NUMERICAL ANALYSIS

FINAL PROJECT 2

Lara HossamElDin Mostafa	6853
Nourine Mohamed AbouElWafa	7152
Nour Hesham Shaheen	7150
Sohayla Khaled Abouzeid	6851

Table of Contents

Project Description

The program consists of the implementation of two different numerical methods, with the aim of comparing the runtime performance of their algorithms against each other.

The two numerical methods are Gauss Elimination and Gauss Seidel.

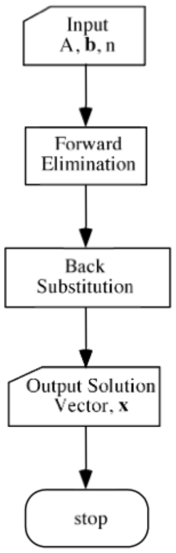
Numerical Methods

Gauss Elimination

Gauss Elimination Method is a procedure for solving systems of linear equations. It is also known as **Row Reduction Technique.** In this method, the problem of solving a system of linear equations having n unknown variables is transformed into a matrix having n rows and $n + 1$ columns. This matrix is also known as **Augmented Matrix**.

After forming $n \times (n + 1)$ augmented matrix, we transform it to an upper triangular matrix by elementary row operations **ERO**. This step is known as **Forward Elimination** Finally, we obtain our result by **Back Substitution**.

Flowchart



Pseudo Code

1. Start

2. Input the Augmented Coefficients Matrix (A):

```
for i = 1 to n:
     for j = 1 to n+1:
         Read Ai,j
     Next j
Next i
```

3. Apply Gauss Elimination on Matrix A:

```
for i = 1 to n-1:
    if Ai,i = 0:
        print "Mathematical Error!"
        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
```

4. Obtaining Solution by Back Substitution:

```
Xn = An,n+1/An,n
    for i = n-1 to 1 (Step: -1)
        Xi = Ai,n+1
        for j = i+1 to n
            Xi = Xi - Ai,j * Xj
        Next j
    Xi = Xi/Ai,i
    Next i
```

5. Stop

*

Matlab Code

```
for i = 1:n-1
    % Partial pivoting
    [~, pivot] = max(abs(A(i:n, i)));
    pivot = pivot + i - 1;
    if pivot ~= i
        A([i pivot],:) = A([pivot i],:);
        b([i pivot]) = b([pivot i]);
    end

    % Elimination
    for j = i+1:n
        factor = A(j,i) / A(i,i);
        A(j,:) = A(j,:) - factor * A(i,:);
        b(j) = b(j) - factor * b(i);
    end
end

% Back substitution
x = zeros(n, 1);
for i = n:-1:1
    x(i) = (b(i) - A(i,i+1:n)*x(i+1:n)) / A(i,i);
    str= str + sprintf(" var (%d) = %f\n",i,x(i));
end

% Display the solution
disp(x)
time = toc(now1);
set(handles.exec,'String',string(time));
set(handles.out,'String',str);
```

Problematic Cases

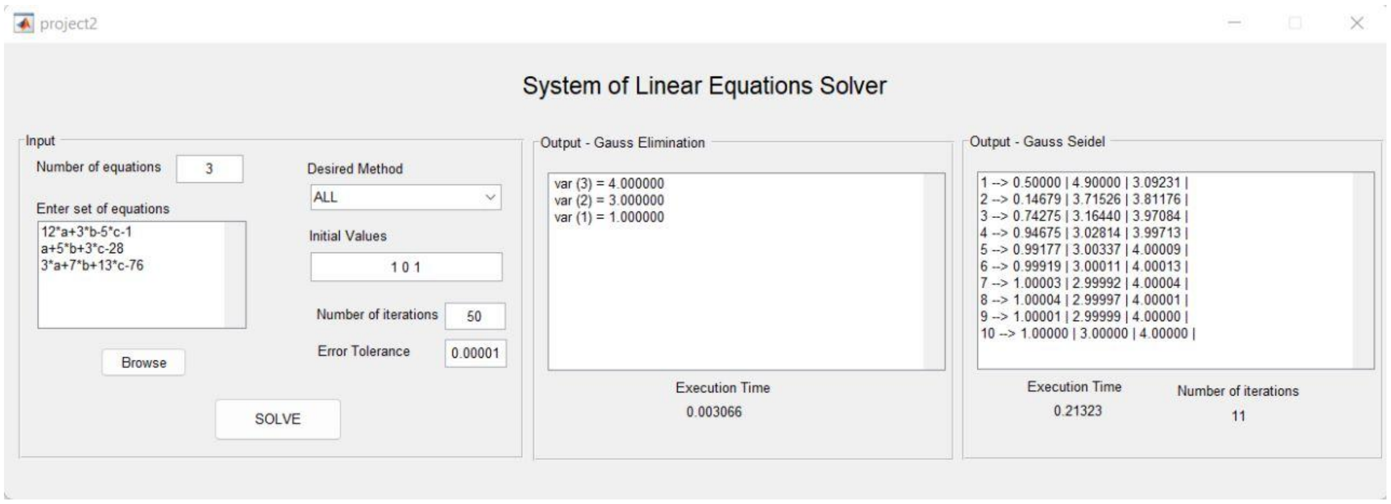
- **When the pivot of R1 = 0:**
  - During both the elimination and the back-substitution phases, it is possible that a **division by zero** can occur. Problems also can arise when a coefficient is very close to zero. The technique of pivoting has been developed to partially avoid these problems.

