NUMERICAL METHODS FOR ENGINEERS

Project Report

Topic: Solution of System of Equations

Method: Gauss Jordan Elimination

Submitted by

Adibhatla Sreekar Sashaank

B130579CS

sashaankadibhatla@gmail.com

1. Introduction:

The following row operations on the augmented matrix of a system produce the augmented matrix of an equivalent system, i.e., a system with the same solution as the original one.

- Interchange any two rows.
- Multiply each element of a row by a nonzero constant.
- Replace a row by the sum of itself and a constant multiple of another row of the matrix. For these row operations, we will use the following notations.
- Ri ↔ Rj means: Interchange row i and row j.
- αRi means: Replace row i with α times row i.
- Ri + αRj means: Replace row i with the sum of row i and α times row j.

The Gauss-Jordan elimination method to solve a system of linear equations is described in the following steps.

1. Write the augmented matrix of the system.

2. Use row operations to transform the augmented matrix in the form described below, which is called the reduced row echelon form (RREF).

(a) The rows (if any) consisting entirely of zeros are grouped together at the bottom of the matrix.

(b) In each row that does not consist entirely of zeros, the leftmost nonzero element is a 1 (called a leading 1 or a pivot).

(c) Each column that contains a leading 1 has zeros in all other entries.

(d) The leading 1 in any row is to the left of any leading 1's in the rows below it. 3. Stop process in step 2 if you obtain a row whose elements are all zeros except the last one on the right. In that case, the system is inconsistent and has no solutions. Otherwise, finish step 2 and read the solutions of the system from the final matrix. Note: When doing step 2, row operations can be performed in any order. Try to choose row operations so that as few fractions as possible are carried through the computation. This makes calculation easier when working by hand.

2.Implementation in C language:

```
#include<stdio.h>
int main()
{
   int i,j,k,n;
   float A[20][20],c,x[10];
   printf("\nEnter the size of matrix: ");
```

```
    scanf("%d",&n);
    printf("\nEnter the elements of augmented matrix row-wise:\n");
    for(i=1; i<=n; i++)
    {
        for(j=1; j<=(n+1); j++)
        {
            printf(" A[%d][%d]:", i,j);
            scanf("%f",&A[i][j]);
        }
    }
    /* Now finding the elements of diagonal matrix */
    for(j=1; j<=n; j++)
    {
        for(i=1; i<=n; i++)
        {
            if(i!=j)
            {
                c=A[i][j]/A[j][j];
                for(k=1; k<=n+1; k++)
                {
                    A[i][k]=A[i][k]-c*A[j][k];
                }
            }
        }
    }
    printf("\nThe solution is:\n");
    for(i=1; i<=n; i++)
    {
        x[i]=A[i][n+1]/A[i][i];
        printf("\n x%d=%f\n",i,x[i]);
    }
    return(0);
}
```