International Journal of Multidisciplinary Education and Research
ISSN: 2455-4588; Impact Factor: RJIF 5.12
www.multieducationjournal.com
Volume 1; Issue 3; May 2016; Page No. 72-77

Comparative analysis of gauss elimination and gauss-Jordan elimination

Megha

Assistatnt Professor, SD (PG) College, Panipat, Haryana, India

Abstract

The article examines the comparisons of execution time between Gauss Elimination and Gauss Jordan Elimination Methods for solving structure of linear equations. It tends to compute unknown variables in linear system. Various types of linear equations have been solved with the help of Gauss Jordan and Gauss Elimination methods using computer program developed in Java Programming language. Numbers of operations drawn in the solutions of linear simultaneous equations have also been calculated. This paper also overtly reveals that the two said methods could be applied to diverse systems of linear equations arising in fields of study like Business, Economics, Physics, Chemistry, etc.
This paper has a propensity to appraise the performance comparison between Gauss Elimination and Gauss Jordan sequential algorithm for solving system of linear equations. Further it has the potential to develop these methods in parallel.

Keywords: Gauss Elimination, Gauss Jordan Elimination, Linear system

Introduction

Solving systems of linear equation is perhaps one of the most perceptible applications of Linear Algebra. A system of linear equations emerges in almost every branch of Science Commerce and Engineering. Countless scientific and engineering problems can acquire the form of a system of linear equations. Many realistic problems in science, economics, engineering, biology, communication, electronics, etc. can be condensed to solve a system of linear equations. These equations may include thousands of variables, so it is vital to solve them as ably.
In linear algebra, Gaussian elimination is an algorithm used to solve systems of linear equations, finding the rank of a matrix and calculating the inverse of an invertible square matrix. Gaussian elimination is measured as the workhorse of computational science for the resolution of a system of linear equations.
Carl Friedrich Gauss, 19th century mathematician recommended this elimination method as an element of his evidence of a fussy theorem. A computational scientist uses this “proof” as a shortest computational method.
Gaussian elimination is a systematic submission of basic row operations to a system of linear equations in order to alter the system to upper triangular form. Once the coefficient matrix is in upper triangular form, reverse substitution is used to find a result. Gaussian elimination seats zeros below each pivot in the matrix initiating with the top row and working downwards. Matrices containing zeros beneath each pivot are said to be in row echelon form.
Gaussian elimination process has two parts. The first part (forward elimination) trim down a given system to either triangular or echelon form. The second part uses back substitution to find the solution of system of linear equation.
The Gauss - Jordan method is a variation of the Gaussian elimination. It is named after Carl Friedrich Gauss and Wilhelm Jordan because it is a variation of Gaussian elimination. In 1887, Jordan described while Gaussian elimination places zeros beneath each pivot in the matrix starting with the top row and working downwards, Gauss Jordan elimination method goes a step advance by placing zeroes above and below each pivot. Every matrix has a reduced row stratum form and Gauss – Jordan elimination is assured to find it.

Background theory

A system of linear equations is a collection of linear equation involving the similar set of variables.
An explanation of a linear system is an assignment of values to the variable x1, x2, x3,…, xn equivalently, such that {xi}n i = 1, each of the equation is satisfied. The set of every probable solution is called the solution set. A linear system solution may behave in any of the following ways:

1) The system has infinitely many solutions (∞)
2) The system has a unique solution (1)
3) The system has no solution (0 or Nil)

For the three variables, each linear equation concludes a plane in 3D spaces and the solution set is the intersection of these planes. Thus, the solution set may perhaps be a line, a single point or the empty set. For variables, each linear equation determines a hyper plane in n-dimensional space. The solution set is the intersection of these hyper-planes, which may be a flat of any dimension. The solution set for two equations in three variables is generally a line.
Generally, the comportment of a linear system is resolute by the relationship between the number of equations and the number of unknown variables. Usually, a system with fewer equations than unknowns has infinitely many solutions such system is also known as an undetermined system. A system with the same number of equation and unknowns has a single unique solution. A system with more equations than unknowns has no solution. There are many methods of solving linear systems. Among them, Gauss and Gauss Jordan elimination methods shall be considered [2]. So, they are discussed in details in the following sections.

