n]==0
      Print("infinite number of solutions")
      Break
    ElseIf temp[n-1][n-1]==0 and temp[n-1][n]!=0
      Print("No solutions")
      Break
```

5. Perform Gauss seidel method:

```
  For  j= 0 to n
   d=augmented matrix(b)[j]
    For i =0 to n
     If i!=j
```

Then d-=A[j][i]*x[i]

If A[j][j]==0

print("Division by zero is detected")

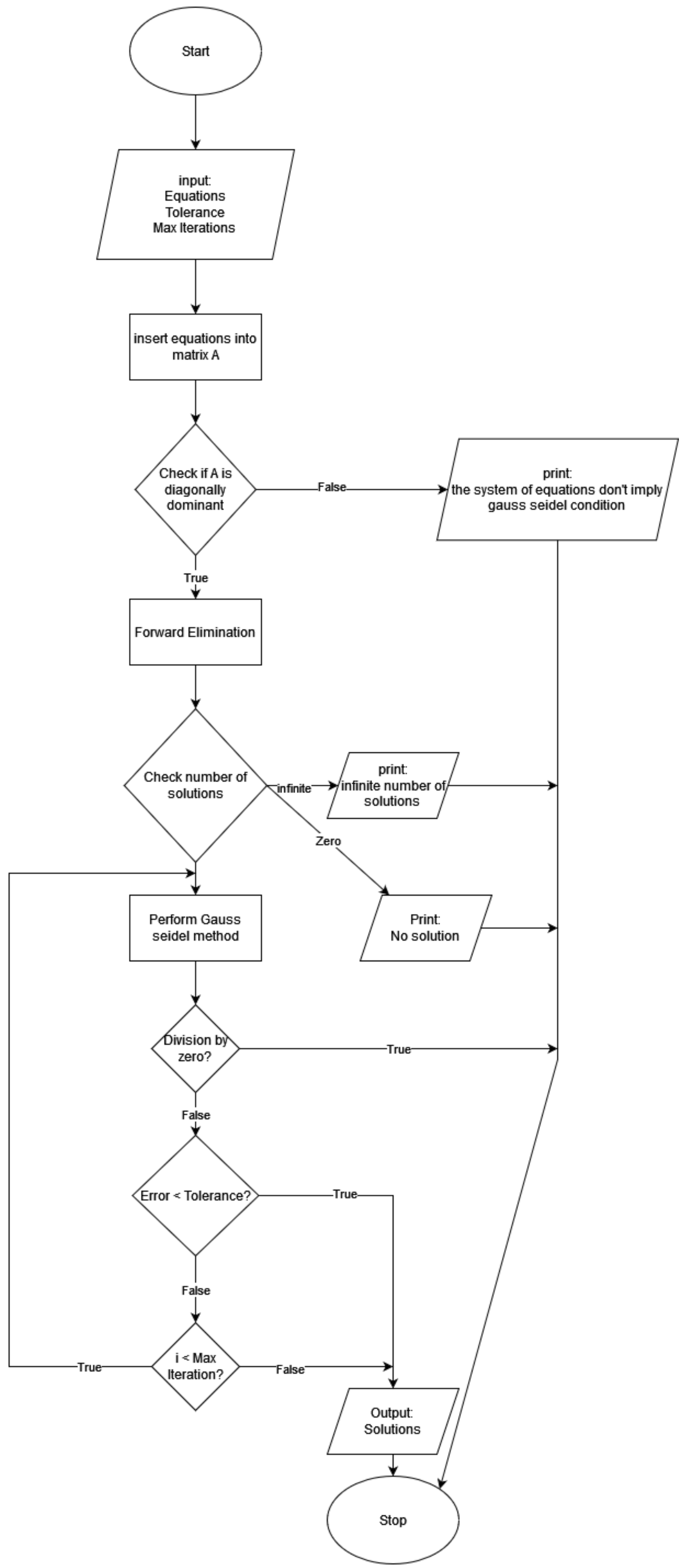
x[j]=d/A[j][j]

6. Stop If error is less than tolerance or the steps is greater than number of iterations given

7. Print value of x1, y1, z1 and so on

8. Stop

Flowchart:


Start
input:
Equations
Tolerance
Max Iterations
insert equations into matrix A
Check if A is diagonally dominant
False
print:
the system of equations don't imply gauss seidel condition
True
Forward Elimination
Check number of solutions
infinite
print:
infinite number of solutions
Zero
Print:
No solution
Perform Gauss seidel method
Division by zero?
True
False
Error < Tolerance?
True
False
i < Max Iteration?
True
False
Output:
Solutions
Stop

3] Gauss Jordan

Pseudo code:

1. Start

2. Input the equations from user and parse it to form augmented Matrix(A)

3. Apply Gauss Elimination on Matrix A:

