Journal of Artificial Intelligence and Engineering Applications

Website: https://ioinformatic.org/

15th February 2026. Vol. 5. No. 2; e-ISSN: 2808-4519

Utilization of MATLAB as a Learning Medium in Basic Mathematical Computation Techniques

Ghufron Makmun Lbs

Computer Science, State Islamic University of North Sumatra
ghufronlubis1@gmail.com

Abstract

Mathematical computational learning is an important component in mastering numerical and programming concepts. MATLAB as a numerical computation software provides various facilities to support the learning process, especially in basic computational engineering materials such as finite iteration, conditioned iteration, and logic processing. This study aims to analyze the use of MATLAB as a medium for learning basic mathematical computational techniques through a series of practicums that include the use of if-else control structures, for loops, while loops, and switch cases. The method used is a practicum experiment by running several MATLAB programs to solve mathematical problems, such as calculating final values, factorials, finite iterations, and calculating the volume of geometric shapes. The practicum results show that MATLAB is able to visualize the computational process effectively, provide direct feedback, and strengthen students' understanding of mathematical logic and algorithm concepts. Thus, MATLAB can be an efficient learning medium in improving students' basic computational skills in the field of numerical methods.

Keywords: *MATLAB, Mathematical Computing, Learning Media, Limited Iteration, Conditioned Iteration*

1. Introduction

The development of modern computing technology has pushed the importance of the ability to perform mathematical calculations quickly, precisely, and efficiently. In various fields of science, especially science and engineering, mathematical computation is the basis for solving complex numerical problems. One of the software widely used in academia and industry to support this process is MATLAB. MATLAB offers a very powerful and interactive numerical computing environment, with a high-level programming language oriented to matrices as well as data visualization capabilities and practical implementation of mathematical algorithms. MATLAB is a programming platform that enables complex simulations and calculations [1][2]. In addition, MATLAB is also used for multivariate analysis and interactive visualization of multivariable functions and can emphasize the efficiency of matrix-based computations, because MATLAB is designed for matrix operations. Thus, MATLAB not only simplifies the computational process, but also increases the effectiveness of interactive learning of mathematical concepts [3].

In learning numerical methods, understanding basic computational techniques such as branching (if-else), looping (for and while), and condition selection (switch case) is the foundation for designing and implementing algorithms [4].However, learning these concepts is often less effective if only presented theoretically without being accompanied by direct simulation. Therefore, using MATLAB as a learning medium is the right solution because it is able to visualize the computational process while providing instant feedback on the program being run [5].

The “Basic Mathematical Computation Techniques” practicum aims to introduce students to various control structures in MATLAB and how these structures are applied to solve mathematical problems, such as calculating student final grades, performing iterations, determining factorials, and calculating the volume of geometric shapes. This practicum is designed to provide an interactive learning experience, so that students not only understand the theory, but are also able to apply simple algorithms in solving mathematical problems. Through practicum activities, students are expected to develop logical and analytical thinking skills computationally, optimize the use of loops, and understand the application of control structures such as branching, looping, and conditional selection. This study aims to illustrate the extent to which MATLAB can be used as an effective learning medium in helping students understand the basic concepts of mathematical computation through a practical and interactive approach.

Through this activity, students are expected to not only be able to understand the basic concepts of programming and computational techniques, but also be able to apply control structures to solve various mathematical problems effectively and efficiently. Thus, this research also provides an overview of the effectiveness of MATLAB as a learning medium that combines theory and practice, as well as its contribution to improving students' computational skills in the modern technological era.

2. Literature

2.1. Matlab

MATLAB (Matrix Laboratory) is a program for numerical analysis and computation as well as an advanced mathematical programming language developed with a matrix-based mindset. As a commercial product of MathWorks, Inc., MATLAB is very useful for processing linear algebra and various other mathematical calculations. This software is equipped with various built-in functions that facilitate tasks that are usually difficult to do manually, especially matrix-based numerical calculations [2].MATLAB is often used to solve various problems involving mathematical operations of elements, matrices, optimization, approximation, and others. Its areas of use include mathematics and computing, algorithm development, modeling programming, simulation and prototyping, data analysis, exploration and visualization, numerical and statistical analysis, and the development of engineering applications.

Matlab is also frequently used in mathematical computing because it has functions for analyzing data. Matlab is used to analyze mathematical data in the form of matrices and numbers in advanced mathematics. Linear programs involving complex calculations can be solved with Matlab. In addition to completing calculations, modeling, and data analysis, Matlab applications can also display graphical images of the results of calculations in Matlab [6].