A. Gauss elimination method

Gaussian elimination is a systematic application of elementary row operations to a system of linear equations in order to convert the system to upper triangular form. Once the coefficient matrix is in upper triangular form, Substitution is used to find a solution. The general procedure for Gaussian elimination can be summarized in the following steps:

1. Write the augmented matrix for the system of linear equation.
2. Use elementary operation on {A/b} to transform A into upper triangular form. If a zero is located on the diagonal, switch the rows until a non zero is in that place. If you are unable to do so, stop; the system has either in finite or no solution.
3. Use back substitution to find the solution of the problem.

Consider the n linear equations in n unknowns

a11x1 + a12x2 + … + a1nxn = a1,n+1
a21x1 + a22x2 + … + a2nxn = a2,n+1
a31x1 + a32x2 + … + a3nxn = a3,n+1
… … … … … …
an1x1 + an3x2 + … + annxn = an,n+1

Where ai,j and ai,j+1 are known constant and xi's are unknowns. The system is equivalent to AX = B

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \cdots \\ x_n \end{bmatrix} = \begin{bmatrix} a_{1,n+1} \\ a_{2,n+1} \\ a_{3,n+1} \\ \cdots \\ a_{n,n+1} \end{bmatrix}$$

Step 1: Store the coefficients in an augmented matrix. The superscript on aij means that this is the first time that a number is stored in location (i, j).

$$\left[\begin{array}{ccccc|c} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} & a_{1,n+1} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} & a_{2,n+1} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} & a_{3,n+1} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} & a_{n,n+1} \end{array}\right]$$

Step 2: If necessary, switch rows so that a11 ≠ 0, then eliminate x1 in row 2 through n. In this process mi1 is the multiple of row1 that is subtracted from row i.

```
for i= 2 to n
    mi1 = ai1 / a11
    ai1 = 0
    for j = 2 to n+1
        aij = aij – mi1 * a1j
    end for
end for
```

The new elements are written aij to indicate that this is the second time that a number has been stored in the matrix at location (i, j). The result after step 2 is

$$\left[\begin{array}{ccccc|c} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} & a_{1,n+1} \\ 0 & a_{22} & a_{23} & \cdots & a_{2n} & a_{2,n+1} \\ 0 & a_{32} & a_{33} & \cdots & a_{3n} & a_{3,n+1} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & a_{n2} & a_{n3} & \cdots & a_{nn} & a_{n,n+1} \end{array}\right]$$

Step 3: If necessary, switch the second row with some row below it so that a22 ≠ 0, then eliminate x2 in row 3 through n. In this process ui2 is the multiple of row 2 that is subtracted from row i.

```
for i = 3 to n
        ui2 = ai2 / a22
        ai2 = 0
        for j = 3 to n+1
        aij = aij – ui2 * a2j
      end for
end for
```

The new elements are written aij indicate that this is the third time that a number has been stored in the matrix at location (i, j). The after step 3 is

$$\left[\begin{array}{ccccc|c} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} & a_{1,n+1} \\ 0 & a_{22} & a_{23} & \cdots & a_{2n} & a_{2,n+1} \\ 0 & 0 & a_{33} & \cdots & a_{3n} & a_{3,n+1} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & a_{n3} & \cdots & a_{nn} & a_{n,n+1} \end{array}\right]$$

Step k+1: This is the general step. If necessary, switch row k with some row beneath it so that akk ! = 0; then eliminate xk in rows k+1 through n. Here uik is the multiple of row k that is subtracted from row i.

```
for i = k +1 to n
        uik = aik/akk
        aik = 0
        for j = k + 1 to n+1
              aij = aij – uik * akj
        end for
end for
```

The final result after xn-1 has been eliminated from row n is

$$\left[\begin{array}{ccccc|c} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} & a_{1,n+1} \\ 0 & a_{22} & a_{23} & \cdots & a_{2n} & a_{2,n+1} \\ 0 & 0 & a_{33} & \cdots & a_{3n} & a_{3,n+1} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & a_{nn} & a_{n,n+1} \end{array}\right]$$

The upper triangularization process is now complete xn = an, n+1/an

```
for i = n to 1 step -1
        sum = 0
        for j = i+1 to n
              sum = sum + aij * xj
        end for
  xi  = (ai, n+1 - sum)/aii
end for
```

Perform the back substitution, get the values of xn, xn-1, xn-2,…, x1.

