Analysis of Gaussian Elimination Method for Solving Systems of Linear Equations

Ogbonnaya C Ngwu, Taylor McRae, Ran Bi
Florida Institute of Technology,
Melbourne, FL. USA.
ongwu2014@my.fit.edu, tmcrae2012@my.fit.edu, rbi2013@my.fit.edu

***Abstract* - In linear algebra, the concept of Gaussian Elimination is really important in finding the solutions of systems of linear equations. Variations of this algorithm exists and many pivoting methods can be applied in order to reduce error in the algorithm. This study focus on the solution of a linear system with n equations and m unknown variables using Gaussian Elimination. The general idea of Gaussian Elimination is discussed including methods of reducing errors and its variations. The algorithm for Gaussian Elimination is then examined to elaborate the ideas behind the algorithm and how parts of it (forward elimination and back substitution) work to achieve efficiency. JProfiler, a JAVA profiling tool, is used to collect runtime data to analyze the algorithm. The experimental model is compared to the theoretical model using a regression model and the coefficient of determination, r2 = 0.99355, is used to evaluate the amount of variance in the experimental model accounted for by the theoretical model. The study concludes the experimental model has the same time complexity as the theoretical model with an error margin of about 0.64%.**

***Keywords* - Gaussian elimination, Linear algebra, Time complexity, Profiling, Partial pivoting, Forward elimination, Back substitution**

I. INTRODUCTION

Gaussian elimination is a linear algebra algorithm used in solving systems of linear equations. The algorithm has a long history tracing back to 179 CE, but its major introduction in the field of linear algebra was in 1810 when Carl Friedrich Gauss developed the symmetric elimination for solving normal equations of minimum square problems.

In the field of Linear Algebra, solutions involving the use of this algorithm depends on the formation of an augmented matrix and the application of the elementary row operations. A matrix from by appending two matrices together is said to be augmented. Generally, system of linear equations can be converted into three vectors. The first vector (matrix) is the coefficient matrix. The coefficient matrix, usually denoted by A, is an n x m matrix that contains the coefficients of the unknown variables. In this matrix, the coefficients of a single equation in the system takes up an entire row and the coefficients for individual variables in each equation is entered in the columns of the equation's row. The variable vector is an m x 1 matrix represented by x and contains all the variables in the system in their correct order. The solution vector is an n x 1 matrix represented by b contains all the solutions in the right hand side of the system. The dot product of the variable and coefficient matrix is equal to the solution vector and is represented thus: Ax = b. This is referred to as the general form of a linear system of equation. The augmented matrix is then formed by appending the solution vector to the coefficient matrix.

Theoretically, all square (n x n) matrices can be converted to an identity matrix by apply the three elementary row operations. The first and perhaps the simplest of the row operations is the row interchange. The row interchange simply involved a swap of two rows adjacent to each other. The next row operation is the scalar multiplication. In this operation, the entire row in the matrix is multiplied by a scalar. A scalar is a field element with only magnitude and no direction. The last and most complex of

the operations is the scalar multiplication and addition operation. Here, each value in a given row (R1) is multiplied by a scalar and then added to a corresponding value (in the same column as the multiplied value) in the target row (R2). These operations form the basis for Gaussian Elimination.

The goal of Gaussian Elimination is to reduce the augmented matrix to its row echelon form. A matrix is said to be in row echelon form if gaussian elimination has operated in the rows of the matrix using the elementary row operations. Specifically, a matrix is said to be in row echelon form if the all the nonzero rows in the matrix is placed above any rows or group of rows with all zeros, and the pivot (leading coefficient; the first nonzero element in the row from the left) is always strictly on the right of the pivot of the row above it. This process is usually referred to forward elimination process. Once the matrix is in its row echelon form, the final step for finding the solution of the linear system is the application of the back substitution process. In back substitution, unlike forward elimination, starts from the bottom of the matrix, and uses the answers to solutions of the variables found from previous substitutions to find the solution of the next variables until all values of the unknown variables are found. A simple variation of the Gaussian Elimination is the Gauss-Jordan Elimination proposed by Wilhelm Jordan in 1888. The Gauss-Jordan Elimination method, unlike Gaussian Elimination, converts the augmented matrix into its reduced row echelon form. A matrix is said to be in reduced row echelon form if it is in reduced row echelon form and the pivot (leading coefficient) is 1 and is the only nonzero element in its column. Gauss-Jordan Elimination is costly in the elimination phase since it uses forward elimination to convert the augmented matrix to its row echelon form and then uses backward elimination to reach the reduced row echelon form. However, it is computationally cheap in substitution since no back or forward substitution is required.