- **Solving N equations when N is around 100:**
  - The problem of **round-off error** can become particularly important when large numbers of equations are to be solved. This is due to the fact that every result is dependent on previous results. Consequently, an error in the early steps will tend to propagate—that is, it will cause errors in subsequent steps.

- **In an ill conditioned system:**
  - This is when small changes in coefficients result in large changes in the solution. An alternative interpretation of ill-conditioning is that a wide range of answers can approximately satisfy the equations. Because round-off errors can induce small changes in the coefficients, these artificial changes can lead to large solution errors for ill-conditioned systems.

Sample run

1.



Number of equations: 3

Enter set of equations:

```
12*a+3*b-5*c-1
a+5*b+3*c-28
3*a+7*b+13*c-76
```

Desired Method: ALL

Initial Values: 1 0 1

Number of iterations: 50

Error Tolerance: 0.00001

Output - Gauss Elimination

```
var (3) = 4.000000
var (2) = 3.000000
var (1) = 1.000000
```

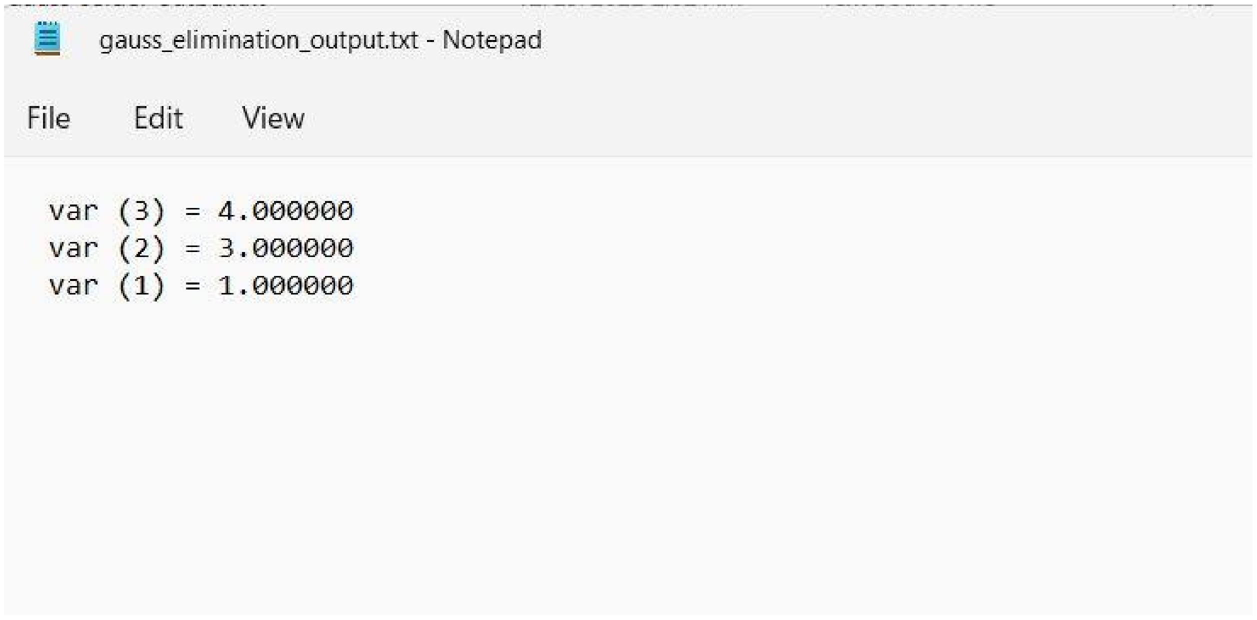
Execution Time: 0.003066

Output - Gauss Seidel

```
1 --> 0.50000 | 4.90000 | 3.09231 |
2 --> 0.14679 | 3.71526 | 3.81176 |
3 --> 0.74275 | 3.16440 | 3.97084 |
4 --> 0.94675 | 3.02814 | 3.99713 |
5 --> 0.99177 | 3.00337 | 4.00009 |
6 --> 0.99919 | 3.00011 | 4.00013 |
7 --> 1.00003 | 2.99992 | 4.00004 |
8 --> 1.00004 | 2.99997 | 4.00001 |
9 --> 1.00001 | 2.99999 | 4.00000 |
10 --> 1.00000 | 3.00000 | 4.00000 |
```