Sequential Algorithm- Gauss Elimination Method

Input: Given Matrix a[1: n, 1: n+1]

Output: x[1: n]

1. for k = 1 to n-1

2. for i = k+1 to n
3. u = aik/akk
4. for j = k to n+1
5. aij = aij – u * akj
6. next j
7. next i
8. next k
9. xn= an,n+1/ann
10. for i = n to 1 step -1
11. sum = 0
12. for j = i+1 to n
13. sum = sum + aij * xj
14. ne xt j
15. xi = (ai, n+1- sum)/aii
16. next i
17. end

B. Gauss Jordan elimination method

Gauss Jordan elimination is a modification of Gaussian elimination. Again we are transforming the coefficient matrix into another matrix that is much easier to solve and the system represented by the new augmented matrix has the same solution set as the original system of linear equations. In Gauss Jordan elimination, the goal is transform the coefficient matrix into a diagonal matrix and the zeros are introduced into the matrix one column at a time. We work to eliminate the elements both above and below the diagonal elements of a given column in one passes through the matrix. The general procedure for Gauss Jordan elimination can be summarized in the following steps:

1. Write the augmented matrix for a system of linear equation
2. Use elementary row operation on the augmented matrix [A/b] to the transform A into diagonal form (pivoting). If a zero is located on the diagonal, switch the rows until a non-zero is in that place. If you are unable to do so, stop; the system has either infinite or no solution.
3. By dividing the diagonal elements and a right-hand side's element in each row by the diagonal elements in the row, make each diagonal elements equal to one.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix}$$

Step 1: Write the above as augmented matrix, we have

$$\left[\begin{array}{cccc|c} a_{11} & a_{12} & a_{13} & a_{14} & b_1 \\ a_{21} & a_{22} & a_{23} & a_{24} & b_2 \\ a_{31} & a_{32} & a_{33} & a_{34} & b_3 \\ a_{41} & a_{42} & a_{43} & a_{44} & b_4 \end{array}\right]$$

Step 2: Pivoting the matrix, that is,

- Interchanging rows, if necessary, to obtain an augmented matrix in which the first entry in the first row is nonzero (e.g. Ri ↔ Rj).
- Add one row to another row, or multiply one row first and then adding it to another (e.g. kRj + Ri → Ri).
- Multiplying a row by any constant greater than zero (e.g. kRi → Ri).

NOTE: → means "replaces", and ↔ means "interchange")
Continue until the final matrix is in row- reduced form, i.e.

$$\left[\begin{array}{cccc|c} a_{11} & 0 & 0 & 0 & b_1 \\ 0 & a_{22} & 0 & 0 & b_2 \\ 0 & 0 & a_{33} & 0 & b_3 \\ 0 & 0 & 0 & a_{44} & b_4 \end{array}\right]$$

Sequential Algorithm – Gauss Jordan Elimination Method

Input: Given Matrix a[1: n, 1: n+1]

Output: x[1: n]

1. for i = 1 to n
2. for j = 1 to n+1
3. if (i ≠ j) then
4. u = aik/akk
5. for k = 1 to n+1
6. ajk = ajk – u * aik
7. end if
8. next k
9. next j
10. next i
11. for i = 1 to n
12. xi = ai,n+1/aii
13. end

Design and implementation of the proposed system

A. Design of the proposed system

The design of the proposed system is illustrated in figure 1. In this system, the algorithm has first
selected to solve the problem. Then send the input problem or the system of linear equations for solving by each algorithm. Here the problem is calculated both of Gauss Jordan and Gauss Elimination algorithm. It is illustrious that the two algorithms give the same results for the same problem. But results show the different execution time for the solution. So, the system will give the disparity of execution time for the problem solving these algorithms.

