LAB EXPERIMENT # 4: Root finding using Gauss Elimination method

Name & Roll No.: Ashraf Al- Khalique, 1801171	**Grade (20)**
Registration Number: 351	
Lab Section: C-2	

4.1 Objectives

- To determine roots of linear equations using ‘Gauss Elimination method’.
- To understand the MATLAB implementation of the ‘Gauss Elimination method’.
- To analyze of results using different initial values and different ranges of error.

4.2 Theory
4.2.1 Gauss Elimination method

The Gauss elimination method is an approach for solving linear equation systems, determining the rank of a matrix, and computing the inverse of an invertible square matrix. Gaussian elimination is named after Carl Friedrich Gauss, a German mathematician and physicist.

Gauss elimination is an exact method for solving a given system of equations in n unknowns by changing the coefficient matrix into an upper triangular matrix and then back substituting for the unknowns.

Gaussian elimination is mainly divided into two steps.

The first phase (Forward Elimination) either reduces a given system to triangular or echelon form, or produces a degenerate equation with no solution, indicating that the system has no solution. This is accomplished by employing elementary.

The second step employs back substitution to solve the above-mentioned system. The first section reduces a matrix to row echelon form using elementary row operations, while the second section reduces it to reduced row echelon form, also known as row canonical form.

4.3 Apparatus
4.3.1 MATLAB

4.4 Gauss Elimination method
4.4.1 Algorithm

Step: 1 Define *Matrix*

Step: 2 Read the number of unknowns: *n*

Step: 3 Read augmented *Matrix (A)* of *n* by *n+1* size

Step: 4 Transform *Augmented Matrix (A)* to Upper Triangular Matrix by

Row Operations.

Step: 5 Obtain Solution by Back Substitution.

Step: 6 Display result.

Step: 7 Stop.

4.4.2 Pseudocode

```
1. Start
2. Input the augmented coefficient 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. Display Solution:
   For i = 1 to n
         Print Xi
   Next i
6. Stop
```

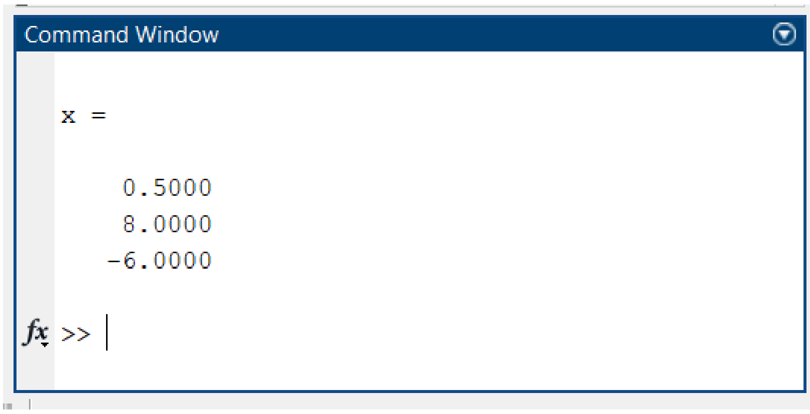
4.4.3 MATLAB Code

```
clc;
close all;
A = [10 2 -1;-3 -6 2;1 1 5];
B = [27;-61.5;-21.5];
```

```
N = length(B);
x=zeros(N,1);
Aug = [ A  B ];
for j = 1: N-1
        for i= j+1 : N
            m = Aug(i, j)/Aug(j, j) ;
            Aug (i, :)= Aug(i,:) - m*Aug(j,:);

        end
end
 Aug;
x(N) = Aug(N, N+1)/Aug(N, N);
for k = N-1:-1:1
    x(k) = (Aug(k, N+1) - Aug(k, k+1:N)*x(k+1:N))/Aug(k, k);
end
x
```

4.4.4 MATLAB Output



4.5 Discussion & Analysis

- **What are the pitfalls of the gauss elimination method?**

Pitfall occurs when one of the variables of *x (x1, x2, x3)* in any equation derived from matrix is missing, and if the round off error is very high.

- **What is the effect of pitfalls?**

Pitfalls resort to a situation where one of the variables goes missing. Thus, gauss elimination method doesn't do much of help.

- **How to overcome pitfalls of Gauss Elimination Method?**
    - **Partial pivoting**

If the divided by zero occurs in the first equation, then that equation is interchanged with the equation no. 2 or 3.

    - **Complete pivoting**

If all the coefficients of the x1 are small then the x1 might be changed with **x2** or *x3*. The ultimate goal is to take the pivot equation in such way that the coefficient of x1 should be large enough.

For the given matrix, there was a minor variation between the calculated numbers and the values got from the MATLAB solution. Because MATLAB has rounded the values. So, we know the inaccuracy here as round off error, and it is insignificant. Thus, there is no chance pitfalls.