

Article

MATLAB-Based Learning of Simultaneous Linear Equation Systems

Mochammad Sholikhin^{1*}

¹ Universitas Islam Lamongan; e-mail : sholikhin@unisla.ac.id

* Corresponding Author : Mochammad Sholikhin

Abstract: Problems involving mathematical models are commonly found across various scientific disciplines, such as physics, chemistry, biology, economics, and engineering. These mathematical models are used to represent complex real-world phenomena in order to analyze and understand them quantitatively. However, in practice, these models often take the form of complex, nonlinear, or high-dimensional equation systems, which are not easily solved using standard analytical methods or conventional algebraic formulas. To overcome these limitations, numerical methods are employed as an alternative approach. Numerical methods provide solutions to mathematical problems by relying on computational techniques and iterative procedures. One of the key topics within numerical methods is the solution of Simultaneous Linear Equation Systems (SLES), which frequently arise in practical applications such as structural analysis, fluid dynamics, and economic modeling including structural assessment, flow dynamics, and economic forecasting. Solving these systems often requires a large number of iterative calculations, making manual computation inefficient and time-consuming. Therefore, the use of computational software such as MATLAB becomes essential in both learning and applying these methods. MATLAB provides a broad array of functions and numerical tools this significantly support an efficient and accurate solution of linear equation systems. In an educational context, the use of MATLAB not only simplifies the computational process but also aids students in understanding fundamental concepts of numerical methods through visualization, simulation, and digital experimentation. Thus, integrating MATLAB into the learning process of numerical methods greatly enhances teaching and learning effectiveness, while also strengthening students' computational skills in solving real-world scientific and engineering problems.

Keywords: matlab; linear equation system; matrix inverse.

1. Introduction

An ability to understand and solve systems of simultaneous linear equations [1] is a fundamental competency in mathematics and engineering education. These systems appear frequently in various real-world applications, including electrical circuits, structural analysis, economic models, and data fitting problems. However, for many students, mastering the concepts and methods used to solve such systems particularly when dealing with multiple variables, can be challenging when approached through traditional, theoretical instruction alone. To address this issue, the integration of technology into the learning process has become increasingly important. One such technological tool is MATLAB [2] (Matrix Laboratory), a high-level programming environment widely used in both academic and professional settings for numerical computation, data analysis, and algorithm development. MATLAB's [3] matrix-oriented structure and powerful visualization capabilities make it

Received: April 16, 2025

Revised: April 30, 2025

Accepted: May 29, 2025

Available Online : Juni 09, 2025

Curr. Ver.: Juni 09, 2025



Copyright: © 2025 by the authors.

Submitted for possible open li-

cense ([https://creativecommons-](https://creativecommons.org/licenses/by-sa/4.0/)

[mons.org/licenses/by-sa/4.0/](https://creativecommons.org/licenses/by-sa/4.0/))

particularly well-suited for teaching and learning topics in linear algebra. This study proposes a MATLAB[4]-based learning approach as an alternative to traditional teaching methods for simultaneous linear equation systems[5]. By incorporating interactive simulations, visual representations, and real-world problem-solving activities into the learning process, this approach aims to improve students' conceptual understanding and practical skills. The use of MATLAB not only facilitates deeper comprehension of abstract mathematical principles but also helps students develop digital literacy and computational thinking, skills that are increasingly essential in the modern STEM landscape. Through this approach, students are expected to engage more actively with the learning material, explore multiple solution strategies, and better appreciate the applications of linear algebra in various fields. This paper outlines the design, implementation, and potential benefits of using MATLAB as an instructional tool to enhance the learning experience for systems of simultaneous linear equations[6].

2. Preliminaries or Related Work or Literature Review

2.1. Definition of a Linear Equation

A linear equation [7] is one of the most fundamental forms of equations in algebra. Although it is considered basic, it plays a crucial role in understanding more advanced mathematical concepts. A linear equation[8] is characterized by the presence of a single variable raised to the power of one (first degree), accompanied by constants. Because the variable is only to the first power and does not involve higher-order exponents, roots, or other non-linear operations, the relationship between the variable and the constants is direct and proportional.

The term “linear” comes from the word “line,” referring to its graphical characteristic: when a linear equation[9] is plotted on a Cartesian coordinate system, it always produces a straight line. This property makes linear equations an effective tool for illustrating and analyzing the relationship between two quantities that are linearly related.