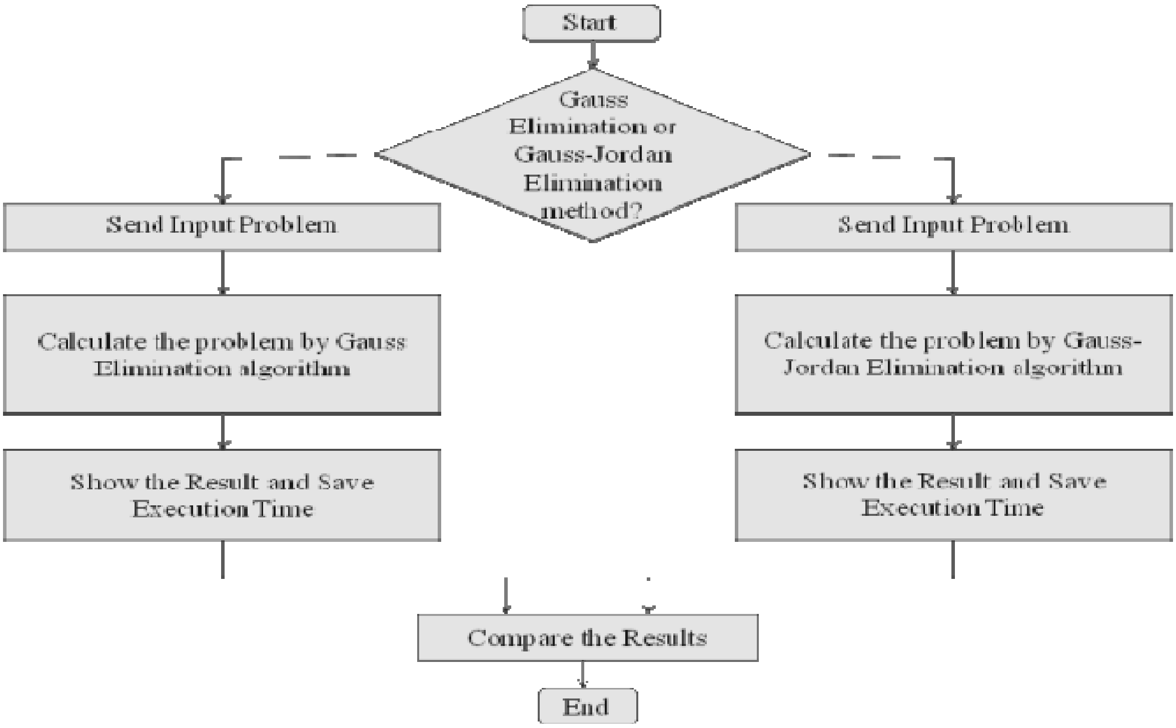

Fig 1: The Proposed System Design

B. Implementation

This system tends to compute unknown variables in linear equations with the help of Gauss Elimination and Gauss Jordan Elimination algorithms. It is implemented by using Java programming language.

When the system is started, the user will see the home window form as illustrated in figure 2. In this form, the user can choose the desire algorithm and number of unknown variables. On the other way, the user can type the desire unknown equations in the text boxes in this form.