```
    For i = 1 to n
        If A[i][i] = 0
          Print("Divided by zero detected")
        Stop

        End If
        For j = i+1 to n
              If i!=j and A[n-1][n-1]!=0
         Ratio = A[j][i]/A[i][i]
         For k = 1 to n+1
           A[j][k] = A[j][k] - Ratio * A[i][k]
         Next k
       Next j
     Next i        //End of Elimination
```

```
If A[n-1][n-1]==0 and A[n-1][n]==0
  Print("infinite number of solutions")
  Break
ElseIf A[n-1][n-1]==0 and A[n-1][n]!=0
  Print("No solutions")
  Break
```

4. Obtaining Solution:

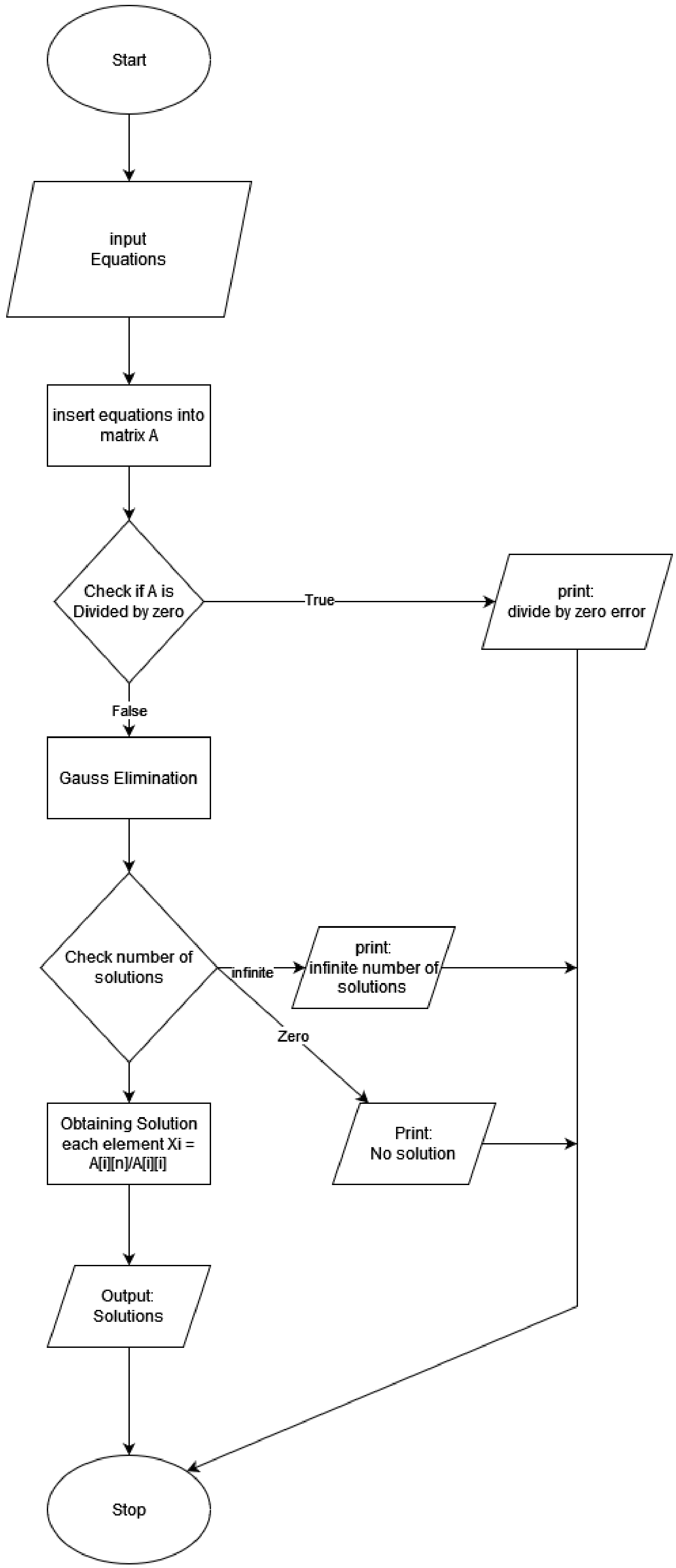
```
For i = 1 to n
Xi = A[i][n]/A[i][i]
Next i
```

5. Display Solution:

