Conceptual and Practical Review of Gaussian Elimination and Gauss-Jordan Reduction

I Putu Alit Sudrastawa[1], Agus Parwata[2], Made Wisnu Adhi Saputra[3], I Gusti Agung Made Wirautama[4], I Wayan Septa Malan Vergantana[5]

[1,2,3,4,5,6] Program Studi Sistem Informasi
Universitas Pendidikan Ganesha
e-mail: igedearisgunadi@undiksha.ac.id

Abstract

In this paper, the implementation of Gaussian elimination and Gauss-Jordan reduction are discussed in detailed explanations. There were several results from various research journals compared to comprehend the concepts and practices of both methods. The goal was to analyze the differences of Gaussian elimination and Gauss-Jordan reduction regarding the application, algorithm, complexity, efficiency, problem-solving ability, and stability. As the result, we found that the Gauss-Jordan reduction has lower time efficiency and greater residual (the error in the result) in problem-solving ability. The rests are somewhat similar. In practice, those differences are a substantial consideration in choosing the most suitable method for a particular linear equations' problem, especially when accuracy and computation time have a significant impact on the result.

Keywords: Gaussian Elimination, Gauss-Jordan Reduction, Algorithm, Big-O Notation

I. Introduction

Computing technology has brought a significant impact in resolving a particular computational problem, e.g., in solving a system of linear equations, a collection of equations with the same set of variables which can be drawn as straight lines in the Cartesian graph. In that case, the stability and efficiency of a method is a primary concern, especially in a large computation for there will be more time needed to solve it than to solve the smaller ones [1]. Therefore, deciding which method to use is an important thing.

One of the most widely used methods for solving a system of linear equations is Gaussian elimination [2]. It is usually taught in high school linear algebra subject. Further, there is a variation of this method (i.e., Gauss-Jordan reduction [3]) which brings a slight improvement in terms of the result.

In some research journals (for example in "Open-Multi Processing of Gauss-Jordan Method for Solving System of Linear Equations" [4], "Modification of Gaussian Elimination for the Complex System of Seismic Observations" [5], etc.) which will be presented in the next section, several concepts and practices are discussed by the authors regarding the Gaussian elimination and Gauss-Jordan reduction, including the development of each method for better usability. In this paper, we summarize those concepts and practices to understand the differences in terms of the application, algorithm, complexity, efficiency, problem-solving ability, and stability of Gaussian elimination and Gauss-Jordan reduction.

II. Discussion

Gaussian elimination (or row reduction) is usually associated with producing the echelon form of a matrix [6]. Anany Levitin [7] explains that the idea of this method is to transform a system of *m* linear equations with *n* unknown variables into an equivalent system which has the same solution as the original equations. The equivalent system is in the form of an upper-triangular matrix (a matrix that all of the entries below the main diagonal are zeros).

Historically, many mathematicians have contributed to the Gaussian elimination method. One of the most well-known contributors to this method is Carl Friedrich Gauss (in the early 19^{th} century). In 1953, an influential American mathematician, George Forsythe [8], misattributed "high school elimination" as "Gaussian elimination" and eventually made Carl F. Gauss appeared as the originator of the method although there are many contributors before him, including physicist Sir Isaac Newton in 1670 [9]. Then in 1888, Wilhelm Jordan [10] improved the Gaussian elimination in order to achieve simpler and more straightforward result in solving a system of linear equations, but with more complex arithmetic computations as the trade-off. It is then known as the Gauss-Jordan reduction method.

The term of Gauss-Jordan reduction refers to a procedure which ends in reduced echelon form (same as the form of an identity matrix if there is a unique solution to the equations), while in Gaussian elimination, the result is in echelon form. In reduced echelon form, every leading coefficient is '1' and is the only non-zero entry in its column [3], while in echelon form this isn't the case.

