Gaussian - Jordan elimination

Kyle Ng

November 8 2023

Contents

1 Introduction **1**
1.1 Overview of Gauss-Jordan elimination 1
1.2 Historical background 2

2 System of linear equations **2**
2.1 Definition of a system of linear equations: 2
2.2 Motivation for using Gauss-Jordan elimination 2

3 Gauss-Jordan elimination algorithm **3**
3.1 Explanation of the Gauss-Jordan elimination process 3
3.2 Pivoting strategies 3
3.3 Row operations and their impact on the matrix 4

4 Row-echelon form **4**
4.1 Definition and properties of row echelon form 4
4.2 Steps to convert a matrix to row-echelon form 5
4.3 Triangular matrix 5

5 Row reduced echelon form **6**
5.1 Definition of reduced row echelon form 6
5.2 Steps to convert a matrix in row echelon form to reduced row-echelon form 6

6 Solving systems of linear equations **6**
6.1 2 x 2 matrix example 6
6.2 3 x 3 matrix example 7
6.3 4 x 4 matrix example 8

7 Matrix inversion **9**
7.1 2 x 2 matrix example 9
7.2 3 x 3 matrix example 10

8 Gauss-Jordan elimination algorithm (MATLAB) **11**
8.1 Solving for matrix inverse 11
8.2 Solving for system of linear equations 12

1 Introduction

1.1 Overview of Gauss-Jordan elimination

It is one of the most useful algorithms in linear algebra that can be applied to **solve systems of linear equations as well as perform certain matrix computations**. Called after for mathematician Carl Friedrich Gauss and Wilhelm Jordan, it is an extension of Gaussian elimination whose purpose is to convert any given matrix into its reduced row-echelon form. Analyzing and solving a **complex system of linear equations or a matrix** can be greatly facilitated by simplifying it in order to expose its core characteristics.

Essentially, Gauss-Jordan elimination consecutively performs elementary row operations on a matrix in order to obtain its **row-echelon form** before finally reducing it to its **lower row-echelon form**. This results into a matrix that depicts the solution domain, which gives information on the type of the system—single/multiple solution or non-existent solutions.

1.2 Historical background

The roots of Gaussian elimination and hence of Gauss- Jordan elimination can be found in old Chinese mathematics texts. Nonetheless, it was the effort of European mathematicians in the 18th and 19th century that brought about the basis for these strategies. A German mathematician – Carl Friedrich Gauss contributed major aspects in systems of linear equations elimination method.

Wilhelm Jordan extended this algorithm into modern version of Gauss-Jordan elimination and made it popular among mathematicians. Gauss-Jordan elimination became a classical procedure of numerical linear algebra and is applied in various studies like physics, engineering, computer science and others.

This algorithm is crucial not only for the history of mathematics, but also in practical applications today. It offers solutions to many linear algebra problems because one could not talk about systems of linear equations and matrices without discussing Gauss-Jordan elimination. Considering the historical meaning, admiration towards Gauss-Jordan elimination is deepened.

2 System of linear equations

2.1 Definition of a system of linear equations:

Linear equations consist of different sets of variables whereby each variable has its corresponding mathematical representation or equation. The general form of a linear equation n unknowns and coefficients can be written as:

$$a_{11}x_1 + a_{12}x_2 + \cdots + a_nx_n + b_1 = 0$$

$$a_{21}x_1 + a_{22}x_2 + \cdots + a_nx_n + b_2 = 0$$

$$\vdots$$

$$a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n + b_m = 0$$

where x_1, x_2, x_n are the unknowns, a_{11}, a_{12}, a_{mn} are the coefficients of the system, and b_1, b_2, b_m are the constant terms.

2.2 Motivation for using Gauss-Jordan elimination

Squaring off or handling a system of linear equations manually may not be easy when more than two equations arise; this is because of increased probability of error. This is where a systematic procedure called Gauss-Jordan elimination comes into the picture. This technique has proven to be very useful in obtaining solutions to linear system equations. This is what motivates it to reduce the system into **reduced row-echelon form (RREF)** whose solution space presents its elements with more clarity.