Linear equations are not only relevant in mathematical theory but also have numerous practical applications in everyday life. For instance, in economics, linear equations can be used to calculate production costs; in physics, they help analyze the relationship between speed and time; and in engineering, they assist in planning and performing technical calculations for various projects. For these reasons, understanding linear equations is essential not only for students but also for anyone involved in science and technology.

For general, the standard form of a one-variable linear equation[10] is written as:

$$cy + d = 0$$

Explanation:

- c is the factor, the number indicating how many times the variable is multiplied.
- y represents the element, a symbol denoting an unknown value.
- d is the constant, a fixed number without a variable.

It is important to note that linear equations [11] have a counterpart in mathematics known as linear inequalities. While linear equations state that two expressions are equal, linear inequalities express relationships such as greater than, less than, greater than or equal to, and maximum or at most. However, in this discussion, we will focus on linear equations, and inequalities will be addressed at another time.

Before we dive deeper into solving linear equations[12], it is important to first understand the basic algebraic terms that form the core components of these equations. Here are explanations of four key terms:

- Coefficient is the number attached to the variable and indicates how many times the variable is counted. For example, in the equation $3y + 2 = 0$, the number 3 states a coefficient from y , meaning that y equals multiplied by three.
- A variable tells a symbol that stands for an unknown value. Variables are essential in algebra as they allow us to form equations from uncertain situations. Commonly, variables are denoted by letters such as x , y , a , b , etc.
- Constant is a fixed value unaffected by the variable. In the equation $3x + 2 = 0$, the number 2 is the constant because it does not change and is not associated with a variable.
- Term is a part of an equation that can consist of a coefficient, variable, and/or constant, separated by addition (+) or subtraction (−) signs. In the example $3x + 2 = 0$, there are two terms: $3x$ and 2.

By understanding these basic components, we will have a solid foundation for comprehending and solving various forms of one-variable linear equations[13]. This knowledge will be especially helpful when we begin discussing solution methods, such as moving terms to the other side, simplifying the equation, or graphing it. Next, let's explore in more detail how the general form of a linear equation[14] is used in different contexts, both in academic settings and everyday life. The general form $ax + b = 0$ is not only used in math exercises but also has broad applications in various fields, such as economics, engineering, and social sciences. For example, in economics, a linear equation can be used to calculate total spending based on the number of items purchased and the price per unit. In engineering, it can represent the relationship between voltage and electric current in a simple circuit. Even in daily life, a linear equation can help us estimate travel time based on speed and distance. To understand and solve linear equations, there is a systematic process we can follow to find the value of the unknown variable. This process is known as solving a linear equation, and its objective is to determine the variable's value that balances both sides of the equation. The process involves:

1. Simplify each side of the equation, if necessary, by performing operations like multiplication, division, or removing parentheses. This helps make the equation easier to work with.
2. Move terms to the appropriate side, usually by adding or subtracting terms on both sides so that variables are grouped on one side and constants on the other.
3. Divide or multiply both sides, if needed, to isolate the variable and obtain its final value.

For example, consider the following equation:

To solve the equation $4y + 8 = 0$:

1. Isolate the variable term:

Deduct 8 from both sides of the equation:

$$\rightarrow 4y + 8 - 8 = 0 - 8 \rightarrow 4y = -8$$

2. Solve for y :

Divide both sides by 4 to isolate y :

$$\rightarrow 4y = -8$$

The solution satisfying the equation is $y = -2$.

By following these systematic steps, we can solve various forms of one-variable linear equations more easily and in a structured way. Understanding how linear equations[15] work and how to solve them not only helps with solving math problems but also develops logical, analytical, and problem-solving skills that are useful in real-life situations.

2.2 The Matrix Laboratory

Software MATLAB (Matrix Laboratory) is an advanced programming platform and mathematical computation suite created by MathWorks.

Since its initial release, MATLAB has become one of the most popular and reliable tools in the fields of science and technology, particularly for performing numerical data analysis, mathematical modeling, system simulation, and the development and implementation of algorithms. This platform is specifically designed to support professionals across various disciplines such as engineering, physics, mathematics, statistics, finance, and computer science. Its use has expanded into industry, academia, and research and development (R&D), making MATLAB a primary tool for data-driven decision-making and computational experimentation.