II.1. Application

The first historically known application of the Gaussian elimination is for solving a system of linear equations. In the further development, this method can also be used to find the rank and base of a matrix, solve the inverse of an invertible square matrix, and calculate the determinant of a square matrix. As for Gauss-Jordan reduction, it has similar functions as Gaussian elimination, except for calculating the determinant which is not possible in Gauss-Jordan reduction. Both methods rely on the elementary row operations to reduce a matrix into an echelon form (in Gaussian elimination) or a reduced echelon form (in Gauss-Jordan reduction). There are three types of elementary row operations that can be performed on a matrix, i.e.:

1) Swap rows positions (this is an essential operation when one or more diagonal entries are equal to zero).
2) Add a scalar multiple of one row to another row.
3) Multiply a row with a non-zero scalar value.

Both the Gaussian elimination and Gauss-Jordan reduction may be summarized as follows: eliminate x_1 from E2, eliminate x_1 and x_2 from E3, eliminate x_1, x_2, and x_3 from E4, and so on. In the result, Gaussian elimination will put the system of linear equations into a triangular matrix form and each unknown variable can be solved using the back-substitution technique, while Gauss-Jordan reduction will put the system into an identity matrix form (if there is a unique solution), consequently it doesn't need back-substitution. Table 1 shows the application example of both Gaussian elimination and Gauss-Jordan reduction using elementary row operations in solving a system of linear equation.

Table 1. Example of Gaussian Elimination and Gauss-Jordan Reduction

Equations	Augmented Matrix
$x_1 + 5x_2 = 7$ $-2x_1 - 7x_2 = -5$	$\begin{bmatrix} 1 & 5 & 7 \\ -2 & -7 & -5 \end{bmatrix}$
Step 1: add twice Row 1 to Row 2 ($R_2 = 2 * R_1$)	
$x_1 + 5x_2 = 7$ $3x_2 = 9$	$\begin{bmatrix} 1 & 5 & 7 \\ 0 & 3 & 9 \end{bmatrix}$
Step 2: multiply Row 2 by 1/3 ($R_2 = R_2 * 1//3$)	
$x_1 + 5x_2 = 7$ $x_2 = 3$	$\begin{bmatrix} 1 & 5 & 7 \\ 0 & 1 & 3 \end{bmatrix}$
Step 3: add -5 times Row 2 to Row 1 ($R_1 = -5 * R_2 + R_1$)	
$x_1 = -8$ $x_2 = 3$	$\begin{bmatrix} 1 & 0 & -8 \\ 0 & 1 & 3 \end{bmatrix}$

The procedure is called Gaussian elimination when it ends in Step 1 (echelon form), while it is called Gauss-Jordan reduction when it ends in Step 3 (reduced echelon form). When using the Gaussian elimination (Step 1), the value of x_2 needs to be calculated, which is equal to 9/3. Then, the first equation can be solved using back-substitution, which is $x_1 + 5(9/3) = 7$, as the result $x_1 = -8$. On the other hand, when the procedure is carried out until step 3, back-substitution will not be needed because the solutions have been found, which are $x_1 = -8$ and $x_2 = 3$.

In the past few years, some researchers have applied and modified both Gaussian elimination and Gauss-Jordan reduction for several uses, e.g., Luke Smith [11], Jun Ji [12], Dasgupta [13], and Tatyana Smaglichenko [5]. Luke Smith improved the Gauss-Jordan reduction by minimizing produced fractions to make the calculations easier. Jun Ji implemented the Gauss-Jordan reduction as an alternative method to Moore-Penrose matrix inversion. Dasgupta modified the Gauss-Jordan reduction to be more resource efficient in matrix inversion. Tatyana Smaglichenko implemented the Gaussian elimination for practical use in seismic observations.