A very simple process that employs the use of elementary row operations simplify steps-by-steps the systems to lead into a matrix that can be easily interpreted. The obtained matrix makes it easy to extract solves, whether the system is unique, infinite or just unsolvable. The systematic approach makes solving easier, but more importantly, it allows a better understanding of the relationships that are linearly related, and the nature of solution space.

Essentially, Gauss- Jordan elimination is efficient, transparent and can deal with different systems of linear equations. Linear algebra is one of the most important tools used by mathematicians, scientists, and engineers when they try to solve problems with linear connections between different variables.

3 Gauss-Jordan elimination algorithm

3.1 Explanation of the Gauss-Jordan elimination process

The Gauss-Jordan elimination is one of the many ways that can be employed when solving Linear Algebra problems through matrices by transforming a given matrix into its **reduced row echelon form** showing vital characteristics of linear system and its solution. The algorithm proceeds through several key steps:

1. **Augmented matrix setup**: Begin with an augmented matrix $[A|B]$, where **A is the coefficient matrix**, and **B is the constant matrix**. The goal is to transform this augmented matrix into $[I|C]$, where **I is the identity matrix**, and **C contains the solution to the system of equations**.

2. **Pivoting**: The first non zero element on the current row should be selected as a pivot. **If the pivot is zero**, switch to another row of a smaller entry for the purpose of creating a nonzero pivot. Divide every entry of the pivot row by the pivot element and normalize it.

3. **Row operations**: Create zeros below, as well as above the pivot with elementary row operations. **Row scaling**: Each row multiplied by a non-zero constant. **Row addition/subtraction**: The sum or difference can be taken by multiply one row with another. **Row swapping**: Swap two rows if needed.

4. **Iterative process**: For every row or column, execute the pivoting and row operations until the matrix is converted into **row echelon form**. Then proceed to the **reduced row echelon form** where the leading entries create a step-shape structure.

5. **Operators and observables**: perform on **row reduced echelon form row operations** by beginning with bottom row that zero should be obtained above every pivot. The **solution is on the right-hand side the matrix**, which represents the solution of a system of linear equations.

3.2 Pivoting strategies

Pivoting is one of the critical steps in the Gauss-Jordan elimination algorithm that selects suitable pivot elements to simplify the procedure and improve numerical stability. There are two main pivoting strategies: partial pivoting and complete pivoting.

1. **Partial pivoting**: In partial pivoting, the pivot element is chosen as the highest absolute value among those elements in the current column. In this way, the pivot is guaranteed not to be too small while minimizing rounding effects and number overflow.

   Motivation for partial pivoting:

   Numerical stability: Partial pivoting ensures that no division is taken by a very small number so that major errors are avoided in elimination procedures.

   Addressing small entries: The method comes in handy even when working on matrices containing small numbers to avoid the division of a near zero value by a similar quantity.

   Procedure for partial pivoting:

   1. Determine which column has largest absolute value from the current row, then choose this particular element as a pivot.

   2. Replace the current row with a lower one if there is no pivot value of zero.

   3. For normalization of a pivot row, divide each of its entries into pivot element.

2. **Complete pivoting**: Complete pivoting is more elaborate in terms of picking the smallest absolute value from each sub matrix consisting the present row and column, whereas partial pivoting selects the biggest absolute value of the whole submatrix. Moreover, this method also improves numerical stability in taking into account how pivots are affected by the rows and the columns.

Motivation for complete pivoting:

Enhanced stability: Pivoting that considers rows and columns surrounding the pivot as a complete process increases the level of prudence which in turn mitigates against big divides.

Procedure for Complete Pivoting:

1. Determine the amount with the greatest numerical magnitude from all of the sub-matrix (the current line and column).

2. To introduce a new pivot, swap the row and column which contains this element in opposite directions to the current ones.

3. Divide all elements of the pivot row by the pivot element and obtain a normalized pivot row.

3.3 Row operations and their impact on the matrix

The basic row operations that lead to a matrix into **reduced row echelon form** are the key steps of the Gaussian elimination. There are three elementary row operations: scaling (by multiplication or division), row additions/subtractions, and row interchanges.