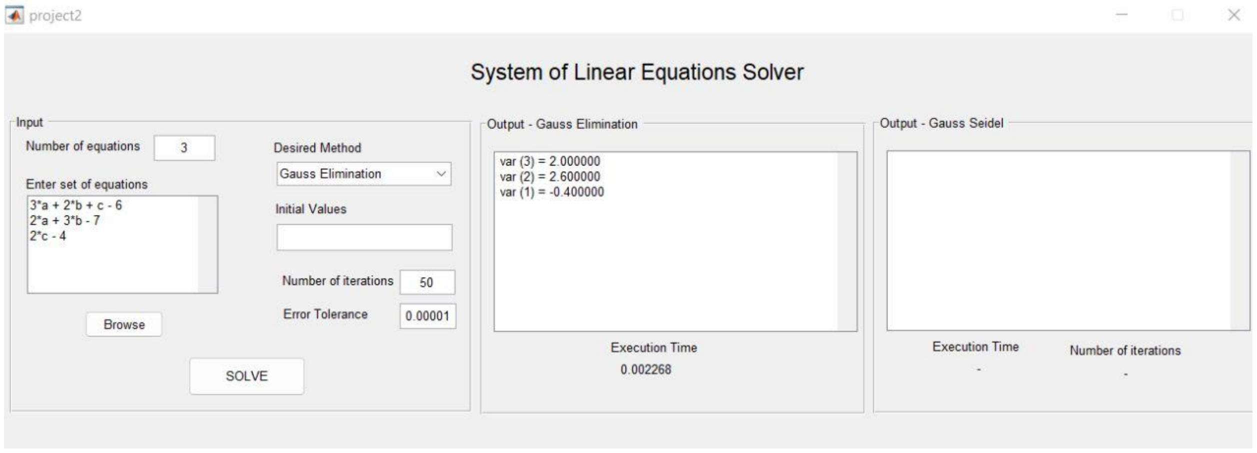
Execution Time: 0.21323

Number of iterations: 11



```
var (3) = 4.000000
var (2) = 3.000000
var (1) = 1.000000
```

2.



System of Linear Equations Solver

Input

Number of equations: 3

Enter set of equations

```
3*a + 2*b + c - 6
2*a + 3*b - 7
2*c - 4
```

Desired Method: Gauss Elimination

Initial Values:

Number of iterations: 50

Error Tolerance: 0.00001

Output - Gauss Elimination

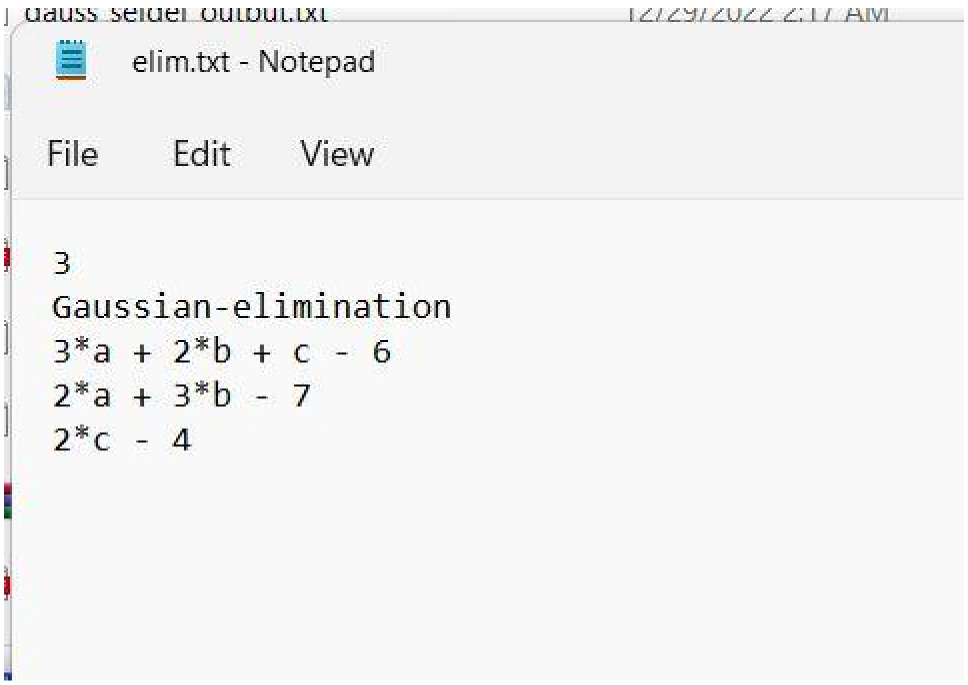
```
var (3) = 2.000000
var (2) = 2.600000
var (1) = -0.400000
```