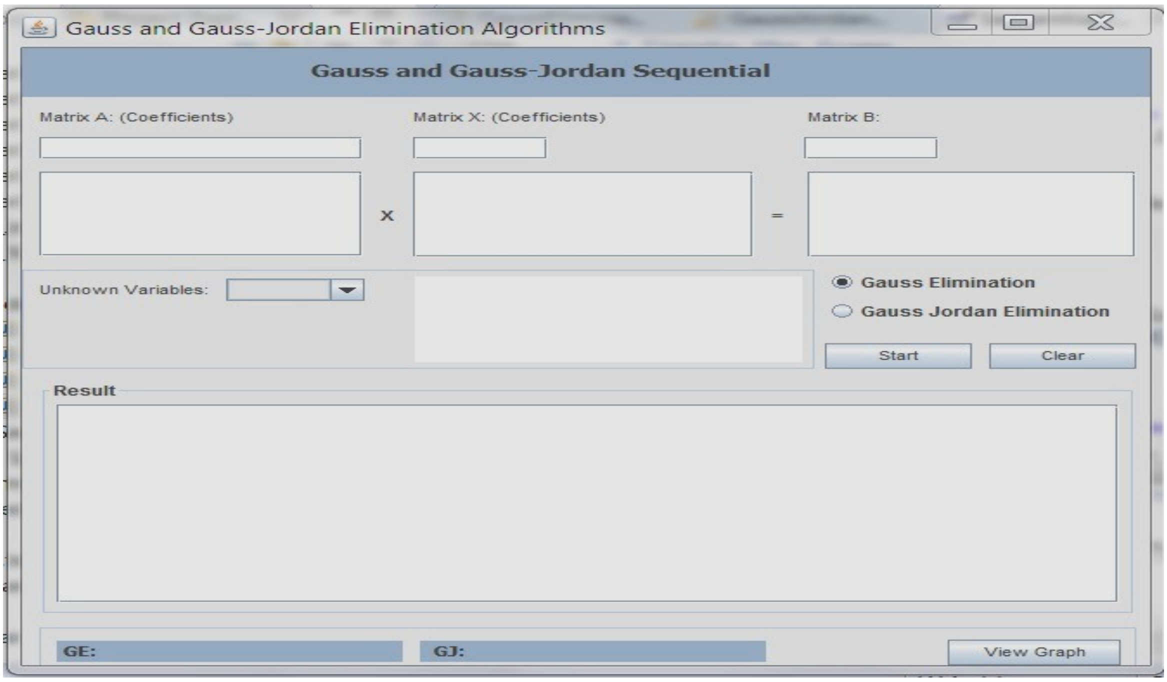

Fig 2: Implementation of the Gauss Elimination and Gauss-Jordan Elimination Methods

Under the Gauss Elimination calculation, the user must choose Gauss Elimination option and define the number of unknown variables. Figure 3 mention the sample of calculation result for five variables. And it can also display the calculation time.

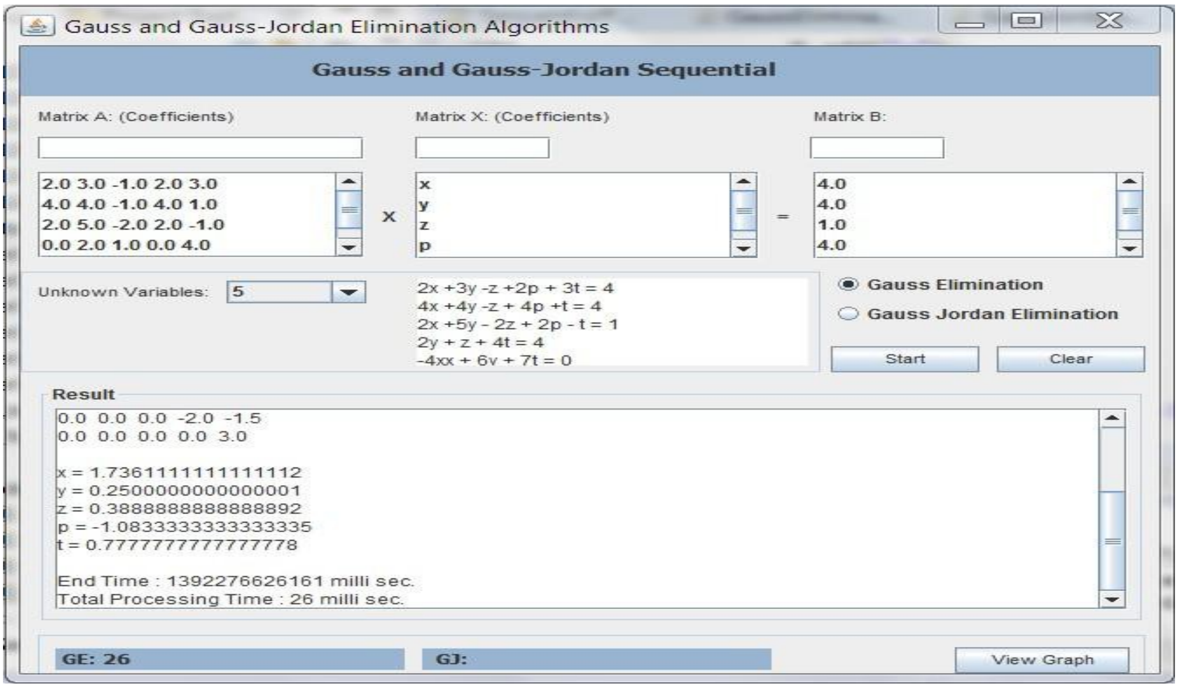

Fig 3: The result for five unknown variables using Gauss Elimination Method

Under the Gauss Jordan Elimination calculation, the user must choose Gauss Jordan Elimination option and define the number of unknown variables. Figure 4 mention the sample of calculation result for five variables. And it can also display the calculation time.