With its intuitive programming interface and syntax structure that closely resembles conventional mathematical notation, MATLAB makes it easier for users from diverse backgrounds, including those not specialized in engineering or computer science to learn and use the language effectively. Its concise and expressive syntax allows users to complete complex tasks with relatively few lines of code. This makes MATLAB highly suitable for education, prototyping, and rapid, efficient development of technical systems.

Central to MATLAB's utility is its proficiency in handling matrix operations for efficiently handling and manipulating matrix-based data structures. Almost all variables in MATLAB are represented as matrices whether scalars, vectors, or multidimensional arrays. This capability is crucial because data in many engineering and scientific applications is often organized in tables or two-dimensional arrays. Mathematical operations such as matrix multiplication, inversion, transposition, and decomposition can be performed with simple, internally optimized syntax, making it faster and more accurate than many general-purpose programming languages.

Moreover, MATLAB delivers a comprehensive suite of integrated operations and dedicated modules for multifaceted data analytics and processing needs, including signal processing, digital image processing, domains span automated regulation, adaptive algorithms, hierarchical neural networks, and massive-scale data mining. These features allow users to focus on analytical or experimental aspects without having to build algorithms from scratch, significantly shortening system development time and reducing the risk of manual computational errors.

MATLAB also offers excellent data visualization capabilities, enabling users to present data and analysis results in the form of 2D and 3D plots, interactive diagrams, and even animations. This greatly aids in interpreting results, validating models, and effectively communicating findings to colleagues or within scientific forums. With support for creating graphical user interfaces (GUIs), MATLAB also allows for the development of interactive applications that others can use without direct access to the source code.

Furthermore, MATLAB integrates well with various platforms and systems, such as Python, C/C++, and Java programming languages, as well as hardware systems like Arduino,

Raspberry Pi, and other signal processing devices. This flexibility makes MATLAB an effective bridge between software development and hardware implementation in a wide range of technical and scientific projects.

Overall, MATLAB is not just a programming language but an integrated environment for exploring, modeling, and solving numerical computing problems. Due to its dependable performance in matrix computations, intuitive syntax, and powerful tools for visualization and data analysis, it serves as a vital resource across various fields, including academia, scientific research, and industrial applications.

Using MATLAB can accelerate innovation, reduce development time, and enhance the quality of data-based analysis and decision-making.

Below are some key aspects and standout features of MATLAB:

1. Numerical Computing Environment

MATLAB is one of the most powerful and comprehensive numerical computing environments available today. It is specifically designed to handle large-scale, high-precision numerical computations, making it highly reliable for various analytical and mathematical modeling processes. Its capability to solve complex computational problems involving thousands or even millions of data elements makes it ideal for many engineering and scientific fields.

MATLAB users can easily perform quantitative data analysis, develop numerical models of physical phenomena, and solve ordinary and partial differential equations (ODEs and PDEs), commonly found in mechanical, thermal, fluid, and electromagnetic systems. MATLAB is also highly effective for dynamic system simulations, such as feedback control systems, vibration systems, or power grids, where complex mathematical models must be simulated accurately and efficiently in both time and frequency domains.

In control engineering, for example, MATLAB is used to design, simulate, and evaluate the performance of controllers such as PID, state-space, and optimal control. Engineers can test system models virtually before real-world implementation, saving both time and costs in the design and development process.

MATLAB also plays an important role in economic and financial modeling, where market systems or consumer behavior are often represented through complex and dynamic mathematical equations. With MATLAB's ability to solve nonlinear equation systems, analyze parameter sensitivity, and simulate various scenarios, economists and financial analysts can make more accurate forecasts and plans.

Another advantage of MATLAB as a numerical computing environment is its support for high-precision floating-point arithmetic and symbolic computing (through integration with the Symbolic Math Toolbox). This provides users with the flexibility to conduct deeper mathematical explorations and validate numerical results using analytical methods.

As such, MATLAB not only offers speed and precision as a numerical computing environment but also provides a unified framework that facilitates scientific exploration, system design, and model-based decision-making more efficiently and effectively.

2. High-Level Programming Language

MATLAB is more than just a numerical computing environment; it is also a high-level programming language designed with readable, expressive syntax that closely resembles mathematical notation. As a result, it is ideally positioned for a broad spectrum of users,

including students and experts in engineering and science, who seek streamlined development of technical and data-driven software applications. MATLAB enables the use of up-to-date programming approaches, including procedural, functional, and OOP methodologies.

With these paradigms, users can build more organized, modular, and manageable program structures. For example, with an object-oriented approach, users can create classes and objects to represent system components or program logic, making application development more flexible and problem-oriented.