II.2. Algorithm

In practice, one doesn't usually deal with equations but instead more of translating them into an augmented matrix, which is more suitable for computing algorithm. In the book of "Introduction to the Design and Analysis of Algorithms 3rd Edition," Anany Levitin [7] gave an example of Gaussian elimination algorithm with partial pivoting by using augmented matrix A [1 to *n*, 1 to *n*] and column-vector B [1 to *n*] as the inputs. The output is an equivalent system (upper-triangular matrix) of A with the corresponding right-hand side values in the (n + 1)st column. The algorithm is as follows:

```
//appends B to A as the last column
for i ← 1 to n do A[i, n + 1] ← B[i]
for i ← 1 to n - 1 do
    //partial pivoting
    p_row ← i
    for j ← i + 1 to n do
        //for partial pivoting
        if |A[j, i]| > |A[p_row, i]|
        p_row ← j
    for k ← i to n + 1 do
        swap (A[i, k], A[p_row, k])
    for j ← i + 1 to n do
        temp ← A[j,i] / A[i, i]
        for k ← i to n + 1 do
            //The actual row operations
            A[j,k]← A[j,k] - A[i, k] ∗
  temp
```

Before eliminating a variable, the algorithm first exchanges rows to move the entry with the selected p_row value to the pivot position using partial pivoting. In this pivoting technique, the algorithm selects the entry with the largest absolute value from the matrix's column or row as a pivot, which in the example using the matrix's row. It is sufficient to reduce round-off error. However, for a particular system, another pivoting technique (i.e., complete pivoting) may be required for better accuracy and stability. It does the similar thing as partial pivoting except it interchanges both rows and columns. Complete pivoting is usually unnecessary, considering the additional cost of searching for the maximal pivot element. Therefore, it is rarely used [14].

Upon completion of the algorithm, the augmented matrix will be in echelon form. Then, the equations can be solved using back-substitution. On the other hand, the Gauss-Jordan reduction has a slightly different algorithm (see the example in subsection 2.3).

II.3. Complexity and Efficiency

The complexity of the Gaussian elimination can be calculated using summation which is determined by how many nested loops exist [7]. It begins from the innermost loop (see the algorithm example in subsection 2.2). Note that there are three nested loops in the example (see the gray marked lines). Here is the summation calculation of the Gaussian elimination complexity:

$$C(n) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \sum_{k=1}^{n+1} 1 = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} (n+1-i+1) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} (n+2-i)$$

$$= \sum_{i=1}^{n-1} (n+2-i)(n-(i+1)+1) = \sum_{i=1}^{n-1} (n+2-i)(n-i)$$

$$= (n+1)(n-1) + n(n-2) + \cdots + 3 \cdot 1 = \sum_{j=1}^{n-1} (j+2)j = \sum_{j=1}^{n-1} j^2 + \sum_{j=1}^{n-1} 2j$$

$$= \frac{(n-1)n(2n-1)}{6} + 2\frac{(n-1)n}{2} = \frac{n(n-1)(2n+5)}{6} \approx \frac{1}{3}n^3 \in O(n^3)$$

Based on the summation, the arithmetic complexity of the Gaussian elimination is equal to $O(n^3)$ by using big-O notation. This complexity also applies when solving the inverse of an invertible square matrix or calculating the determinant of a square matrix. The complexity simulation of this method is given in Figure 1.

Theoretically, Gauss-Jordan reduction shares the same complexity of $O(n^3)$ as the Gaussian elimination but is a bit slower because of the additional steps to form the reduced echelon form [1]. In many cases, achieving reduced echelon form is unnecessary. Therefore, Gaussian elimination is often preferred.