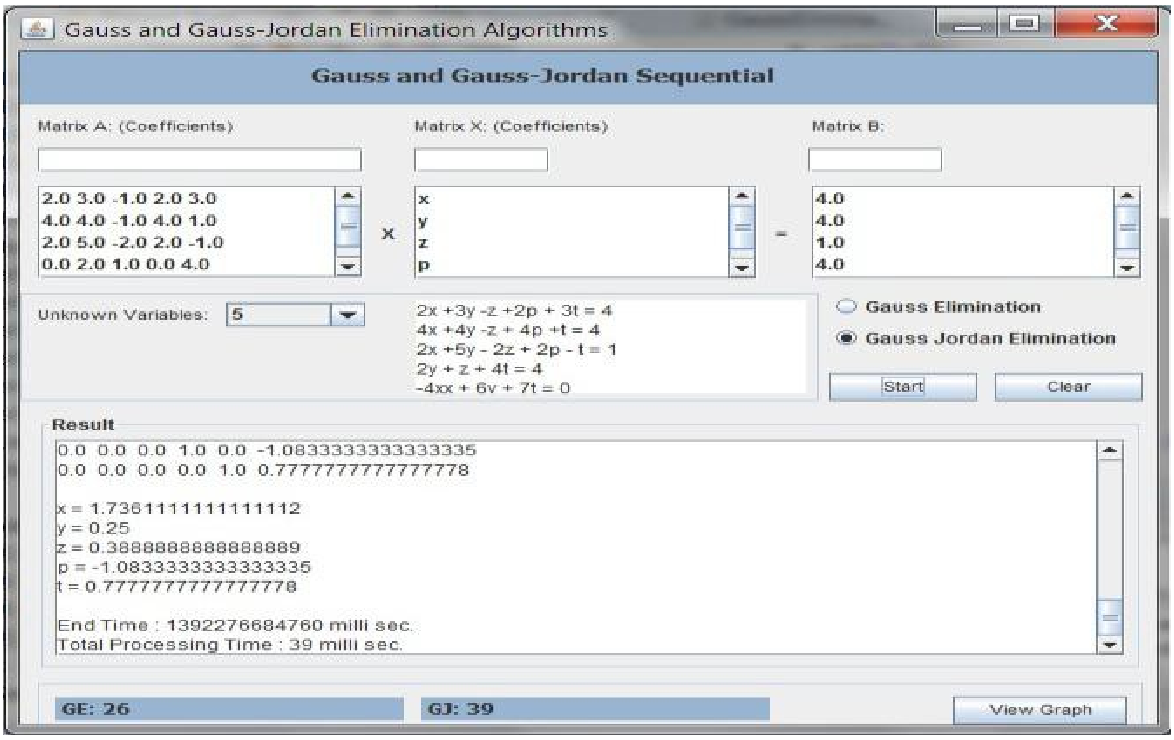

Fig 4: The results for five variables using Gauss Elimination and Gauss -Jordan Elimination Methods

Finally, the system shows the comparison of execution time or calculation time of these two algorithms.

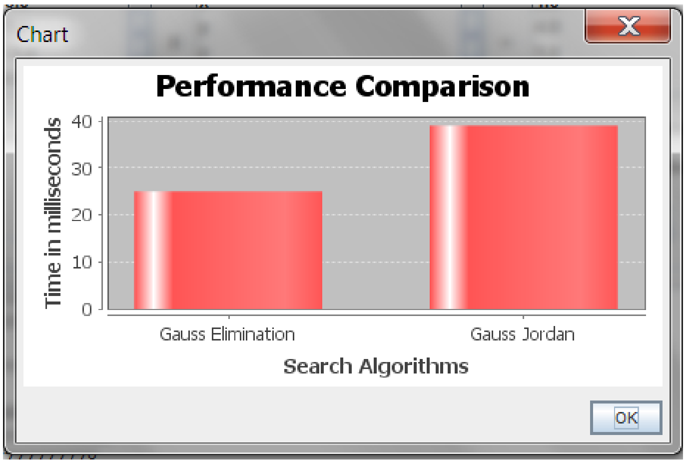

Fig 5: The chart for the comparison of two algorithms

The comparison of execution time between the Gauss Elimination and Gauss Jordan Elimination is mentioned in Table 1 and figure 5.These results are carried out on the calculation of unknown variables 2, 3, 4, 5, 6 and 7.