In addition, MATLAB can also be used for modeling, simulation, graphing, and computational development. At first glance, MATLAB may seem like a calculator for engineering calculations and plotting, but in reality, this software is much more sophisticated. MATLAB can facilitate numerical computation, data analysis, system simulation in engineering, and widespread code distribution and use [7].

2.2. Looping

Looping is a programming language and algorithm that is used to repeat a command/instruction made in a script according to a predetermined number of times. Looping aims to shorten the time of program statements that must be written in large numbers [8].Looping is a form that is often found in an application program. Looping the process, will carry out the process repeatedly as long as the condition is still True (or where the desired limit value has not been reached, and the condition will stop if the condition changes to False (or the limit value has been reached). In programming, there are three types of loops, namely using the For, While, and Do-While statements [9].

1) A for loop is a type of loop used when the number of repetitions is known or can be determined in advance. This structure consists of three main parts: initialization, condition, and increment/decrement. At each iteration, the condition is checked first; if it is true, the block of commands is executed. The for loop is very efficient for repetitive and structured processes, such as array iterations or repeated calculations with a specific limit.
2) The while loop is an entry-controlled loop, meaning the condition is checked before the loop block is executed. If the condition is true, the instructions within the loop are executed. The while loop is used when the number of iterations is unknown at the outset and depends on a condition that changes during the process, such as reading data until a certain condition is met.
3) The do-while loop is an exit-controlled loop because the condition is checked after the commands inside the loop are executed. Therefore, the code block inside the do-while loop is always executed at least once, even if the loop condition is false from the start. This loop is suitable for use in situations that require initial execution before checking the condition, such as repeating menu displays.

2.3. Conditional Statements

Conditional statements (decision-making statements) is a fundamental programming concept that determines the flow of program execution based on certain conditions. This structure allows programs to make logical decisions, allowing the instructions executed to adapt to the circumstances or user input. In many programming languages, branching works by evaluating expressions to true or false, thus relying heavily on the logical relationships and comparison operators used in the syntax [10].

Basically, Conditional statements consists of several common forms, such as if, if–else, else-if (ladder), nested if, and switch-case. Each has different characteristics and purposes. The if–else statement, for example, is used when a condition has only two possible outcomes, while the else-if ladder is needed when there are more than two conditions that need to be tested sequentially. For cases of testing values sourced from a single variable with many fixed choices, the switch-case structure is more efficient to use. The existence of these structures makes the program flow more systematic, predictable, and easier to manage, especially in large-scale applications that require complex logic control [10].

In the context of programming education, a common error is the use of sequential if statements for conditions that are actually mutually exclusive. In such situations, using exclusive-if without else-if can be confusing, reduce readability, and increase the risk of logical errors. Nurollahian's study shows that this anti-pattern frequently occurs in elementary and intermediate students and is often related to a lack of understanding of proper control structure design. The use of else-if is recommended to make the exclusivity of the condition clearer and the programmer's intent more easily understood [11].

Conditional statements also plays a crucial role in advanced applications, not just in educational contexts. In the development of artificial intelligence-based systems, robotics, and autonomous programming, branching is at the heart of the system's reactive behavior. Liang demonstrated that a language model (LLM) that generates policy code uses structures such as if-else and while to organize the robot's decision flow based on sensory input. This demonstrates that branching remains a fundamental element even in modern computing involving machine learning and autonomous control [12].

2.4. Mathematical Computing

Computational mathematics is generally associated with the use of computational thinking in mathematics learning activities, while computational mathematics refers to the scientific study of numerical methods and algorithms to solve complex mathematical problems computationally. Computational thinking (CT) is an integral part of the mathematical thinking process. There are four main cognitive activities that connect computational thinking to mathematical problem solving: the process of translating a situation into a computational or mathematical representation, performing mathematical and computational reasoning, translating the results back to the original context, and evaluating the appropriateness of the solution. These activities emphasize that CT plays a role not only as a technical addition, but also as a foundation in the modern mathematical thinking process, especially when problems are solved through computational models [13].

Computing integrated into mathematics learning can help students develop algorithmic skills, abstraction, decomposition, and logical thinking patterns that support a deeper understanding of mathematical concepts. The study showed that various strategies such as the use of robotics, programming, visual language, and computational logic activities were effective in improving students' computational abilities. Thus, computational mathematics has become an important approach in modern mathematics education, where technology and computational modeling are used to enrich mathematical understanding [14].