Execution Time
0.002268

Output - Gauss Seidel

Execution Time
-

Number of iterations
-

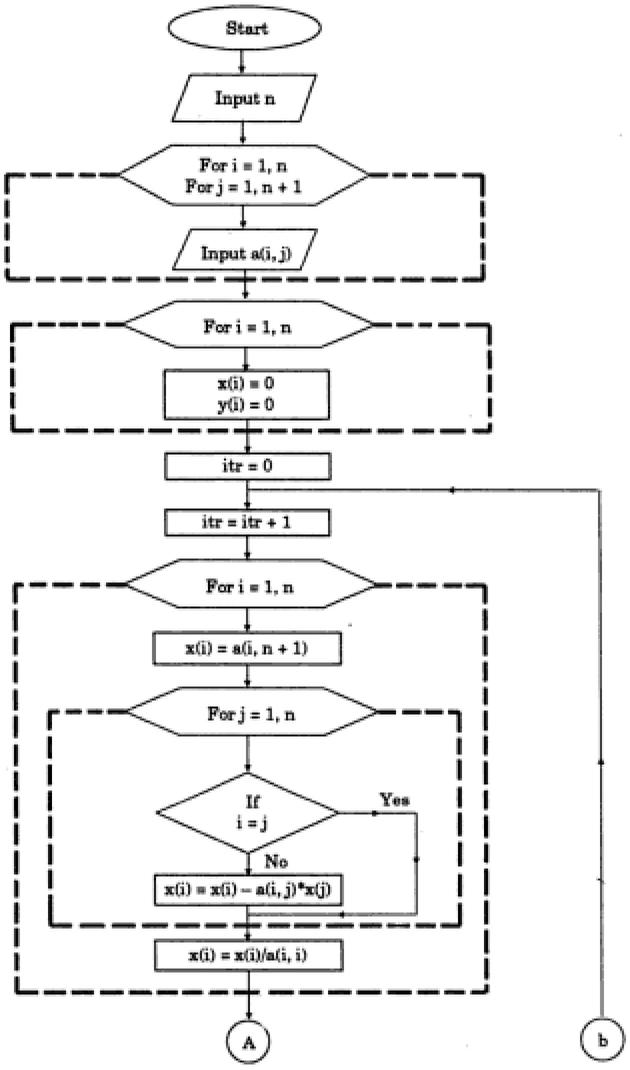


```
3
Gaussian-elimination
3*a + 2*b + c - 6
2*a + 3*b - 7
2*c - 4
```

Gauss seidel

The Gauss-Seidel method is an iterative approach for solving systems of linear equations. In this method, when first given a system of linear equations, we arrange them in diagonally dominant form. For guaranteed convergence, the system must be in Diagonally Dominant Form. Now, we can start the solution process by choosing guesses for the x's. A simple way to obtain initial guesses is to assume that they are all zero.

Flowchart



Pseudo Code

```
1. Start
2. Arrange given system of linear equations in diagonally dominant form
3. Read tolerable error (e)
4. Convert the first equation in terms of the first variable, second equation in terms of the second variable and so on.
5. Set initial guesses for x0, y0, z0 and so on
6. Substitute value of y0, z0 ... from step 5 in first equation obtained from step 4 to calculate new value of x1. Use x1, z0, u0 .... in the second equation obtained from step 4 to calculate the new value of y1. Similarly, use x1, y1, u0... to find new z1 and so on.
7. If| x0 - x1| > e and | y0 - y1| > e and | z0 - z1| > e and so on then go to step 9
8. Set x0=x1, y0=y1, z0=z1 and so on and go to step 6
9. Print value of x1, y1, z1 and so on
10. Stop
```

Matlab Code

```matlab
function gauss_seidel(handles,A,b,maxiter,tol,init)
now1 = tic();
% Case gauss-seidel:
disp("in gauss-seidel");
str = "";
n = length(str2num(init));
disp(n)
disp(A)
disp(b)

% Set the convergence tolerance and maximum number of iterations
maxiter = str2num(maxiter);
% Perform the Gauss-Seidel iterations
% Set the initial guess for the solution
tokens = split(init, ' ');
x = cellfun(@str2double, tokens);
```

```
error  = 100;
iter = 1
disp("This is tolerance")
disp(tol)
res = zeros(n,maxiter);
temp = [];
while(iter < maxiter & error > str2double(tol))
   str = str+ sprintf("%d -->",iter);
   xold = x;
   disp(xold)
   for i = 1:n
      sigma = 0;
      for j = 1:n
         if j ~= i
            sigma = sigma + A(i,j)*x(j);
         end
      end
      x(i) = (b(i) - sigma)/A(i,i);
      temp(i) = abs((x(i)-xold(i))/x(i));
   end
   disp(temp)
   error = max(temp);
   disp(error)
   for k = 1:n
        str = str + sprintf(" %.5f |",x(k));
        res(k,iter) = x(k);
   end
       str = str + sprintf("\n");
       iter = iter + 1;
end

% Curve between the number of iterations and the obtained root value at
this
% iteration for all the methods in the same graph.
s = "";
for i=1:n
    row = res(i, :);
    s = sprintf("Var %d",i);
    figure('Name',s,'NumberTitle','off')
    plot(row(1:iter));
    hold on;
    legend(s)
```

