LAB EXPERIMENT # 5: Solution of a linear system using 'Jacobi Method'

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

5.1 Objectives

- To determine Solution of a linear system using 'Jacobi Method'
- To understand the MATLAB implementation of the 'Jacobi Method'
- To analyze of results using different initial values and different ranges of error.

5.2 Theory

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.

Set of equations:

$$10x_1+2x_2-x_3=27$$
$$-3x_1-6x_2+2x_3= - 61.5$$
$$x_1+x_2+5x_3= - 21.5$$

solving by Jacobi Method,

$$x_1^{(k+1)} = (1/10)\ (27 - 2x_2^{(k)} + x_3^{(k)})$$
$$x_2^{(k+1)} = (1/-6)\ (-61.5 + 3x_1^{(k)} - 2x_3^{(k)})$$
$$x_3^{(k+1)} = (1/5)\ (-21.5 - x_1^{(k-1)} - x_2^{(k+1)})$$

By putting k=1,2, 3…. we can get the 1st ,2nd,3rd……. iterations. We put the roots from iteration 1 in 2nd iteration and get the root of 2nd iteration and so on until we can the same root from consecutive two iterations.

5.3 Apparatus

- MATLAB

5.4 Algorithm

<u>Step: 1</u> Start

<u>Step: 2</u> Arrange the given linear equation system in diagonally dominating form.

<u>Step: 3</u> 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 x_0, y_0, z_0 and so on

Step: 6 Substitute value of x_0, y_0, z_0 ... from step 5 in equation obtained in step 4 to calculate new values x_1, y_1, z_1 and so on

Step: 7 If$| x_0 - x_1| > e$ and $| y_0 - y_1| > e$ and $| z_0 - z_1| > e$ and so on then go to step 9

Step: 8 Set $x_0=x_1$, $y_0=y_1$, $z_0=z_1$ and so on and go to step 6

Step: 9 Print value of x_1, y_1, z_1 and so on

Step: 10 Stop.

5.5 Pseudocode

```
1. Start
2. Define A, b,x;
3. Integer i, j, k, n, delta = 0.001
4. n = size(A), Kmax
5. For k = 1 to kmax do
         Y = x
         For i = 1 to n do
               Sum = bi
               Diag = Aii
6. If |diag|< delta
         Then
         Output "Diagonal element too small"
         Return

   End if
7. For j = 1 to n do

         If j ≠ i then
         Sum = sum - Aij.yj
   End If
   End for
8. Xi = sum/diag
         End For
         Output k,x
9. If |x-y|<e
         Then ouput k, x
               Return

         End  if
   End for
10.      Stop
```

5.6 MATLAB Code

```
clc;
close all;
A= [10 2 -1;-3 -6 2;1 1 5]; %Coefficient matrix
B= [27;-61.5;-21.5]; %Source vector
I= [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
I=Z;
end
```

5.7 MATLAB Output

```
Command Window

A =

    10     2    -1
    -3    -6     2
     1     1     5


B =

   27.0000
  -61.5000
  -21.5000

No of Iteration: 20
No of tolerance: .0001
```

5.8 Discussion & Analysis

Numerical methods are strategies for formulating mathematical problems so that they can be solved using arithmetic operations. They invariably require a huge number of time-consuming arithmetic calculations. There are numerous ways for solving a linear system with multiple linear equations. One of them is the Jacobi technique, which is an iterative procedure. In this case, 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.

- **How does the choice of the initial guess affect the solution?**

For the first iteration of the Jacobi technique, the initial guess is all the root zero. It facilitates the next stage of iteration.

- **Discuss the convergence of the Jacobi Method**

The Jacobi method's convergence is substantially slower than the Gauss Seidel methods. Because it first discovers the iteration's root and uses it in the next iteration. It took ten iterations to reach our tolerance solution here.

- **How the speed of convergence and the error tolerance are related?**

According to the results of the experiment, if the tolerance is higher, we will require fewer iterations to obtain the desired roots. For example, when we set the tolerance to.001, we received the necessary roots on the seventh iteration. However, for tolerance.0001, we obtained the desired root after the tenth repetition. As a result, the speed of convergence is inversely related to the tolerance for mistake.