1. **Elementary row operations**

   Scaling: Multiply each element of a particular row by a non-zero quantity.

   Row addition/subtraction: Multiple of adding or subtracting one row from another.

   Row interchange: Swapping two rows.

2. **Impact on the matrix**:

   1. The row operations make systematic alterations to the matrix through a gradual introduction of zeros both above and under the pivot element.

   2. These are the operations that retain the equality between the original system of linear equations and the resulting matrix.

   3. The last matrix is reduced echelon form and it represents compactly and meaningfully all solution space.

4 Row-echelon form

4.1 Definition and properties of row echelon form

Row-echelon form is an important step in the Gaussian elimination process, putting the matrix in an orderly fashion that helps to solve a system of linear equations. Such features or characteristics are defined in such a process that leads to transformations towards **reduced row echelon form**.

Properties of row-echelon form:

1. The first entry in each nonzero row is the leading entry and it lies immediately to the right of the leading entry in the row above it.

2. One appears in the upper most part of every row.

3. Zero entries occur beneath every leading subentry within all rows.

4. Any zeroes that occur in rows are put at the tail end.

   These are some of the important properties that make it easy to understand, as well as more manageable in solving linear systems or other operations with matrices. This process of **Gauss-Jordan elimination** comprises successive application of appropriate row operations whose aim here is to convert that initial matrix into its **row echelon form**.

4.2 Steps to convert a matrix to row-echelon form

1. **Pivoting**: Choose a pivot element. A typical selection of a pivot is usually the first nonzero entry for each row. On the other hand, full, partial or any other pivoting techniques could be applied for better numerical stability.

2. **Row operations**: Create zeros below the pivot by performing row operations. These are mostly calculations that involve removing multiples of one row by another. Basically, it intends to convert the matrix into a triangular one by setting zeros below its pivots.

3. **Iterative process**: Reiterate the above processes for every row till the whole matrix meets the characteristics of row echelon form.

4. **Completing the transformation**: After the matrix satisfies the characteristics of REF, the leading elements in each row can be taken as 1 through the operations of row scaling provided the necessity ensues.

4.3 Triangular matrix

Triangular matrix is a kind of square matrix whose characteristic distribution of non-zero elements makes it unique. A triangular matrix is one where zero entries are present to the left of or right of the main diagonal. Every element on the main diagonal stretching from the upper left corner down to the lower right corresponds to a non-zero value in that order.

There are two primary types of triangular matrices:

1. **Upper triangular matrix**:

   An upper triangular matrix has zero entry below the main diagonal. Non-zero elements are found on as well as above the main diagonal. A general upper triangle matrix upper triangular matrix U of order n is given by:

$$\mathrm{U} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ 0 & a_{22} & a_{23} & \dots & a_{2n} \\ 0 & 0 & a_{33} & \dots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & a_{nn} \end{bmatrix}$$

2. **Lower triangular matrix**:

   All the entries in a lower triangular matrix lie above the main diagonal and therefore are zero. Non-zero elements are found either in main diagonal or under it. A general form for a lower triangular matrix L of order n is given by:

$$\mathrm{L} = \begin{bmatrix} a_{11} & 0 & 0 & \dots & 0 \\ a_{21} & a_{22} & 0 & \dots & 0 \\ a_{31} & a_{32} & a_{33} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{bmatrix}$$

3. **Properties and significance**:

   Efficient storage: Triangular matrices are organized in some form of order, which means that they consume less storage space than normal matrices. This particular property is critical when it comes to memory optimization.

   Computational efficiency: Some computational operations like multiplication of triangular matrices and solving system of linear equations may be simpler than in square cases. This structure leads to simplified algorithms using zero entry exploitation.

Solving linear systems: Solving systems of linear equations is dependant on triangular matrices. Back-substitution is one of the essential stages of solving linear systems and here it is easier because we have a triangular form.