Users can also create custom functions and scripts for reusable tasks, which is particularly useful for projects involving automated processes, parameter testing, or repetitive data analysis. The readable and consistent structure of MATLAB code promotes effective team collaboration.

Another standout feature is the ability to develop graphical user interfaces (GUIs) easily through App Designer or GUIDE. With this, users can design interactive application interfaces without writing extensive code, enabling the creation of standalone desktop applications that others can use without needing to understand the underlying algorithms.

Moreover, MATLAB provides debugging and code performance analysis tools that are essential for development. Features like breakpoints, profilers, and live scripts help users identify errors, evaluate algorithm efficiency, and document analyses dynamically. Thus, MATLAB is not only a computation tool but a complete platform for developing software solutions that can be integrated into industrial or academic systems.

Furthermore, MATLAB supports code compilation into executables (.exe) and integration with other languages like Python, C/C++, and Java, allowing cross-platform interoperability and implementation. This offers developers great opportunities to turn prototypes into real applications deployable across various operational systems.

In short, by providing a strong and flexible high-level programming environment, MATLAB enables users not only to solve numerical problems but also to design, build, and distribute functional and efficient technical applications across multiple sectors.

3. Matrix Management and Operations

One of MATLAB's most fundamental and distinguishing strengths is its matrix management and operation capabilities. True to its name, MATrix LABoratory, MATLAB is built on the premise that almost all data, whether scalars, vectors, 2D arrays, or multidimensional arrays are treated as matrices. This approach offers remarkable advantages in terms of simplicity, efficiency, and speed when handling complex mathematical operations.

With this matrix-based structure, users can perform various linear algebra operations directly, such as multiplication, inversion, determinant calculation, transposition, and LU, QR, and SVD decomposition, as well as eigenvalue-eigenvector analysis, often with just one or two lines of code. These operations are critical in many technical and scientific applications such as dynamic system analysis, signal and image processing, machine learning, and structural mechanics computations.

For instance, in modeling mechanical or electrical systems, large linear equation systems often represent the interrelationships of system components. MATLAB can solve these systems quickly using optimized solvers, making it very useful for real-time simulations and parameter-heavy scenario analysis.

MATLAB also provides special matrix operators and functions optimized for large-scale array processing. Operators such as `.*`, `./`, and `.^` allow element-wise operations without the need for explicit loops, resulting in more concise and efficient code.

Its matrix manipulation capabilities also include flexible indexing and slicing, enabling users to easily access, modify, and extract specific parts of matrices. This is particularly useful in big data applications where memory efficiency and data access speed are critical.

In relation to machine learning and AI, where most data is processed in the form of matrices or tensors, MATLAB excels with strong integration and also Utilizing resources such as the Statistics and Machine Learning Toolbox and the Deep Learning Toolbox, which leverage matrix operations for model training, classification, and regression. MATLAB's streamlined and accessible matrix operations position it as a preferred solution for tackling advanced numerical tasks. This strength not only accelerates computations but also allows users to focus on the core logic and concepts of the problems without being burdened by technical implementation details.

4. Data Visualization

MATLAB offers numerous flexible and user-friendly tools for visualizing data interactively. Including both straightforward 2D plotting and intricate 3D rendering with animations, MATLAB enables users to represent data and analysis results visually, making them easier to understand and present whether in technical reports or scientific presentations.

5. Applications in Industry and Academia

MATLAB is widely used beyond academia. In industry, it is used for system design and analysis, signal and image processing, control system development, and financial and risk modeling. In research, MATLAB is often used for numerical experiments, algorithm prototyping, and data exploration in scientific projects.

6. User Community and Support

MATLAB has an active global user community, complete with discussion forums, file exchanges, and comprehensive official documentation. This enables both new and experienced users to easily find solutions, inspiration, and learning resources to deepen their expertise.

7. Toolboxes (Add-Ons)

A major strength of MATLAB is its vast library of toolboxes, specialized add-on packages designed for specific needs like the Signal Processing Toolbox or the Image Processing Toolbox, Deep Learning Toolbox, Financial Toolbox, and many more. These toolboxes provide ready-to-use functions that accelerate development and reduce the workload of building systems from scratch.