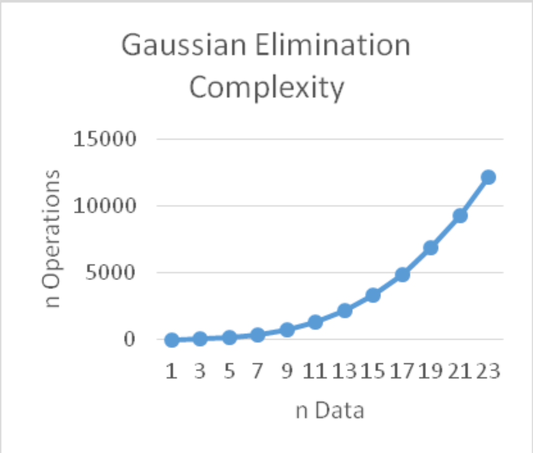

Figure 1. Gaussian Elimination's Complexity

However, Gaussian elimination algorithm isn't always the fastest for modern computing. Some programming libraries (for example BLAS, LINPACK, and LAPACK) can exploit specific computer hardware architecture and analyze the structure of the augmented matrix to select the best algorithm automatically. As in the case of matrix inversion, there are better alternatives to Gaussian elimination such as Coppersmith-Winograd algorithm [15] with the complexity of $O(n^{2.375477})$ and Strassen algorithm [16] with the complexity of $O(n^{2.807355})$. While in calculating the determinant of a small square matrix (i.e., 2x2, 3x3, or 4x4), Laplace expansion, Sarrus' rule, Triangle's rule, Chio's condensation, or Dodgson's condensation will be much simpler with just a single calculation [17].

As for the time efficiency, in the article of "System of Linear Equations, Gaussian Elimination," Gharib et al. [1] showed that the Gaussian elimination is more efficient over the Gauss-Jordan reduction in solving systems of linear equations. Table 2 shows the performance comparison of both Gaussian elimination and Gauss-Jordan reduction (tested by Gharib et al.).

Table 2. Execution Time of Gaussian Elimination

No of Variables	Gaussian's Execution Time (in Millisecond)	Gauss-Jordan's Execution Time (in Millisecond)
2	14	25
3	16	31
4	20	36
5	26	39
6	29	56
7	46	76

In their test, they used a similar Gaussian elimination algorithm with the example that used by Anany Levitin [7], but with an addition to the back-substitution algorithm and without pivoting. Originally, the algorithms (both Gaussian elimination and Gauss-Jordan reduction that were used by Gharib et al.) were used by Nai-Kuan Tsao [18] in NASA's technical memorandum of "On the Equivalence of Gaussian Elimination and Gauss-Jordan Reduction in Solving Linear Equations" in 1989. Here are the algorithms:

```
//begin reduction to tringular form
//using Gaussian Elimination
for i = 1 to n - 1 do
    for k = i + 1 to n do
        A[k,i] = fl (A[k,i] / A[i,i])
        for j = i + 1 to n + 1 do
            A[k,j] = fl (A[k,j] - A[k,i] *
    A[i,j]
//begin back-substitution
X[n] = fl (A[n,n + 1] / A[n,n])
for i = n - 1 downto 1 do
    for j = n downto i + 1 do
        A[i,n + 1] = fl (A[i,n + 1] -
    A[i,j] *
        X[j])
    X[i] = fl (A[i,n + 1] / A[i,i])
```

```
//begin reduction to diagonal form
//using Gauss-Jordan Reduction
for i = 1 to n do
    for k = 1 to n (except i) do
        A[k,i] = fl (A[k,i] / A[i,i])
        for j = i + 1 to n + 1 do
            A[k,j] = fl (A[k,j] - A[k,i] *
    A[i,j])
//begin solving diagonal system
for i = 1 to n do
    X[i] = fl (A[i,n + 1] / A[i,i])
```

In 2011, Michailidis and Margaritis [4] improved the efficiency of Gauss-Jordan reduction in multicore processing by implementing a pipelining algorithm. The general idea of the algorithm is that each thread executes n successive steps of Gauss-Jordan reduction on the rows it holds. In order to do that, each processing thread must obtain the index of the pivot row, send the pivot index immediately to the next thread, and then proceed with the steps of Gauss-Jordan method. In their report, Michailidis and Margaritis concluded that the implementation of pipelining algorithm achieved good performance in large systems. For practical use, this improvement hasn't widely implemented yet, especially for small equation systems, considering the computational cost of utilizing additional processor cores.

