**LAB EXPERIMENT # 7:** Compare the solution of a linear system obtained using ‘Gauss Elimination method’ & ‘Jacobi Method’

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

## 7.1 Objectives

- To determine roots of linear equations using ‘Gauss Elimination method’
- To determine Solution of a linear system using ‘Jacobi Method’
- To understand the MATLAB implementation of the ‘Gauss Elimination Method’ & ‘Jacobi Method’
- To analyze of results using different initial values and different ranges of error

## 7.2 Theory

### 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. The second step employs back substitution to solve the above-mentioned system.

### Jacobi method

In numerical linear algebra, the Jacobi method is an iterative algorithm for determining the solutions of a strictly diagonally dominant system of linear equations. Each diagonal element is solved for, and an approximate value is plugged in. The process is then iterated until it converges. This algorithm is a stripped-down version of the Jacobi transformation method of matrix diagonalization. The method is named after Carl Gustav Jacob Jacobi.

## 7.3 Apparatus

- MATLAB

## 7.4 Algorithm

## Gauss elimination method

**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.

## Jacobi method

**Step: 1** Start

**Step: 2** Arrange the given linear equation system in diagonally dominating form.

**Step: 3** Read tolerable error (e)

**Step: 4** Convert the first equation to the first variable, the second equation to the second variable, and so on.

**Step: 5** Set initial guesses for x0, y0, z0 and so on

**Step: 6** Substitute value of x0, y0, z0 ... from step 5 in equation obtained in step 4 to calculate new values x1, y1, z1 and so on

**Step: 7** If| x0 - x1| > e and | y0 - y1| > e and | z0 - z1| > e and so on then go to step 9

**Step: 8** Set x0=x1, y0=y1, z0=z1 and so on and go to step 6

**Step: 9** Print value of x1, y1, z1 and so on

**Step: 10** Stop.

## Comparison

**Step: 1** If output of gauss elimination method is equal to output of Jacobi method, print “Both methods provide equal output” if not, print “Both methods don’t provide equal output”

## 7.5 Pseudocode

## Gauss elimination method

```
Start
Input the augmented coefficient matrix (A):
For i = 1 to n
      For j = 1 to n+1
            Read Ai,j
            Next j
      Next i
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
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
Display Solution:
      For i = 1 to n
            Print Xi
            Next i
Stop
```

## Jacobi method

```
Start
Define A, b,x;
Integer i, j, k, n, delta = 0.001
n = size(A), Kmax
For k = 1 to kmax do
      Y = x
      For i = 1 to n do
      Sum = bi
      Diag = Aii
      If |diag|< delta
      Then
            Output "Diagonal element too small"
      Return
End if
      For j = 1 to n do
            If j ≠ i then
                  Sum = sum – Aij.yj
            End If
      End for
      Xi = sum/diag
End For
Output k,x
If |x-y|<e
            Then ouput k, x
            Return
```

```
    End if
End for
Stop
```