Table I: The Comparison of Execution Time between Gauss Elimination and Gauss-Jordan Elimination Method.

Numbers of Variables	Execution Time for Gauss Elimination method (Milliseconds)	Execution Time for Gauss Jordan Elimination method (Milliseconds)
2	14	25
3	16	31
4	20	36
5	26	39
6	29	56
7	46	76

According to these results, the Gauss Elimination method is faster than Gauss Jordan Elimination method.

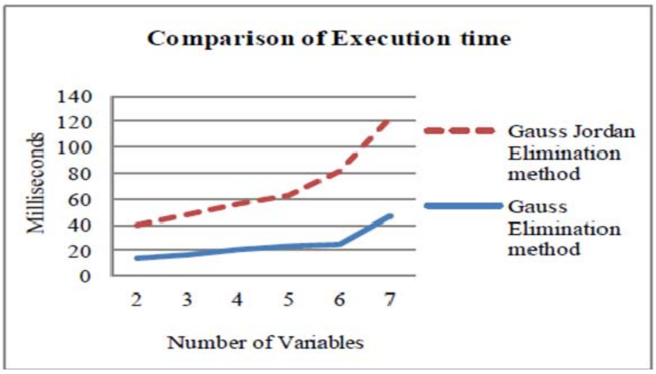

Fig 6: The Comparison of Execution times

Conclusions

Many scientific and engineering problems can take the form of a system of linear equations.

These equations may contain thousands of variables, so it is significant to solve them as efficiently as possible. In this article, the unknown variables in linear system are carried out by using Gauss Elimination Method and Gauss -Jordan Elimination Method. And the experiment is confirmed with the help of java programming language. The results are carried out by solving 2, 3, 4, 5, 6 and 7 unknown variables. According to these experimental results, Gauss Elimination Method is faster than the Gauss Jordan Elimination method. Therefore Gauss Elimination Method is more efficient than the Gauss Jordan Elimination method.

Gaussian Elimination helps to put a matrix in row echelon form, while Gauss-Jordan Elimination puts a matrix in reduced row echelon form. For miniature systems, it is usually more convenient to use Gauss-Jordan elimination and explicitly solve for each variable represented in the matrix system. However, Gaussian elimination in itself is occasionally computationally more efficient for computers. Also, Gaussian elimination is all you need to determine the rank of a matrix (an important property of each matrix) while going through the trouble to put a matrix in reduced row echelon form is not worth it to only solve for the matrix's rank.

References

1. Luke Smith, Joan Powell. An Alternative Method to Gauss-Jordan Elimination: Minimizing Fraction Arithmetic, the Mathematics Educator, 2011.
2. K. Rajalakshmi. Parallel Algorithm for Solving Large System of Simultaneous Linear Equations, IJCSNS International Journal of Computer Science and Network Security. 2009; 9:7.
3. Adenegan, Kehinde Emmanuel1, Aluko. Tope Moses: Gauss and Gauss-Jordan elimination methods for solving system of linear equations: comparisons and applications, Journal of Science and Science Education, Ondo. 19th November, 2012.
4. R.B Srivastava, Vinod Kumar. Comparison of Numerical Efficiencies of Gaussian Elimination and Gauss-Jordan Elimination methods for the Solutions of linear Simultaneous Equations, Department of Mathematics M.L.K.P.G.
5. College Balrampur. U.P., India, November 2012.
6. T.J. Dekker, W. Hoffmann. Rehabilitation of the Gauss- Jordan algorithm, Department of Computer Systems, University of Amsterdam, Kruislann 409, 1098 SJ Amsterdam, The Netherlands, 1989.