Computational mathematics is the fundamental and applied study of numerical methods, algorithms, and simulations used in solving mathematical and physical problems. This field encompasses research on the stability, accuracy, convergence, and efficiency of algorithms used to model real-world phenomena, ranging from differential equations and fluid dynamics to scientific data processing. Thus, computational mathematics serves as a scientific framework that provides performative methods for various modern scientific computing applications [15].

3. Research methods

This study uses a practical experiment method to analyze the use of MATLAB as a learning medium for basic mathematical computation techniques. The practicum is carried out by running a number of MATLAB programs designed to implement basic control structures, namely branching (if-else), looping (for and while), and selecting conditions using switch case. Each program is run to solve a specific mathematical problem, so that students can understand the relationship between the program logic and the resulting computational results. The practicum is carried out through the following stages:

1) Conditional Relative Value (If - Else). Students create a program to calculate final course grades based on the weighted scores for quizzes, midterm exams, and final exams. The program uses an if–else control structure to determine the grade. This process trains students in understanding branching logic.
2) Limited Iteration (for loop). The program being run displays an example of a loop with a predetermined number of iterations. Students learn how the for loop structure is used when the number of iterations is known.
3) Conditioned Iteration (While Loop). Students run two programs: a factorial calculation program using a for loop and a while loop program to gradually reduce values until a stopping condition is reached. These steps help students understand how loops operate based on specific conditions.
4) Absolute Value Conditions and Switch Case. Students create programs to calculate the volume of various geometric shapes (boxes, cylinders, and cones) using switch case. This control structure helps students understand the selection of conditions involving multiple options.

4. Results and Discussion

The MATLAB practical work on basic mathematical computation techniques produces several program outputs that demonstrate how basic control structures can be used to solve various mathematical problems. Each experiment provides a concrete illustration of the programming logic flow, the computational process, and the effectiveness of MATLAB as a learning medium. In addition, this practical work helps students understand how structured calculations work through the application of formulas and decision-making using if-else conditions. Through these programming exercises, students can also improve their analytical skills and understanding of algorithm concepts, thus being able to connect mathematical theory with the implementation of programs that run systematically.

4.1. Relative Value Conditional (If–Else)

This program is designed to calculate a student's final grade based on three assessment components: quiz scores, mid-term exam scores, and final exam scores. The program first displays the program's identity information and then prompts the user to enter the three scores. After all scores are entered, the program calculates the final grade using specific weights for each component. Next, the program determines the student's grade using an if-elseif-else conditional structure. Finally, the program displays the final score along with the grade obtained. The calculation of the final value is based on the weight of each component, namely: quiz scores 20%, mid-term exam scores 40%, final exam scores 40%, the final value formula is:

$$\text{NA} = \frac{20\% \times U1 + 40\% \times U2 + 40\% \times U3}{100} \quad (1)$$

U1 = quiz scores
U2 = mid-term exam scores
U3 = final exam scores

Determination of grade is done based on the final score obtained by the student according to the following conditions:

$\texttt{Grade A : NA} \geq 80$

Grade B : NA ≥ 70
Grade C : NA ≥ 60
Grade D : NA ≥ 50
Grade E : NA < 50

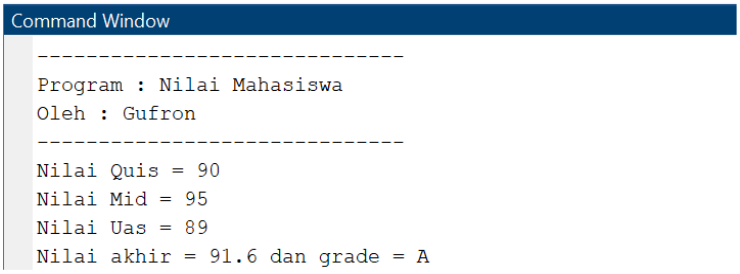

Fig. 1: Program results using if else

This implementation demonstrates that MATLAB branching accurately processes logical conditions, and students can understand how decisions are made within specific boundaries. The module demonstrates that the program runs correctly and produces grades within the specified range.

4.2. Limited Iteration (For Loop)

The next experiment applies the for loop structure to generate a specific number of iterations. The use of the for loop demonstrates how MATLAB executes repeated instructions when the number of iterations is known in advance. In this experiment, a program is created to calculate the factorial of a number using a structured iteration process. The program asks for input in the form of n, then uses a for loop from 1 to n–1 to perform repeated multiplications to produce the value n!. The output, as shown in the module, shows a sequence of iterations working systematically until the correct factorial is obtained. This experiment helps students understand the concept of finite iteration and demonstrates how the for loop can be used to solve mathematical problems that require repeated operations. Thus, the for loop command is one of the most basic and important control structures in programming because it allows for the efficient and predictable execution of repeated instructions.