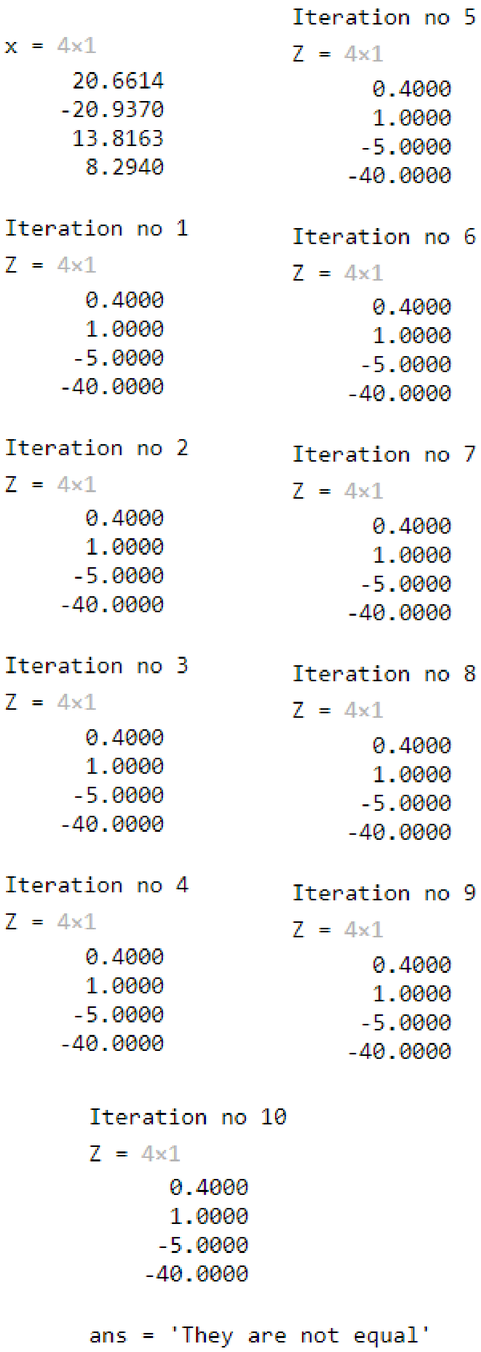
### 7.6 MATLAB Code

```matlab
%gauss elimination method
clc;
close all;
A = [25 30 10 -2;10 20 15 3;15 12 -4 2;1 0 2 -1];  %Coefficient matrix
B = [10;20;20;40];
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
```

```matlab
%jacobi method
I= [0;0;0;0]; %Initial guess
itr= input('No of Iteration: '); %no of iteration
e= input('No of tolerance: '); %tolerance
L=length(B);
Z=zeros(L,1);
for j=1:itr
for i=1:L
Z(i)=(B(i)/A(i,i))-(A(i,[1:i-1,i+1:L])*I([1:i-1,i+1:L]))/A(i,i);
end
fprintf('\nIteration no %d',j)
Z
if abs(Z-I)<e
break
end
end
I=Z;
```

```matlab
if Z==x
    'Both method provide equal output'
else
    'Gauss Elimination & jacobi method provides different output'
end
```

### 7.7 MATLAB Output



```
x = 4×1
     20.6614
    -20.9370
     13.8163
      8.2940

Iteration no 1
Z = 4×1
      0.4000
      1.0000
     -5.0000
    -40.0000

Iteration no 2
Z = 4×1
      0.4000
      1.0000
     -5.0000
    -40.0000

Iteration no 3
Z = 4×1
      0.4000
      1.0000
     -5.0000
    -40.0000

Iteration no 4
Z = 4×1
      0.4000
      1.0000
     -5.0000
    -40.0000

Iteration no 5
Z = 4×1
      0.4000
      1.0000
     -5.0000
    -40.0000

Iteration no 6
Z = 4×1
      0.4000
      1.0000
     -5.0000
    -40.0000

Iteration no 7
Z = 4×1
      0.4000
      1.0000
     -5.0000
    -40.0000

Iteration no 8
Z = 4×1
      0.4000
      1.0000
     -5.0000
    -40.0000

Iteration no 9
Z = 4×1
      0.4000
      1.0000
     -5.0000
    -40.0000

Iteration no 10
Z = 4×1
      0.4000
      1.0000
     -5.0000
    -40.0000

ans = 'They are not equal'
```

### 7.8 Discussion & Analysis

In gauss elimination method, root finding method is used. In Jacobi method, iteration procedure is followed where the initial value of each root is assumed to be zero, and the roots are substituted in the equations again to obtain the roots for the second iteration. The second iteration's root was then substituted for the third iteration, and so on until two consecutive equations' roots were equivalent. Here, output obtained from gauss elimination method, and output obtained from Jacobi method are not equal. It is because the given linear equation does not fulfill the condition for Jacobi method. The condition is that the co-efficient matrix should be diagonally dominant. A diagonally dominant matrix is one in which the magnitude (without considering signs) of the diagonal term in each row is greater than the sum of the other elements in that row.