Applications in numerical methods: In various numerical approaches such as LU decomposition, Cholesky decomposition, and Gaussian elimination, triangular matrices are widely applied. Here, these techniques take advantage of the characteristics of a triangular matrix toward better numerical stability and performance.

5 Row reduced echelon form

5.1 Definition of reduced row echelon form

Reduced row echelon form is a more specialized form than **row echelon form**. It contains additional structures and features that make is possible to simplify systems of linear equations and perform certain matrix operations more easily.

Properties of reduced row echelon form:

1. It is in row echelon form.
2. All leading entries (pivots) are 1.
3. The leading entry in each row is the only nonzero entry in its column.
4. All entries above and below a leading entry are zeros.

These additional conditions set **row reduced echelon form** apart by ensuring not only triangularity but also uniqueness of representation and ease of interpretation. Matrices in **row reduced echelon form** are especially advantageous for determining unique, parametric, or inconsistent solutions to systems of linear equations.

5.2 Steps to convert a matrix in row echelon form to reduced row-echelon form

1. **Gaussian elimination**: Eliminate any nonzero entries above each pivot ensuring zeroes both above and under the leading entry of each row using row operations.
2. **Backward substitution**: Adjust the matrix such that the first element of each row is equal to 1. For instance, this can entail lengthening the rows so that vertical entries number one appear as tall as possible.
3. **Iterative process**: Systematically apply row operations to all rows, including each pivot until the whole matrix becomes suitable for the properties of the **row reduced echelon form**.
4. **Completing the transformation**: Ascertain if the matrix meets the strict necessities that a **row reduced echelon form** should meet i.e., there are zeros adjacent to each pivot as well as having zeros in every other column save for one containing one non-zero element.

6 Solving systems of linear equations

6.1 2 x 2 matrix example

$$\begin{bmatrix} 2x + 3y & = 8 \\ 4x - 2y & = 2 \end{bmatrix}$$

Step 1: Augmented matrix

$$\begin{bmatrix} 2 & 3 & | & 8 \\ 4 & -2 & | & 2 \end{bmatrix}$$

Step 2: Row echelon form

$R_2 = R_2 - 2R_1$

$$\begin{bmatrix} 2 & 3 & | & 8 \\ 0 & -8 & | & -14 \end{bmatrix}$$

$R_2 = -\frac{1}{8}R_2$

$$\begin{bmatrix} 2 & 3 & | & 8 \\ 0 & 1 & | & \frac{7}{4} \end{bmatrix}$$

$R_1 = R_1 - 3R_2$

$$\begin{bmatrix} 2 & 0 & | & \frac{17}{4} \\ 0 & 1 & | & \frac{7}{4} \end{bmatrix}$$

This matrix is now in **row echelon form**.

Step 3: Reduced row echelon form

$R_1 = \frac{1}{2}R_1$

$$\begin{bmatrix} 1 & 0 & | & \frac{17}{8} \\ 0 & 1 & | & \frac{7}{4} \end{bmatrix}$$

Conclusion

The **reduced row echelon form** of the augmented matrix represents the solutions to the system of equations. In this case, $x = \frac{17}{8}$ and $y = \frac{7}{4}$.

6.2 3 x 3 matrix example

$$\begin{bmatrix} 2x + 3y - z & | & 7 \\ 4x + 7y + z & | & 1 \\ 6x + 11y - 3z & | & 8 \end{bmatrix}$$

Step 1: Augmented matrix

$$\begin{bmatrix} 2 & 3 & -1 & | & 7 \\ 4 & 7 & 1 & | & 1 \\ 6 & 11 & -3 & | & 8 \end{bmatrix}$$

Step 2: Row echelon form

$R_2 = R_2 - 2R_1$

$$\begin{bmatrix} 2 & 3 & -1 & | & 7 \\ 0 & 1 & 3 & | & -13 \\ 0 & 2 & -3 & | & -6 \end{bmatrix}$$

$R_3 = R_3 - 3R_1$

$$\begin{bmatrix} 2 & 3 & -1 & | & 7 \\ 0 & 1 & 3 & | & -13 \\ 0 & 0 & -9 & | & 20 \end{bmatrix}$$

This matrix is now in **row echelon form**.