MATLAB is not just a programming language; it is a comprehensive technical computing ecosystem. Its flexibility, ease of use, and broad capabilities make it an invaluable tool for accelerating innovation, research, and development across various fields of science and technology. For professionals and researchers alike, mastering MATLAB can be a crucial asset in addressing modern computational challenges.

3. Proposed Method

In this study, the proposed method focuses on the use of MATLAB as an interactive learning medium to understand and solve systems of simultaneous linear equations. This method aims to improve both the theoretical grasp of linear algebra principles and the hands-on abilities needed to tackle mathematical problems through numerical and visual techniques.

The proposed method consists of the following steps:

1. Introduction to Theoretical Concepts

Students are first introduced to the fundamental theories of linear equation systems, including solution methods such as Gaussian elimination, substitution, and matrix methods (inversion and Gauss-Jordan elimination). The goal of this stage is to provide a conceptual foundation before moving on to computational implementation.

2. Interactive Simulation Using MATLAB

After understanding the theoretical concepts, students are given MATLAB-based practical modules designed to explore various solution methods for linear equation systems. These modules include the use of basic commands such as `inv()`, `rref()`, and the matrix operator `\` to solve systems in the form of $Ax = b$.

3. Visualization of Process and Solutions

MATLAB is used to visualize the solutions in the form of two-dimensional or three-dimensional graphs (if the system involves only two or three variables). This aims to help students better understand the geometric interpretation of linear systems, such as the point of intersection of lines or planes as the solution.

4. Application to Real-World Case Studies

To improve the relevance of the learning process, this method also includes exercises where students apply systems of linear equations to real-world contexts, such as electrical circuit analysis, simple economic models, or resource distribution problems. Students are tasked with constructing the mathematical model of the problem and then solving it using MATLAB.

5. Evaluation and Learning Reflection

Evaluation is carried out through practical quizzes, individual reports, and learning reflections. Additionally, automated feedback from MATLAB scripts is provided to indicate whether the submitted solutions are correct and in accordance with the chosen method.

Advantages of the Proposed Method

1. Interactive and Practical: Utilizes MATLAB's visualization and interactive features to enhance student engagement and understanding.
2. Problem-Based Approach: Encourages students to relate abstract concepts to real-world applications.
3. Strengthening Digital Competency: Students not only learn mathematics but also gain technical skills in MATLAB programming.

With this approach, the learning process for solving systems of linear equations is expected to be not only theoretical but also practical and contextual, making it more effective in building conceptual understanding and students' problem-solving abilities.

4. Results and Discussion

We will use MATLAB in this session to solve a linear equations system (SLE) using the inverse of a matrix. The equations to be solved are as follows.

$$\text{Equation 1 : } 10x_1 - x_2 + 2x_3 + 0x_4 = 6$$

$$\text{Equation 2 : } -x_1 + 11x_2 - x_3 + 3x_4 = 25$$

$$\text{Equation 3 : } 2x_1 - x_2 + 10x_3 - x_4 = -11$$

$$\text{Equation 4 : } 0x_1 + 3x_2 - x_3 + 8x_4 = 15$$

Step-by-Step Solution:

1. Matrix Representation

Represent the SLE as $C \cdot Y = E$:

C = Coefficient matrix (4×4)

Y = Variable vector [y_1 ; y_2 ; y_3 ; y_4]

E = Constant vector

From the equations:

$$A = [10, -1, 2, 0; -1, 11, -1, 3; 2, -1, 10, -1; 0, 3, -1, 8];$$

$$B = [6; 25; -11; 15];$$

Solution via Matrix Inverse

Compute the solution using:

$$X = \text{inv}(A) \times B$$

% Define matrix C and vector E

$$C = [10, -1, 2, 0; -1, 11, -1, 3; 2, -1, 10, -1; 0, 3, -1, 8];$$

$$E = [6; 25; -11; 15];$$

% Calculate inverse and solution

$$C_inv = \text{inv}(C);$$

$$Y = C_inv * E;$$

% Display results

$$\text{disp('Solution:');}$$

$$\text{fprintf('y}_1 = \%0.4f \backslash n', Y(1));$$

$$\text{fprintf('y}_2 = \%0.4f \backslash n', Y(2));$$

$$\text{fprintf('y}_3 = \%0.4f \backslash n', Y(3));$$

$$\text{fprintf('y}_4 = \%0.4f \backslash n', Y(4));$$

Output:

