DELHI TECHNOLOGICAL UNIVERSITY

SCIENTIFIC COMPUTING
(MC-204)
PRACTICAL FILE

SUBMITTED TO:
PROF. DINESH UDAR
MR. ANKIT SHARMA

SUBMITTED BY:
NITYA MITTAL
(2K19/MC/089)

EXPERIMENT 2

AIM

Write the MATLAB program of Gauss Jacobi method for solving the system of linear equations.

THEORY

The Jacobi method is a method of solving a matrix equation on a matrix that has no zeros along its main diagonal (Bronshtein and Semendyayev 1997, p. 892). Each diagonal element is solved for, and an approximate value 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 Jacobi method is easily derived by examining each of the n equations in the linear system of equations $\mathbf{A}\mathbf{x} = \mathbf{b}$ in isolation. If, in the ith equation

$$\sum_{j=1}^{n} a_{ij} x_j = b_i,$$

solve for the value of x_i while assuming the other entries of x remain fixed. This gives

$$x_i^{(k)} = \frac{b_i - \sum_{j \neq i} a_{ij} x_j^{(k-1)}}{a_{ii}},$$

which is the Jacobi method.

In this method, the order in which the equations are examined is irrelevant, since the Jacobi method treats them independently. The definition of the Jacobi method can be expressed with matrices as

$$\mathbf{x}^{(k)} = \mathbf{D}^{-1}(\mathbf{L} + \mathbf{U})\mathbf{x}^{(k-1)} + \mathbf{D}^{-1}\mathbf{b},$$

where the matrices $\mathbf{D}$, $-\mathbf{L}$, and $-\mathbf{U}$ represent thediagonal, strictly lower triangular, and strictly upper triangular parts of $\mathbf{A}$, respectively.

SOURCE CODE

```
%gauss jacobi method
clear all;
close all;


%the function
%20x + y - 2z = 17
%3x + 20y - z = -18
%2x - 3y + 20z = 25
disp("The equations:")
disp("20x + y - 2z = 17")
```

```
disp("3x + 20y - z = -18")
disp("2x - 3y + 20z = 25")


%the matrix
A=[20,1,-2;3,20,-1;2,-3,20];


%check variable initialized
check=1;


%checking for diagonally dominant matrix
for i=1:3
    sum=0;
    for j=1:3
        if(i~=j)
            sum=sum+abs(A(i,j));
        end
        if(abs(A(i,i))<sum)
            check=0;
            break;
        end
    end
end
%if matrix is diagonally dominant
if(check==1)
    disp("The matrix is diagonally dominant")
    %substitution for x y and z
    syms x y z;
    fx(y,z)=(17 - y + 2*z)/20;
    fy(x,z)=(-18-3*x+z)/20;
    fz(x,y)=(25-2*x+3*y)/20;


    %initial values
    x0=0;
    y0=0;
    z0=0;

    %applying the iteration 6 times
    for i=1:6
        x1=fx(y0,z0);
        y1=fy(x0,z0);
        z1=fz(x0,y0);
        x0=x1;
        y0=y1;
        z0=z1;
    end
```

```
    %rounding of the numbers
    X=round([x1;y1;z1]);


    disp('The column matrix X=[x;y;z] ')
    disp(X)



else
    disp("The matrix is not diagonally dominant")
end
```

OUTPUT

Case 1: Matrix is Diagonally Dominant

```
>> gauss_jacobi
The equations:
20x + y - 2z = 17
3x + 20y - z = -18
2x - 3y + 20z = 25
The matrix is diagonally dominant
The column matrix X=[x;y;z]
 1
-1
 1
```

Case 2: Matrix is NOT Diagonally Dominant

```
>> gauss_jacobi
The equations:
x + y - 2z = 17
3x + y - z = -18
2x - 3y + z = 25
The matrix is not diagonally dominant
```