Step 3: Reduced row echelon form

$R_3 = -\frac{1}{9}R_3$

$$\begin{bmatrix} 2 & 3 & -1 & | & 7 \\ 0 & 1 & 3 & | & -13 \\ 0 & 0 & 1 & | & -\frac{20}{9} \end{bmatrix}$$

$R_2 = R_2 - 3R_3$

$$\begin{bmatrix} 2 & 0 & -10 & | & 33 \\ 0 & 1 & 0 & | & -\frac{59}{9} \\ 0 & 0 & 1 & | & -\frac{20}{9} \end{bmatrix}$$

$R_1 = R_1 - 3R_3$

$$\begin{bmatrix} 2 & 0 & 0 & | & \frac{73}{9} \\ 0 & 1 & 0 & | & -\frac{59}{9} \\ 0 & 0 & 1 & | & -\frac{20}{9} \end{bmatrix}$$

Conclusion

The **reduced row echelon form** of the augmented matrix represents the solutions to the system of equations. In this case, $x = \frac{73}{18}$, $y = -\frac{59}{9}$, and $z = -\frac{20}{9}$.

6.3 4 x 4 matrix example

$$\begin{bmatrix} 2x + 3y - z + 4w & = 8 \\ 4x - 2y + 6z - w & = 2 \\ x + 5y + 2z - 3w & = 5 \\ 3x - y + 4z + 2w & = 9 \end{bmatrix}$$

Step 1: Augmented matrix

$$\begin{bmatrix} 2 & 3 & -1 & 4 & | & 8 \\ 4 & -2 & 6 & -1 & | & 2 \\ 1 & 5 & 2 & -3 & | & 5 \\ 3 & -1 & 4 & 2 & | & 9 \end{bmatrix}$$

Step 2: Row echelon form

$R_2 = R_2 - 2R_1$

$$\begin{bmatrix} 2 & 3 & -1 & 4 & | & 8 \\ 0 & -8 & 8 & -9 & | & -14 \\ 1 & 5 & 2 & -3 & | & 5 \\ 3 & -1 & 4 & 2 & | & 9 \end{bmatrix}$$

$R_2 = -\frac{1}{8}R_2$

$$\begin{bmatrix} 2 & 3 & -1 & 4 & | & 8 \\ 0 & 1 & -1 & \frac{9}{8} & | & \frac{7}{4} \\ 1 & 5 & 2 & -3 & | & 5 \\ 3 & -1 & 4 & 2 & | & 9 \end{bmatrix}$$

$R_1 = R_1 - \frac{3}{2}R_3$

$$\begin{bmatrix} 2 & 0 & 1 & \frac{11}{4} & | & \frac{11}{2} \\ 0 & 1 & -1 & \frac{9}{8} & | & \frac{7}{4} \\ 1 & 5 & 2 & -3 & | & 5 \\ 3 & -1 & 4 & 2 & | & 9 \end{bmatrix}$$

This matrix is now in **row echelon form**.

Step 3: Reduced row echelon form

$R_1 = \frac{1}{2}R_1$

$$\begin{bmatrix} 1 & 0 & \frac{1}{2} & \frac{11}{8} & | & \frac{11}{4} \\ 0 & 1 & -1 & \frac{9}{8} & | & \frac{7}{4} \\ 1 & 5 & 2 & -3 & | & 5 \\ 3 & -1 & 4 & 2 & | & 9 \end{bmatrix}$$

Conclusion

The **reduced row echelon form** of the augmented matrix represents the solutions to the system of equations. In this case, $x = \frac{11}{8}$, $y = \frac{9}{8}$, $z = 2$, and $w = 1$.