II.4. Problem-Solving Ability and Stability

Regarding the problem-solving ability, Foster [2] reported that the Gaussian elimination with partial pivoting could lead to a substantial failure. She also stated that "in n x n matrices, the error growth is proportional to 2^{n-1}, therefore, for moderate or large n, theoretically, there is a potential for disastrous error growth." As for Gauss-Jordan reduction, Peters and Wilkinson [19] reported that if the matrix is ill-conditioned, the residual (the error in the result) will often be much greater than that corresponding to the Gaussian elimination.

However, the Gaussian elimination method (with or without pivoting) can also fail to give a unique solution in a square matrix if the determinant is zero (the same goes for Gauss-Jordan reduction). It indicates that there are infinitely many solutions or otherwise no solution. Table 3 shows an example when the Gaussian elimination fails in giving a unique solution.

Table 3. Example of Gaussian Elimination's Failure without Pivoting

Equations	Augmented Matrix
$x + y + z = 3$ $2x + 4y + z = 8$ $6x + 10y + 4z = 22$	$\begin{bmatrix} 1 & 1 & 1 & 3 \\ 2 & 4 & 1 & 8 \\ 6 & 10 & 4 & 22 \end{bmatrix}$

Step 1: subtract Row 2 to 2 times Row 1 ($R_2 = R_2 - 2 * R_1$)	
$x + y + z = 3$ $2y - z = 2$ $6x + 10y + 4z = 22$	$\begin{bmatrix} 1 & 1 & 1 & 3 \\ 0 & 2 & -1 & 2 \\ 6 & 10 & 4 & 22 \end{bmatrix}$
Step 2: subtract Row 3 to 6 times Row 1 ($R_3 = R_3 - 6 * R_1$)	
$x + y + z = 3$ $2y - z = 2$ $4y - 2z = 4$	$\begin{bmatrix} 1 & 1 & 1 & 3 \\ 0 & 2 & -1 & 2 \\ 0 & 4 & -2 & 4 \end{bmatrix}$
Step 3: subtract Row 3 to 2 times Row 2 ($R_3 = R_3 - 2 * R_2$)	
$x + y + z = 3$ $2y - z = 2$ $0z = 0$	$\begin{bmatrix} 1 & 1 & 1 & 3 \\ 0 & 2 & -1 & 2 \\ 0 & 0 & 0 & 0 \end{bmatrix}$

In the example, the determinant of the augmented matrix is equal to zero. It can be calculated by multiplying all of the absolute diagonal values from the echelon form (which is $1 \cdot 2 \cdot 0 = 0$) or using Sarrus' rule:

$|A| = aei + bfg + cdh - ceg - bdi - afh$

$= 1 \cdot 4 \cdot 4 + 1 \cdot 1 \cdot 6 + 1 \cdot 2 \cdot 10 - 1 \cdot 4 \cdot 6 - 1 \cdot 2 \cdot 4 - 1 \cdot 1 \cdot 10$

$= 16 + 6 + 20 - 24 - 8 - 10$

$= 0$

On the result (see the last row on Table 3), the equation of $0z = 0$ seems trivial, but it actually indicates that there are infinitely many solutions available. In this case, 'z' can be replaced with any real number to specify a solution to the equations. If for instance the variable 'z' is equal to 't' number, then the second equation will become $2y - t = 2$, as the result $y = ½ (t + 2)$.

As for the stability, Gene H. Golub and Charles F. Van Loan [20] described that for general matrices, Gaussian elimination with partial pivoting is usually considered to be stable. On the other hand, Peters and Wilkinson [19] reported that the overall stability of Gauss-Jordan reduction is comparable with that corresponding to the Gaussian elimination using partial pivoting and back-substitution.

3. Conclusion

Gauss-Jordan reduction and Gaussian elimination have similar functions, which are to solve a system of linear equations, find the rank and base of a matrix, and solve the inverse of an invertible square matrix (in exception of calculating the determinant of a square matrix, which Gauss-Jordan reduction is incapable). Both methods are quite similar in terms of algorithm, arithmetic complexity, and stability. In theory, although the Gaussian elimination shares the same complexity of $O(n^3)$ as Gauss-Jordan reduction, in practice, the latter has lower time efficiency and greater residual (the error in the result) in problem-solving ability.