The answer:

$$y_1 = 1.0000$$

$$y_2 = 2.0000$$

$$y_3 = -1.0000$$

$$y_4 = 1.0000$$

Verification by Substitution:

$$1. \text{Equation 1: } 10(1) - (2) + 2(-1) + 0(1) = 10 - 2 - 2 = 6 \rightarrow \text{Valid!}$$

$$2. \text{Equation 2: } -(1) + 11(2) - (-1) + 3(1) = -1 + 22 + 1 + 3 = 25 \rightarrow \text{Valid!}$$

$$3. \text{Equation 3: } 2(1) - (2) + 10(-1) - (1) = 2 - 2 - 10 - 1 = -11 \rightarrow \text{Valid!}$$

$$4. \text{Equation 4: } 0(1) + 3(2) - (-1) + 8(1) = 0 + 6 + 1 + 8 = 15 \rightarrow \text{Valid!}$$

Important Notes:

1. Numerical Efficiency:

While $X = \text{inv}(A)*B$ works for small matrices, the backslash operator $X = A \setminus B$ is preferred:

Better numerical stability

Higher computational efficiency (uses LU/QR decomposition)

% Recommended alternative

$X = A \setminus B$; *% Solves $AX=B$ without explicit inversion*

1. Invertibility Conditions:

The inverse exists only if:

A is square ($n \times n$)

$\det(A) \neq 0$ (non-singular matrix)

MATLAB Warnings:

If A is ill-conditioned (nearly singular), MATLAB issues:

Warning: Numerical instability detected: matrix is almost singular or badly scaled.

2. Shortcomings of the Technique

Matrix inverse is computationally inefficient for large matrices ($>100 \times 100$).

For large/sparse systems, use:

Iterative methods (Jacobi, Gauss-Seidel)

Matrix decomposition (LU, Cholesky)

It can be concluded that the solution $X = [1; 2; -1; 1]$ is verified and consistent. This demonstrates a valid application of the matrix inverse method for small SLEs in MATLAB, while emphasizing best practices for numerical computation. For larger systems, always prefer $A \setminus B$ over explicit inversion.

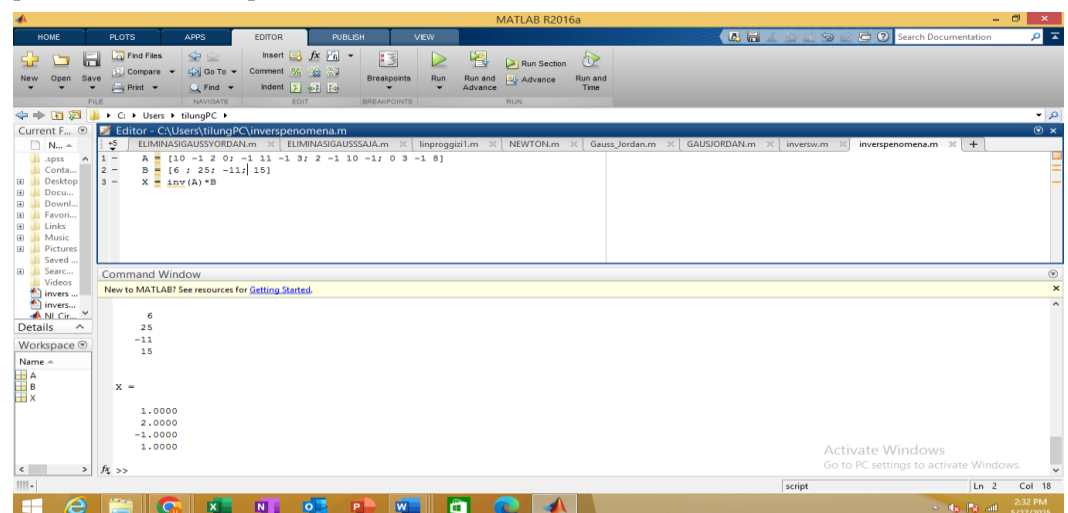


Figure 1. The following is an image of the results of matrix inversion based on Matlab.

6. Conclusions

The conclusion of the title "MATLAB-Based Learning of Simultaneous Linear Equation Systems" indicates that the use of MATLAB as a learning tool has great potential in helping individuals understand and solve systems of linear equations simultaneously. MATLAB, with its various advanced features and data processing capabilities, allows users to analyze and visualize complex mathematical problems in a more intuitive way. Thus, users can not only solve equations but also gain deeper insights into the relationships between variables within the system. Additionally, this software supports interactive simulations and experiments, which can enrich the learning experience and encourage further exploration. With this approach, it is hoped that students can enhance their understanding and skills in mathematics, as well as prepare themselves for greater challenges in the future. The use of MATLAB in

this educational context also reflects the importance of technology in modern learning processes, where interactivity and visualization are key to understanding complex concepts.