7 Matrix inversion

7.1 2 x 2 matrix example

Let's consider the matrix A:

$$A = \begin{bmatrix} 2 & 3 \\ 1 & 4 \end{bmatrix}$$

The augmented matrix is:

$$\begin{bmatrix} 2 & 3 & | & 1 & 0 \\ 1 & 4 & | & 0 & 1 \end{bmatrix}$$

Perform Gaussian Jordan elimination:

$R_2 = R_2 - \frac{1}{2}R_1$

$$\begin{bmatrix} 2 & 3 & | & 1 & 0 \\ 0 & \frac{5}{2} & | & -\frac{1}{2} & 1 \end{bmatrix}$$

$R_2 = \frac{2}{5}R_2$

$$\begin{bmatrix} 2 & 3 & | & 1 & 0 \\ 0 & 1 & | & -\frac{1}{5} & \frac{2}{5} \end{bmatrix}$$

$R_1 = R_1 - 3R_2$

$$\begin{bmatrix} 2 & 0 & | & \frac{8}{5} & -\frac{6}{5} \\ 0 & 1 & | & -\frac{1}{5} & \frac{2}{5} \end{bmatrix}$$

$R_1 = \frac{1}{2}R_1$

$$\begin{bmatrix} 1 & 0 & | & \frac{4}{5} & -\frac{3}{5} \\ 0 & 1 & | & -\frac{1}{5} & \frac{2}{5} \end{bmatrix}$$

The inverse of the matrix A is:

$$A^{-1} = \begin{bmatrix} \frac{4}{5} & -\frac{3}{5} \\ -\frac{1}{5} & \frac{2}{5} \end{bmatrix}$$

7.2 3 x 3 matrix example

Let's consider the matrix B:

$$B = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

The augmented matrix is:

$$\begin{bmatrix} 1 & 2 & 3 & | & 1 & 0 & 0 \\ 4 & 5 & 6 & | & 0 & 1 & 0 \\ 7 & 8 & 9 & | & 0 & 0 & 1 \end{bmatrix}$$

Perform Gaussian Jordan elimination:

$R_2 = R_2 - 4R_1$

$$\begin{bmatrix} 1 & 2 & 3 & | & 1 & 0 & 0 \\ 0 & -3 & -6 & | & -4 & 1 & 0 \\ 7 & 8 & 9 & | & 0 & 0 & 1 \end{bmatrix}$$

$R_3 = R_3 - 7R_1$

$$\begin{bmatrix} 1 & 2 & 3 & | & 1 & 0 & 0 \\ 0 & -3 & -6 & | & -4 & 1 & 0 \\ 0 & -6 & -12 & | & -7 & 0 & 1 \end{bmatrix}$$

$R_3 = -\frac{1}{3}R_2$

$$\begin{bmatrix} 1 & 2 & 3 & | & 1 & 0 & 0 \\ 0 & -3 & -6 & | & -4 & 1 & 0 \\ 0 & 2 & 4 & | & \frac{7}{3} & 0 & -1 \end{bmatrix}$$

$R_2 = R_2 - \frac{2}{3}R_3$

$$\begin{bmatrix} 1 & 2 & 3 & | & 1 & 0 & 0 \\ 0 & -3 & 0 & | & -\frac{10}{3} & 1 & \frac{2}{3} \\ 0 & 2 & 4 & | & \frac{7}{3} & 0 & -1 \end{bmatrix}$$

$R_1 = R_1 - 2R_2$

$$\begin{bmatrix} 1 & 0 & -1 & | & \frac{8}{3} & -2 & -\frac{4}{3} \\ 0 & -3 & 0 & | & -\frac{10}{3} & 1 & \frac{2}{3} \\ 0 & 2 & 4 & | & \frac{7}{3} & 0 & -1 \end{bmatrix}$$

$R_2 = -\frac{1}{3}R_2$

$$\begin{bmatrix} 1 & 0 & -1 & | & \frac{8}{3} & -2 & -\frac{4}{3} \\ 0 & 1 & 0 & | & \frac{10}{9} & -\frac{1}{3} & -\frac{2}{9} \\ 0 & 2 & 4 & | & \frac{7}{3} & 0 & -1 \end{bmatrix}$$