Regular Gaussian Elimination operates on the assumption that the system is stable and the pivot values are relatively large enough and will not result in an error. However, this is not always true. When the pivot is small, the operations involving the pivot usually result in minute errors. In numerical analysis, this is referred to as stability problem. A stability problem occurs when a small error in input results in a large error in the output. In Gaussian Elimination and its variation, Gauss-Jordan Elimination, small pivot values (A[i][i]) results in a stability problem. The stability problem can be eliminated with the use of a pivoting procedure. One of the pivoting methods is complete pivoting. Complete pivoting, also known as maximal pivoting, conducts a row and column interchange in order to use the largest absolute value element in the augmented matrix as the pivot. This procedure is rarely used since the cost of the interchange most usually outweighs its efficiency in achieving numerical stability. Most often used is the partial pivoting method. In partial pivoting, the pivot is chosen to be the largest absolute value in the column that is being considered to have a pivot. This is done by searching the entire column below the current value used as initial pivot to find the largest absolute value and then performing a row swap. A variation of the partial pivoting method is the scaled partial pivoting in which the value selected is the largest (not absolute) value in the column. Generally, partial pivoting method is valuable in reducing round-off errors in the elimination process.

II. ALGORITHM OVERVIEW

The algorithm for solving systems of linear equations can be divided into two subprotocols for forward elimination and backward substitution. The computer code for the forward elimination is shown below:

```
public static void doForwardElimination() {
    for (int j = 0; j < A.length - 1; j++) {
        for (int i = j+1; i < A.length; i++) {
            float factor = 0 - (A[i][j]/A[j][j]);
            for (int l = 0; l < A.length; l++) {
                A[i][l] += factor * A[j][l];
            }
            b[i] = b[i] + factor * b[j];
        }
    }
}
```

Fig. 1 Forward elimination subprotocol for Gaussian Elimination

The program uses only the third row operation to reduce the coefficient array to the row echelon form. The concept of augmented matrix is not used for simplicity. Operations are applied to the coefficient array and solution array independently. The program traverses through all columns in the array except the last one. This makes sense since there are no elements below the pivot element of the array to reduce. For each column, a pivot element is selected. The pivot element is then used to reduce the elements below it to zero by subtracting a scalar multiple of the pivot from the elements in the column. The factor is also applied to the solution vector as would in a true augmented matrix.

The second part of the solution is the back substitution process. The computer code for the back substitution is shown below:

```
public static void doBackwardSubstitution() {
    for (int j = A.length - 1; j > 0; j--) {
        for (int i = j+1; i < A.length; i++) {
            b[j] -= A[j][i] * x[i];
        }
        x[j] = b[j] * A[j][j];
    }
}
```

Fig. 2 Back substitution subprotocol for Gaussian Elimination

In the back substitution, the process is started from the bottom right of the coefficient array. On each iteration, the respective solution array element is reduced by subtracting the values of the product of the known variables and their coefficients. The solution for the iteration is then the product of the reduced solution array element and the pivot element of the current iterated position.

The cost of the backward substitution can be removed by using the Gauss-Jordan Elimination method. However, this makes the elimination cost approximately twice that of Gaussian Elimination. The Gaussian Elimination is thus analyzed for performance.