References

- [1] X. Iqboljon, "System of linear algebraic equations and methods of their solution," *Multidiscip. J. Sci. Technol.*, vol. 4, no. 1, pp. 39–44, 2024.
- [2] N. P. Dongoran, S. A. Wardani, A. C. S. Pane, V. A. B. Sihombing, and V. C. O. Siregar, "PREDIKSI ANGKA PERNIKAHAN DI SUMATERA BARAT 5 TAHUN MENDATANG MENGGUNAKAN IMPLEMENTASI MATLAB DENGAN DERET TAYLOR," *J. Inov. Pendidik.*, vol. 7, no. 2, 2025.
- [3] A. Hysa, M. M. Mahmoud, and A. Ewais, "An Investigation of the Output Characteristics of Photovoltaic Cells Using Iterative Techniques and MATLAB®2024a Software," *Control Syst. Optim. Lett.*, vol. 3, no. 1, pp. 46–52, 2025.
- [4] C. H. Pratomo, W. Kurniawan, and M. A. Kholik, "PENGOLAHAN CITRA DIGITAL DETEKSI NOMOR PLAT KENDARAAN BERBASIS GUI MATLAB," *J. Sci. Res. Dev.*, vol. 7, no. 1, pp. 204–212, 2025.
- [5] S. M. R. H. Moosavi, M. A. F. Araghi, and S. Ziari, "An algorithm for solving a system of linear equations with Z-numbers based on the neural network approach," *J. Intell. & Fuzzy Syst.*, vol. 46, no. 1, pp. 309–320, 2024.
- [6] S. Singh and G. Panda, "Bounding the Solution Set of Overdetermined System of Interval Linear Equations," *Bull. Iran. Math. Soc.*, vol. 51, no. 2, p. 23, 2025.
- [7] A. Wibowo, A. SYAEKHU, and others, "Analysis of Students' Errors in Solving Mathematics Story Problems on the System of Linear Equations with Two Variables," *J. Digit. Learn. DISTANCE Educ.*, vol. 2, no. 12, pp. 872–878, 2024.
- [8] M. ur Rahman, M. Sun, S. Boulaaras, and D. Baleanu, "Bifurcations, chaotic behavior, sensitivity analysis, and various soliton solutions for the extended nonlinear Schrödinger equation," *Bound. Value Probl.*, vol. 2024, no. 1, p. 15, 2024.
- [9] Y. Mulian, "On the Exact Solution for the Schrödinger Equation. *Particles* 2024, 7, 1095–1119," 2024.
- [10] A. Patterson and L. Lapworth, "Measurement schemes for quantum linear equation solvers," *Quantum Sci. Technol.*, vol. 10, no. 2, p. 25037, 2025.
- [11] H. R. Maharani, M. A. Basir, and N. L. Ningrum, "Creative Thinking in Prospective Mathematics Teachers: Flexibility in Solving Systems of Linear Equation," *J. Pendidik. Progresif*, vol. 14, no. 2, pp. 810–820, 2024.
- [12] B. Wang and J. Li, "Discrete-time distributed algorithms for solving linear equations via layered coordination," *Int. J. Syst. Sci.*, vol. 56, no. 3, pp. 638–657, 2025.
- [13] J. Liu and Z. Li, "Bifurcation analysis and soliton solutions to the Kuralay equation via dynamic system analysis method and complete discrimination system method," *Qual. Theory Dyn. Syst.*, vol. 23, no. 3, p. 126, 2024.
- [14] W. A. Faridi, A.-M. Wazwaz, A. M. Mostafa, R. Myrzakulov, and Z. Umurzakhova, "The Lie point symmetry criteria and formation of exact analytical solutions for Kairat-II equation: Paul-Painlevé approach," *Chaos, Solitons & Fractals*, vol. 182, p. 114745, 2024.
- [15] O. D. Adeyemo and C. M. Khalique, "Application of conserved quantities using the formal Lagrangian of a nonlinear integro partial differential equation through optimal system of one-dimensional subalgebras in physics and engineering," *Open Phys.*, vol. 22, no. 1, p. 20230155, 2024.