$R_3 = R_3 - 2R_2$

$$\begin{bmatrix} 1 & 0 & -1 & | & \frac{8}{3} & -2 & -\frac{4}{3} \\ 0 & 1 & 0 & | & \frac{10}{9} & -\frac{1}{3} & -\frac{2}{9} \\ 0 & 0 & 4 & | & \frac{5}{3} & \frac{2}{3} & -\frac{7}{9} \end{bmatrix}$$

$R_3 = \frac{1}{4}R_3$

$$\begin{bmatrix} 1 & 0 & -1 & | & \frac{8}{3} & -2 & -\frac{4}{3} \\ 0 & 1 & 0 & | & \frac{10}{9} & -\frac{1}{3} & -\frac{2}{9} \\ 0 & 0 & 1 & | & \frac{5}{12} & \frac{1}{6} & -\frac{7}{36} \end{bmatrix}$$

$R_1 = R_1 + R_3$

$$\begin{bmatrix} 1 & 0 & 0 & | & \frac{37}{4} & -\frac{5}{3} & -\frac{11}{12} \\ 0 & 1 & 0 & | & \frac{10}{9} & -\frac{1}{3} & -\frac{2}{9} \\ 0 & 0 & 1 & | & \frac{5}{12} & \frac{1}{6} & -\frac{7}{36} \end{bmatrix}$$

The inverse of the matrix B is:

$$B^{-1} = \begin{bmatrix} \frac{37}{4} & -\frac{5}{3} & -\frac{11}{12} \\ \frac{10}{9} & -\frac{1}{3} & -\frac{2}{9} \\ \frac{5}{12} & \frac{1}{6} & -\frac{7}{36} \end{bmatrix}$$

8 Gauss-Jordan elimination algorithm (MATLAB)

8.1 Solving for matrix inverse

Below is the MATLAB code implementing Gauss-Jordan elimination for a matrix inverse:

```
function [A_inverse] = gaussJordan(A)
    % Gauss-Jordan elimination to find inverse of matrix A

    [m, n] = size(A);
    if m ~= n
        error('Input matrix must be square.');
    end

    % Augmenting the matrix with the identity matrix
    augmented_matrix = [A, eye(n)];

    % Applying Gauss-Jordan elimination
    for i = 1:n
        % Pivoting
        pivot = augmented_matrix(i, i);
        if pivot == 0
            error('Matrix is singular or badly conditioned.');
        end
        augmented_matrix(i, :) = augmented_matrix(i, :) / pivot;

        % Row operations to get zeros in other rows
        for j = 1:n
            if i ~= j
                factor = augmented_matrix(j, i);
                augmented_matrix(j, :) = augmented_matrix(j, :) - factor * augmented_matrix(i, :);
            end
        end
    end

    % Extracting the inverse from the augmented matrix
    A_inverse = augmented_matrix(:, n + 1:end);
end
```

8.2 Solving for system of linear equations

Below is the MATLAB code implementing Gauss-Jordan elimination to solve $Ax = b$:

```
function x = gaussJordanSolve(A, b)
    % Gauss-Jordan elimination to solve Ax = b

    [m, n] = size(A);
    if m ~= n
        error('Coefficient matrix must be square.');
    end

    if size(b, 1) ~= n || size(b, 2) ~= 1
        error('Invalid dimensions for the right-hand side vector b.');
    end

    % Augmenting the matrix with the right-hand side vector
    augmented_matrix = [A, b];

    % Applying Gauss-Jordan elimination
    for i = 1:n
        % Pivoting
        pivot = augmented_matrix(i, i);
        if pivot == 0
            error('Matrix is singular or badly conditioned.');
        end
        augmented_matrix(i, :) = augmented_matrix(i, :) / pivot;

        % Row operations to get zeros in other rows
        for j = 1:n
            if i ~= j
                factor = augmented_matrix(j, i);
                augmented_matrix(j, :) = augmented_matrix(j, :) - factor * augmented_matrix(i, :);
            end
        end
    end

    % Extracting the solution from the augmented matrix
    x = augmented_matrix(:, n + 1);
end
```