III. Time Complexity: Sample, Measure, Result And Error Analysis

A profiler is used to obtain runtime data required to measure the performance of the algorithm. For this, JProfiler, a commercially licensed profiling tool for JAVA programs and applications, is used. With JProfiler, it is possible to isolate a single method and test its performance without interference from other parts of the code or even the garbage collector. The sample input ranged from 2 to 15 for both i and j where i is the row size and j is the column size of the coefficient matrix. The size of j is usually equal or greater than i by 1 in each test case. This criterion was chosen in order to access performance of the algorithms at upper endpoints rather than midpoints or lower endpoints. The data collected was recorded in Microsoft Excel for statistical analysis. To unify the input size, i and j is added to form the independent variable (i+j). Time was measured in microseconds (μs).

Theoretically, the performance of the algorithm used is given by O(n^3). The algorithm's time complexity is cubic. To compare the test model to the theoretical model, a graph of the theoretical model and regression line is superimposed on the graph as shown below:

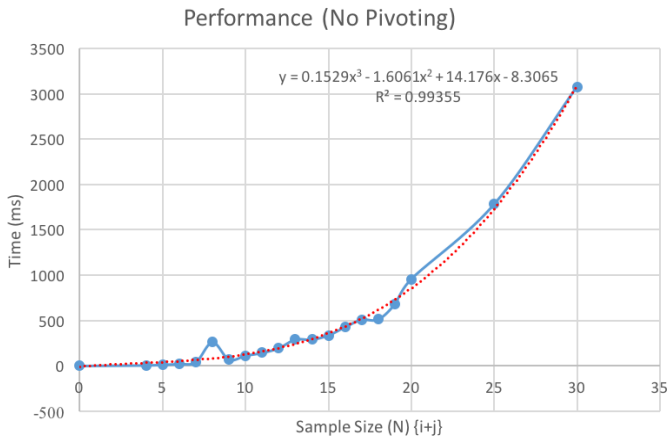

Fig. 3 Time performance graph of Gaussian Elimination

From direct observation of the graph above, the line of best fit is cubic indicating support for the experimental model being in line with the theoretical model, the coefficient of determination, r2 = 0.99355, implying that the graph of y = 0.1529x^3 + 14.176x - 8.3065 (regression model) accounts for 99.36% of the variance of the experimental model. In other words, the experimental model matched the theoretical model with a non-significant error of approximately 0.64%. A numerical error analysis can be performed to support this data.

IV. Applications

Besides the use in solving system of linear equations, the other important applications of Gaussian Elimination in real life are computing determinants of a square matrix, and finding the inverse of a matrix.

To compute the determinants of a matrix using Gaussian Elimination, the procedures are to add the scalar multiples of one row to another row first, until all the entries below the main diagonal are equals to zero, and because of the action of additions above will not change the determinant, then the product of diagonal entries is the determinant.

The variant of Gaussian Elimination, Gauss-Jordan elimination, can be used to find the inverse of a specific invertible matrix. The procedures are to form a n*2n block matrix firstly, the n*2n matrix is an augmented matrix consisting of the original n*n matrix on the left side and a n*n identity matrix on the

right side and then use the application of elementary row operations to find the echelon form that reduced from this n*2n matrix. Since the original matrix is invertible, therefore the left block of the n*2n matrix should been reduced to the identity matrix, in this case, the right block of the final matrix is the inverse of the original matrix.

V. Conclusion

Generally, Gaussian Elimination method is an efficient way of solving systems of linear equations. It is one of the oldest algorithms in history and still in use today. The Gauss-Jordan Elimination method removes the cost of back substitution with a greater cost for elimination. It does not necessarily perform better than traditional Gaussian Elimination but excels in cases where back substitution is extremely computational costly. For simple systems of linear equations, Gaussian Elimination with back substitution has a lower overhead cost. Overall, Gaussian Elimination is used to reduce a matrix to its echelon form and help simplify complex equations using a pivoting point.

References:

[1] R. L. Burden, J. D. Faires, and A. M Burden, *Numerical Analysis*, 10th Ed. Boston, Massachusetts: Cengage Learning, 2016.

[2] S. J. Leon, *Linear Algebra with Applications*, 9th. Ed. New York: Pearson Education, 2015.