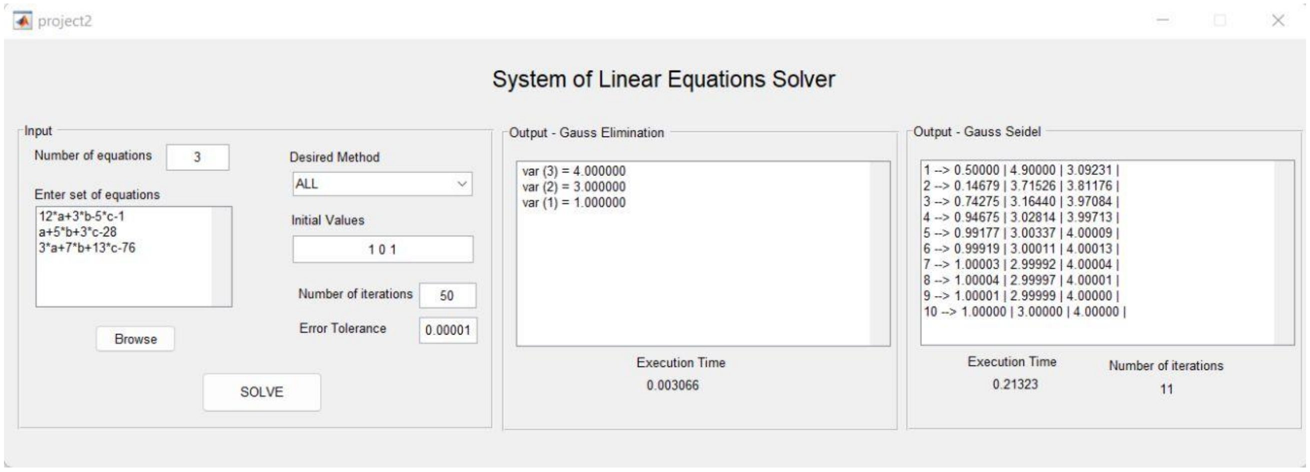
```
end

% Print the solution and number of iterations
time = toc(now1);
set(handles.exec2,'String',string(time));
set(handles.out2,'String',str);
set(handles.num_iter2,'String',string(iter));

% Open a file for writing
fid = fopen('gauss_seidel_output.txt', 'w');

% Write the text to the file
fprintf(fid, '%s', str);

% Close the file
fclose(fid);
```

Problematic Cases

- **Divergence:**
  - A pitfall of the Gauss-Seidel method: not all systems of equations will converge. Only systems that have a diagonally dominant coefficient matrix can converge.
- **In large nonbanded systems:**
  - Inefficient in time and memory.

Sample run

1.



Input

Number of equations: 3

Enter set of equations:

```
12*a+3*b-5*c-1
a+5*b+3*c-28
3*a+7*b+13*c-76
```

Desired Method: ALL

Initial Values: 1 0 1

Number of iterations: 50

Error Tolerance: 0.00001

Output - Gauss Elimination

```
var (3) = 4.000000
var (2) = 3.000000
var (1) = 1.000000
```