```
For i = 1 to n
    Print Xi
Next i
```

6. Stop

Flowchart:


Start
input
Equations
insert equations into matrix A
Check if A is Divided by zero
True
print:
divide by zero error
False
Gauss Elimination
Check number of solutions
infinite
print:
infinite number of solutions
Zero
Print:
No solution
Obtaining Solution each element Xi = A[i][n]/A[i][i]
Output:
Solutions
Stop

4] LU Decomposition

Pseudo code:

1. Start

2. Input the equations from user and parse it to form augmented Matrix(A)

3. copy input matrix in a temp and perform forward elimination to check that the system of equations has a unique solution

```
For i = 1 to n
    If temp[i][i] = 0
        Print("Divided by zero detected")
    Stop

    End If
    For j = i+1 to n
            If i!=j and temp[n-1][n-1]!=0
     Ratio = temp[j][i]/temp[i][i]
     For k = 1 to n+1
        temp[j][k] = temp[j][k] - Ratio * temp[i][k]
     Next k
  Next j
Next i          //End of Elimination
If temp[n-1][n-1]==0 and temp[n-1][n]==0
```

Print(“infinite number of solutions”)

Break

ElseIf temp[n-1][n-1]==0 and temp[n-1][n]!=0

Print(“No solutions”)

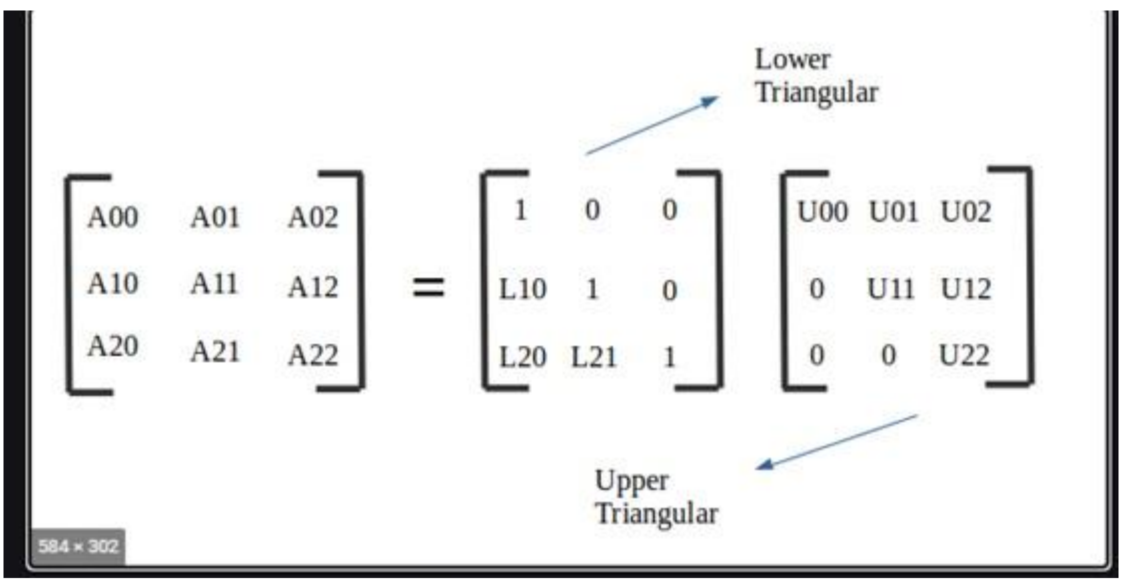
Break

4. Apply Lu decomposition on Matrix A:

a. Set the diagonal o Matrix L by 1

b. Apply forward elimination to obtain Matrix (U) and calculate the rest of the coefficient of (L) Matrix

c.



d. perform forward substitution Ly=b

For I =0 to n

Y[i]=b[i]

For k =0 to i

y[i]=y[i]-y[k]*L[i][k]

5. Perform backward substitution on Ux=y

For I =0 to n

X[i]=y[i]

For k =n-1 to i

j=k-1

x[i]=x[i]-x[j+1]*U[i][j+1]

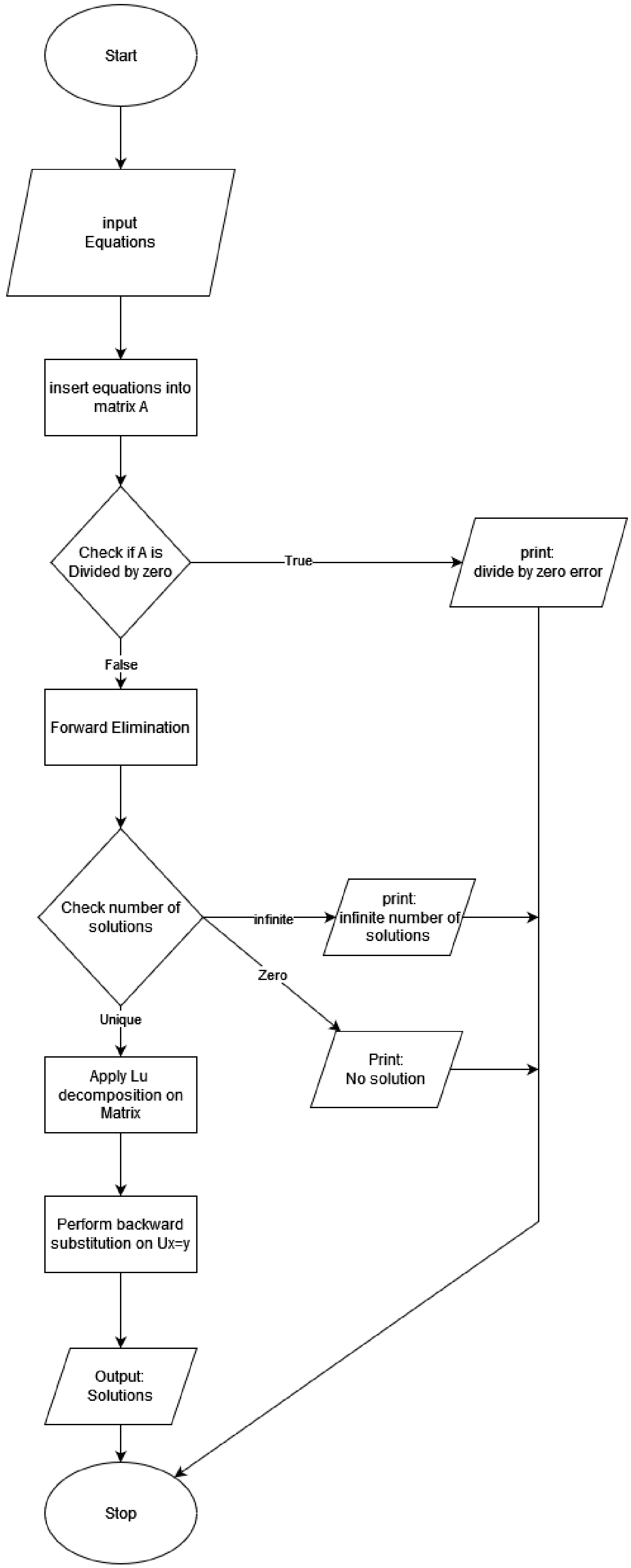
x[i]=x[i]/U[i][i]

6. Obtaining Solution:

Return Array x

7.Stop

Flowchart:



Start
input
Equations
insert equations into matrix A
Check if A is Divided by zero
True
print:
divide by zero error
False
Forward Elimination
Check number of solutions
infinite
print:
infinite number of solutions
Zero
Print:
No solution
Unique
Apply Lu decomposition on Matrix
Perform backward substitution on Ux=y
Output:
Solutions
Stop