References

[1] S. Gharib, S. R. Ali, R. Khan, N. Munir, and M. Khanam, "System of Linear Equations, Gaussian Elimination," *Comput. Sci. Technol. C Softw. Data Eng.*, vol. 15, no. 5, pp. 23–26, 2015.

[2] L. V. Foster, "Gaussian Elimination with Partial Pivoting can Fail in Practice," *Matrix Anal. Appl.*, vol. 15, no. 4, pp. 1354–1362, 1994.

[3] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*. SIAM: Society for Industrial and Applied Mathematics; Har/Cdr edition, 2000.

[4] P. D. Michailidis and K. G. Margaritis, "Open Multi Processing (OpenMP) of Gauss-Jordan Method for Solving System of Linear Equations," in *Proceedings of the 11th IEEE International Conference on Computer and Information Technology, CIT 2011*, 2011, pp. 314–319.

[5] T. A. Smaglichenko, "Modification of Gaussian Elimination for the Complex System of Seismic Observations," *Complex Syst.*, vol. 20, pp. 229–241, 2012.

[6] K. E. Atkinson, *An Introduction to Numerical Analysis (2nd Edition)*, 2nd Ed. Iowa: John Wiley & Sons, 1989.

[7] A. Levitin, *Introduction to the Design and Analysis of Algorithms*, 3rd Ed. Philadelphia: Pearson, 2011.

[8] J. F. Grcar, "Mathematicians of Gaussian Elimination," *Not. Am.*

Math. Soc., vol. 58, no. 6, pp. 782–792, 2011.

[9] Princeton University, *The Princeton Companion to Mathematics*, 1st Ed. Princeton: Princeton University Press, 2008.

[10] S. C. Althoen and R. McLaughlin, “Gauss-Jordan Reduction: A Brief History,” *The American Mathematical Monthly*, vol. 94, no. 2. pp. 130–142, 1987.

[11] L. Smith and J. Powell, “An Alternative Method to Gauss-Jordan Elimination : Minimizing Fraction Arithmetic,” *Math. Educ.*, vol. 20, no. 2, pp. 44–50, 2011.

[12] J. Ji, “Gauss-Jordan Elimination Methods for the Moore-Penrose Inverse of a Matrix,” *Linear Algebra Appl.*, vol. 437, no. 7, pp. 1835–1844, 2012.

[13] D. Dasgupta, “In-Place Matrix Inversion by Modified Gauss-Jordan Algorithm,” *Appl. Math.*, vol. 4, pp. 1392–1396, 2013.

[14] A. Edelman, “The Complete Pivoting Conjecture for Gaussian Elimination is False,” *Math. J.*, pp. 1–11, 1992.

[15] D. Coppersmith and S. Winograd, “Matrix Multiplication via Arithmetic Progressions,” *Symb. Comput.*, vol. 9, pp. 251–280, 1990.

[16] J. Huang, T. M. Smith, G. M. Henry, and R. A. Van De Geijn, “Strassen’s Algorithm Reloaded,” in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis SC’16*, 2016, pp. 690–701.

[17] D. Hajrizaj, “New Method to Compute the Determinant of a 3x3 Matrix,” *Int. J. Algebr.*, vol. 3, no. 5, pp. 211–219, 2009.

[18] N. Tsao, *On the Equivalence of Gaussian Elimination and Gauss-Jordan Reduction in Solving Linear Equations*. Cleveland: National Aeronautics and Space Administration, 1989.

[19] G. Peters and J. H. Wilkinson, “On the stability of Gauss-Jordan elimination with pivoting,” *Commun. ACM*, vol. 18, no. 1, pp. 20–24, 1975.

[20] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd Ed. Baltimore: Johns Hopkins University Press, 1996.