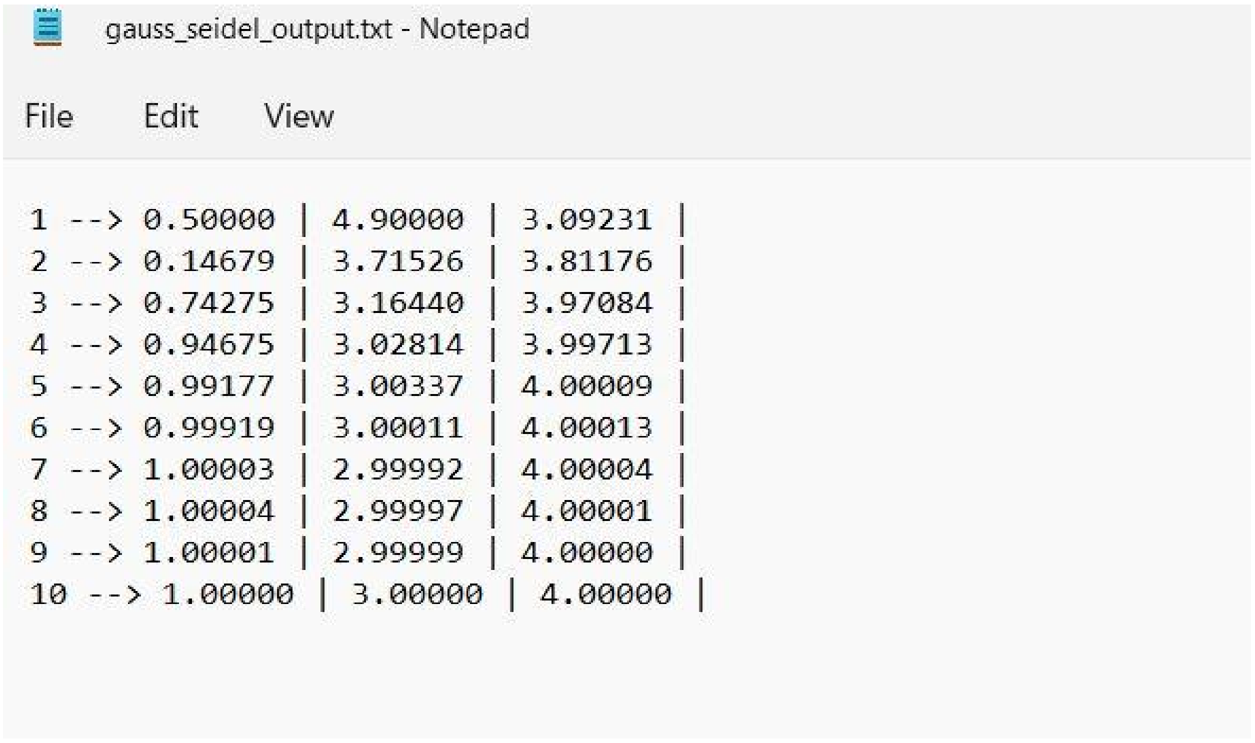
Execution Time
0.003066

Output - Gauss Seidel

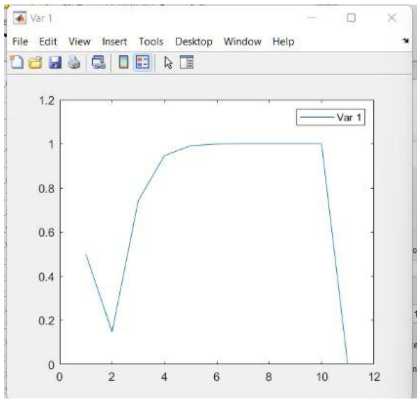
```
1 --> 0.50000 | 4.90000 | 3.09231 |
2 --> 0.14679 | 3.71526 | 3.81176 |
3 --> 0.74275 | 3.16440 | 3.97084 |
4 --> 0.94675 | 3.02814 | 3.99713 |
5 --> 0.99177 | 3.00337 | 4.00009 |
6 --> 0.99919 | 3.00011 | 4.00013 |
7 --> 1.00003 | 2.99992 | 4.00004 |
8 --> 1.00004 | 2.99997 | 4.00001 |
9 --> 1.00001 | 2.99999 | 4.00000 |
10 --> 1.00000 | 3.00000 | 4.00000 |
```

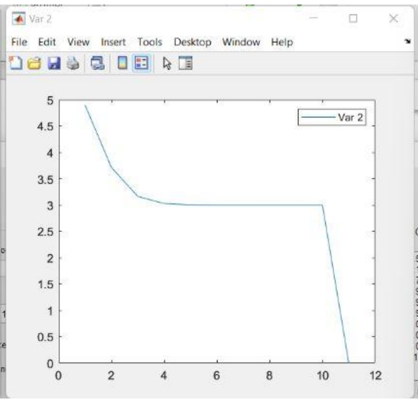
Execution Time
0.21323

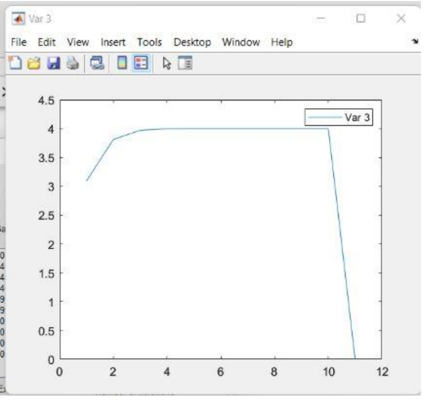
Number of iterations
11



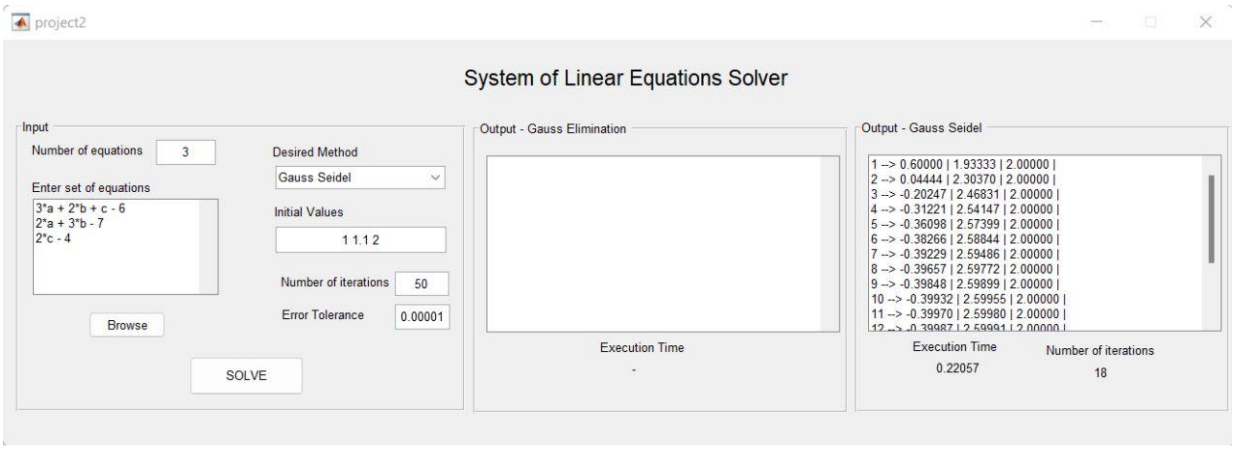
```
1 --> 0.50000 | 4.90000 | 3.09231 |
2 --> 0.14679 | 3.71526 | 3.81176 |
3 --> 0.74275 | 3.16440 | 3.97084 |
4 --> 0.94675 | 3.02814 | 3.99713 |
5 --> 0.99177 | 3.00337 | 4.00009 |
6 --> 0.99919 | 3.00011 | 4.00013 |
7 --> 1.00003 | 2.99992 | 4.00004 |
8 --> 1.00004 | 2.99997 | 4.00001 |
9 --> 1.00001 | 2.99999 | 4.00000 |
10 --> 1.00000 | 3.00000 | 4.00000 |
```