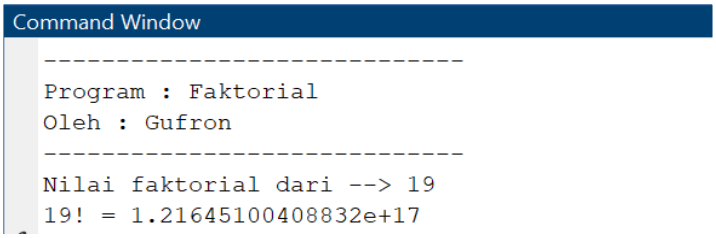

Fig. 2: Program results using for loop

4.3. Conditioned Iteration (While Loop)

In the next experiment, a while loop is used to calculate how many times a tank volume (v1) can be reduced using the bucket volume (v2) until it reaches zero or less. Unlike a for loop, which has a fixed iteration limit, a while loop operates based on a specific condition; the program will continue iterating as long as the condition is met. In this program, the loop condition is v1 > 0, so the reduction process will continue until the tank volume no longer meets that condition. Each iteration step displays the change in the tank volume after being reduced by the bucket volume, while also incrementing the iteration counter (n). The program output displays all the reduction steps as well as the final iteration count, which is the number of times the bucket must be used to empty the tank. This experiment demonstrates how the while loop structure is used in situations where the number of iterations is unknown in advance, making it very useful for dynamic condition-based computations.

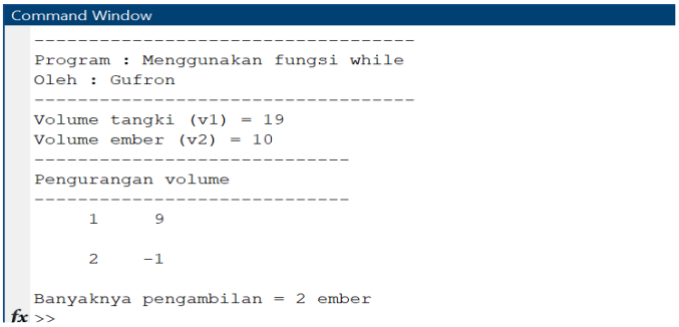

Fig. 3: Program results using while loop

4.4. Control Structure (Switch Case)

The final experiment implements a switch–case control structure to select one of several volume calculation formulas based on a user's choice: the volume of a box, a cylinder, or a cone. The program prompts the user to select one of the options and then executes the code block corresponding to that choice. For each case, the program prompts for the required geometric parameters (e.g., length, width, height, or radius), then calculates the volume using the relevant mathematical formula and displays the result. The switch–case approach makes the code cleaner and more readable when there are many mutually exclusive conditional branches than using numerous if–elseif statements. Furthermore, this experiment trains students to map user choices to appropriate computational actions and emphasizes the importance of input validation (e.g., ensuring positive dimension values) for meaningful calculation results. Thus, this lab demonstrates the usefulness of switch–case in situations with many discrete options and facilitates the systematic application of selective control structures in MATLAB programming.

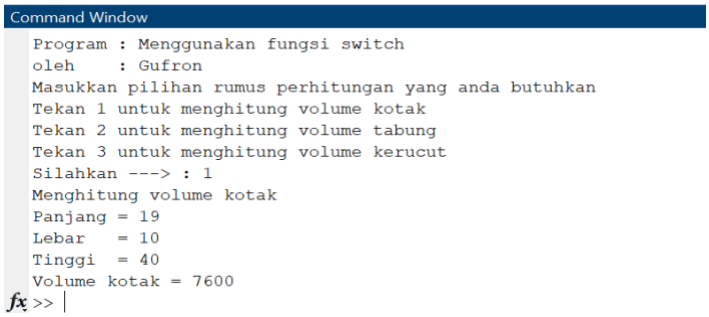

Fig. 4: Program results using switch case

5. Conclusion

Based on the results of the basic mathematical computational techniques practicum using MATLAB, it can be concluded that MATLAB is an effective learning medium because it is able to visualize the computational process, provide instant feedback, and make it easier for students to understand the concepts of mathematical logic and algorithms. Various basic control structures such as if-else, for, while, and switch case can be implemented well to solve mathematical problems, ranging from value calculations, limited iterations, factorials, to volume calculations of geometric shapes. This computation-based practicum also improves students abilities in designing simple algorithms, understanding program logic flows, and familiarizing themselves with numerical computation processes systematically. Thus, the use of MATLAB can be an alternative learning method that is more interactive and practical compared to conventional methods.