Var 1
File Edit View Insert Tools Desktop Window Help
Var 1
1.2
1
0.8
0.6
0.4
0.2
0
0 2 4 6 8 10 12


Var 2
File Edit View Insert Tools Desktop Window Help
Var 2
5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
0 2 4 6 8 10 12


Var 3
File Edit View Insert Tools Desktop Window Help
Var 3
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
0 2 4 6 8 10 12

2.



Number of equations: 3

Enter set of equations:

```
3*a + 2*b + c - 6
2*a + 3*b - 7
2*c - 4
```

Desired Method: Gauss Seidel

Initial Values: 1 1.1 2

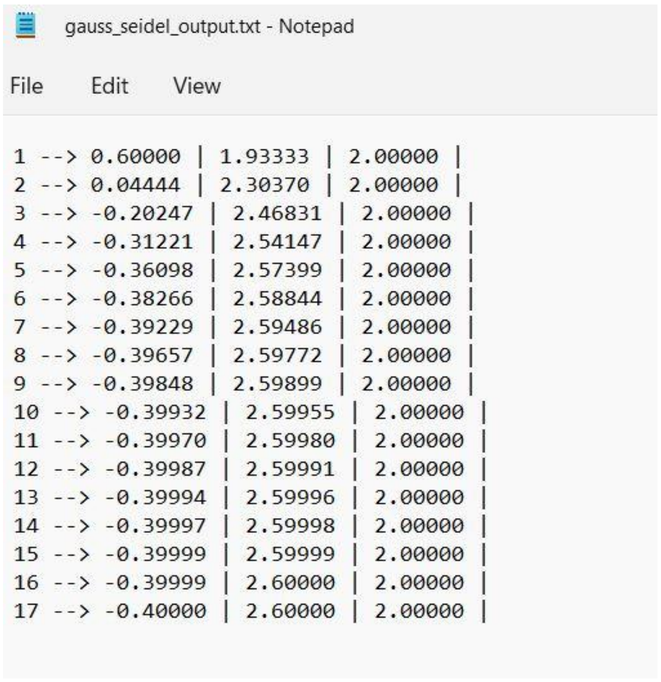
Number of iterations: 50

Error Tolerance: 0.00001

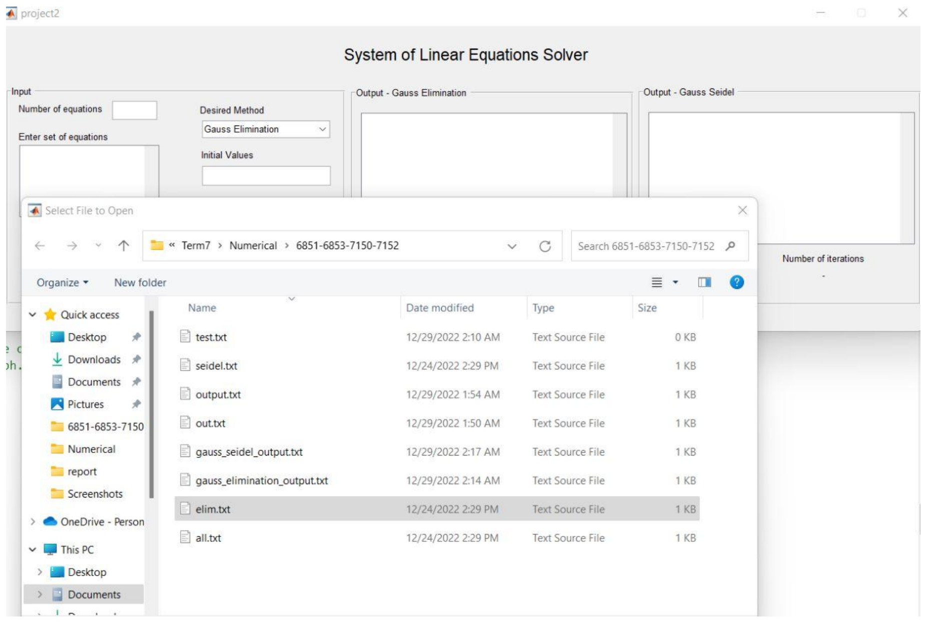
```
1 --> 0.60000 | 1.93333 | 2.00000 |
2 --> 0.04444 | 2.30370 | 2.00000 |
3 --> -0.20247 | 2.46831 | 2.00000 |
4 --> -0.31221 | 2.54147 | 2.00000 |
5 --> -0.36098 | 2.57399 | 2.00000 |
6 --> -0.38266 | 2.58844 | 2.00000 |
7 --> -0.39229 | 2.59486 | 2.00000 |
8 --> -0.39657 | 2.59772 | 2.00000 |
9 --> -0.39848 | 2.59899 | 2.00000 |
10 --> -0.39932 | 2.59955 | 2.00000 |
11 --> -0.39970 | 2.59980 | 2.00000 |
12 --> -0.39987 | 2.59991 | 2.00000 |
```

Execution Time: 0.22057

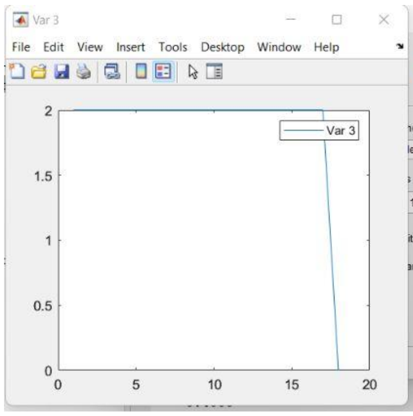
Number of iterations: 18

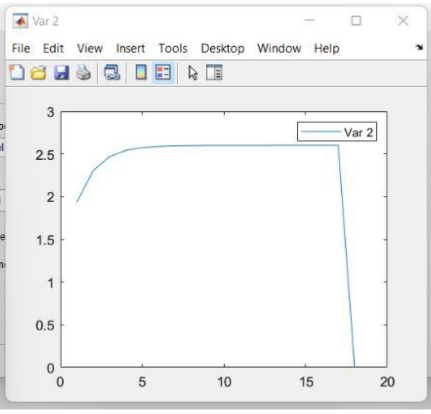


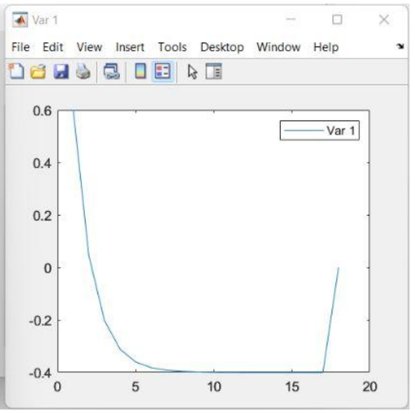
```
1 --> 0.60000 | 1.93333 | 2.00000 |
2 --> 0.04444 | 2.30370 | 2.00000 |
3 --> -0.20247 | 2.46831 | 2.00000 |
4 --> -0.31221 | 2.54147 | 2.00000 |
5 --> -0.36098 | 2.57399 | 2.00000 |
6 --> -0.38266 | 2.58844 | 2.00000 |
7 --> -0.39229 | 2.59486 | 2.00000 |
8 --> -0.39657 | 2.59772 | 2.00000 |
9 --> -0.39848 | 2.59899 | 2.00000 |
10 --> -0.39932 | 2.59955 | 2.00000 |
11 --> -0.39970 | 2.59980 | 2.00000 |
12 --> -0.39987 | 2.59991 | 2.00000 |
13 --> -0.39994 | 2.59996 | 2.00000 |
14 --> -0.39997 | 2.59998 | 2.00000 |
15 --> -0.39999 | 2.59999 | 2.00000 |
16 --> -0.39999 | 2.60000 | 2.00000 |
17 --> -0.40000 | 2.60000 | 2.00000 |
```



Name	Date modified	Type	Size
test.txt	12/29/2022 2:10 AM	Text Source File	0 KB
seidel.txt	12/24/2022 2:29 PM	Text Source File	1 KB
output.txt	12/29/2022 1:54 AM	Text Source File	1 KB
out.txt	12/29/2022 1:50 AM	Text Source File	1 KB
gauss_seidel_output.txt	12/29/2022 2:17 AM	Text Source File	1 KB
gauss_elimination_output.txt	12/29/2022 2:14 AM	Text Source File	1 KB
elim.txt	12/24/2022 2:29 PM	Text Source File	1 KB
all.txt	12/24/2022 2:29 PM	Text Source File	1 KB







Behavior Analysis

	Gauss elimination	Gauss seidel
Execution Time	0.002268 s	0.22057 s
Number of iterations	-	18