References

[1] A. N. Nasution and Yahfizham, "Systematic Literatur Review : Software Matematika MatLab Sebagai Media Belajar untuk Meningkatkan Kemampuan Komputasi Siswa," *Merkurius : Jurnal Riset Sistem Informasi dan Teknik Informatika*, vol. 2, no. 4, pp. 20–27, 2024, doi: 10.61132/merkurius.v2i3.113.

[2] M. Fatwa, R. Ristu, S. Pandiangan, and E. Supriyadi, "PENGAPLIKASIAN MATLAB PADA PERHITUNGAN MATRIKS," *Papanda Journal of Mathematics and Sciences Research*, vol. 1, no. 2, pp. 81–93, Dec. 2022.

[3] N. P. Dongoran, A. C. S. Pane, and S. A. Wardani, "Pemanfaatan MATLAB dalam Analisis Turunan Parsial : Visualisasi dan Implementasi Fungsi Multivariat," *Jurnal Pengabdian Masyarakat Sains dan Teknologi*, vol. 3, no. 4, pp. 92–97, Dec. 2024, doi: 10.58169/jpmsaintek.v3i4.638.

[4] T. Limbong and Sriadhi, *Pemrograman Web Dasar*. Medan: Yayasan Kita Menulis, 2021.

[5] L. N. Azizah and K. Khalimaturofi'ah, "Penerapan MATLAB dalam Pembelajaran Analisis Numerik: Studi Kuantitatif pada Hasil Belajar Mahasiswa," *Journal of Instructional Technology*, vol. 6, no. 2, pp. 135–140, Oct. 2025, doi: 10.20527/j-instech.v6i2.17019.

[6] E. N. Ardina, Erlinasari, and Supari, "Pengenalan dan Implementasi Matlab Untuk Siswa SMK Walisongo Semarang," *Jurnal Abdimas PHB*, vol. 8, no. 1, pp. 226–232, 2025.

[7] A. M. Ilyas, *Belajar MATLAB dan Contoh Analisis Aliran Daya*. Ternate: PT. Kamiya Jaya Aquatic, 2024.

[8] D. Rahmat, S. A'zizah, and S. Mulyani, "PERKALIAN MENGGUNAKAN BAHASA PEMROGRAMAN PERULANGAN (LOOPING) BERBANTU SUBLIME TEXT DAN XAMPP," *Djtechno: Journal of Information Technology Research*, vol. 3, no. 2, pp. 2776–8546, 2022.

[9] I. A. Dianta, *Logika dan Algoritma Untuk Merancang Aplikasi Komputer*. Semarang: Yayasan Prima Agus Teknik, 2021.

[10] G. Yasmeen, "Implementation of Decision Making and Iterative Statements in C If-Then-Else, Switch, While, For, Do While," *Advanced Innovations in Computer Programming Languages*, vol. 2, no. 2, pp. 1–11, 2020.

[11] S. Nurollahian, M. Hooper, A. Salazar, and E. Wiese, "Use of an Anti-Pattern in CS2: Sequential if Statements with Exclusive Conditions," *SIGCSE 2023 - Proceedings of the 54th ACM Technical Symposium on Computer Science Education*, vol. 1, pp. 542–548, Mar. 2023, doi: 10.1145/3545945.3569744.

[12] J. Liang *et al.*, "Code as Policies: Language Model Programs for Embodied Control," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2023)*, London: IEEE, May 2023, pp. 1–8. doi: 10.1109/ICRA48891.2023.10160591.

[13] M. Kallia, S. P. van Borkulo, P. Drijvers, E. Barendsen, and J. Tolboom, "Characterising computational thinking in mathematics education: a literature-informed Delphi study," *Research in Mathematics Education*, vol. 23, no. 2, pp. 159–187, 2021, doi: 10.1080/14794802.2020.1852104.

[14] S. Subramaniam, S. M. Maat, and M. S. Mahmud, "Computational thinking in mathematics education: A systematic review," *Cypriot Journal of Educational Sciences*, vol. 17, no. 6, pp. 2029–2044, Jun. 2022, doi: 10.18844/cjes.v17i6.7494.

[15] H. A. Matevossian and F. dell'Isola, "'Computational Mathematics and Mathematical Physics'—Editorial I (2021–2023)," *Axioms*, vol. 12, no. 9, pp. 1–6, Sep. 2023, doi: 10.3390/axioms12090824.