IJLAI Transactions on Science and Engineering (ISSN: 3006-3418) 2025, vol. 3, no. 4, pp: 1–7

Research on the Teaching Mode of Numerical Analysis Course Based on Algorithmic Thinking

Junlin Li[1], Weiwei Sun[1], Keke Hu[1], Haifeng Li[1], and Jiayuan Yan[2]

[1] School of Mathematics and Statistics, Fuyang Normal University, Fuyang 236037 Anhui, China
[2] School of Artificial Intelligence, Henan University, Zhengzhou 450046 Henan, China
Received Sep. 13, 2025; Revised and Accepted Oct. 1, 2025

Abstract. Numerical analysis, as an important branch of mathematics, is widely applied in modern engineering, science and technology fields. In traditional teaching, numerical analysis focuses on imparting computational methods while neglecting the cultivation of algorithmic thinking. With the development of technology, especially the rise of artificial intelligence, the application of numerical analysis is becoming increasingly widespread. This requires students not only to master numerical methods but also to possess the ability to solve problems innovatively. For this purpose, the teaching mode of numerical analysis courses based on algorithmic thinking was explored. By integrating algorithmic thinking in teaching, students' ability to flexibly apply algorithms was cultivated, and their ability to solve complex problems was enhanced, laying a solid foundation for their future career development and academic research.

Keywords: numerical analysis, artificial intelligence, teaching mode, algorithmic thinking

1. Introduction

Numerical analysis, as an important branch of mathematics, mainly studies the numerical solutions of mathematical problems on computers. It approximates or approximately solves various complex mathematical problems through a series of numerical methods and algorithms [1]. In the fields of modern engineering, science, and technology, numerical analysis plays an indispensable role. Especially when dealing with complex mathematical models that cannot be solved analytically, numerical analysis provides efficient solutions for researchers and engineers. With the rapid development of technology, especially the continuous progress of artificial intelligence, the application scope of numerical analysis has become increasingly wide, and its importance in various aspects such as scientific computing, simulation, and optimization design has been continuously increasing. Against this background, the teaching mode of the numerical analysis course is facing major changes. In 2018, the Ministry of Education issued the "Innovation Action Plan for Artificial Intelligence in Higher Education", emphasizing the promotion of educational and teaching reforms in schools and the exploration of new teaching models based on artificial intelligence [2].

Traditional numerical analysis teaching usually focuses on teaching various numerical solution methods, such as interpolation methods, Newton's method, methods for solving linear equations, etc. However, with the increasing complexity of engineering and scientific research tasks, a single computing tool is no longer sufficient to meet the requirements. Numerical analysis is not only a collection of mathematical methods but also a way of thinking [3]. Especially when facing complex problems, how to start from the essence of the problem and reasonably select and design appropriate algorithms has become the key to solving problems. At present, some scholars have studied teaching methods based on algorithm design [4,5]. In addition, with the rise of technologies such as artificial intelligence and big data, numerical analysis is widely applied in many modern fields such as structural optimization, fluid mechanics, weather simulation, and financial modeling. Among them, algorithm design and optimization are the core aspects, directly affecting the computing efficiency and the accuracy of the solution. Therefore, it is crucial to cultivate students' algorithmic thinking.

Algorithmic thinking is a systematic problem-solving approach that emphasizes starting from the overall situation, deriving step by step, reasonably allocating resources, and optimizing the execution path to ultimately achieve problem-solving [6,7]. In the teaching of the numerical analysis course, students should not only master numerical methods but also select appropriate algorithms according to the characteristics of the problem, optimize the execution efficiency, and control errors. This paper aims to study how to improve the teaching effect of the numerical analysis course through a teaching model based on algorithmic thinking. For this purpose, first, the challenges in the existing teaching will be analyzed, then the implementation steps of the new teaching model will be given, and finally, through specific teaching practice cases, it will be shown how to integrate the cultivation of algorithmic thinking into teaching. In the context of modern engineering and science and technology, the teaching of the numerical analysis course is not just the imparting of technology but also the cultivation of thinking patterns.

Through this teaching model, students can not only improve their numerical calculation ability but also expand the thinking framework for solving practical problems, laying a solid foundation for their future career and academic research.

2. The limitations of the traditional teaching mode of numerical analysis courses

Numerical analysis courses play an important role in the science and engineering education system. Its core objective is to cultivate students' ability to apply numerical algorithms and solve practical problems. However, traditional teaching methods often focus on theoretical explanations and algorithm derivations, paying less attention to the integration of algorithm design thinking and actual application scenarios. As a result, although students can understand mathematical derivations and master numerical methods, there is still room for improvement in engineering practice and scientific research innovation. From the perspective of teaching models, traditional numerical analysis courses have certain limitations.

2.1. Overemphasis on the imparting of theoretical knowledge

The traditional teaching method of numerical analysis focuses on the derivation of algorithms and the explanation of mathematical principles, neglecting the specific implementation and practical application of algorithms. When students understand the mathematical background of algorithms, they lack a comprehensive understanding of how algorithms solve practical problems.

2.2. Disconnection between practice and theory

Although numerical analysis has a wide range of application backgrounds, traditional teaching often lacks a close connection with actual engineering problems. Students' learning of numerical algorithms mainly stays at the theoretical and paper calculation levels, lacking the opportunity to use algorithms to solve practical problems in actual scenarios.

2.3. Lack of cultivation of students' algorithmic thinking

Many algorithms in the numerical analysis course have a rigorous theoretical basis in mathematics. However, when it comes to specific applications, students often do not form a systematic thinking framework. Students are unable to compare and select different numerical algorithms, nor can they start from engineering problems and design reasonable and efficient numerical algorithms.

2.4. Single learning methods

The teaching method in traditional classrooms often mainly involves lectures. Students passively listen to the lectures, lacking opportunities for active thinking and participation. Homework and exams are also often limited to the operation of formulas and the application of methods, failing to effectively cultivate students' innovative abilities and the ability to solve practical problems.

Therefore, in order to improve students' practical application abilities and algorithmic thinking, the Numerical Analysis course needs to innovate the teaching mode and cultivate students' ability to flexibly apply numerical algorithms. Especially in the process of solving complex engineering problems, students should be able to optimize the calculation process by designing efficient algorithms.

3. Teaching mode of numerical analysis course based on algorithmic thinking

In order to better promote the development of students' algorithmic thinking, the teaching mode of numerical analysis course must be innovated, emphasizing the cultivation of students' thinking mode and problem-solving abilities. In the traditional teaching mode of numerical analysis, students tend to focus on memorizing algorithms and applying formulas, while neglecting the analysis of the essence of problems and the cultivation of problem-solving ideas. Therefore, in the process of innovating the teaching mode, algorithmic thinking should be integrated into all aspects of teaching, emphasizing the cultivation of students' understanding of algorithms and practical application abilities.

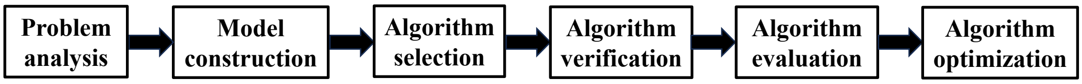

Fig. 1. Flowchart of the teaching mode of numerical analysis based on algorithmic thinking.

The teaching mode of the numerical analysis course based on algorithmic thinking is a teaching framework that closely combines the learning of numerical analysis algorithms with practical applications. The specific process is presented in Fig. 1

This model systematically guides students through six key steps, from theoretical learning to practical application, from algorithm selection to optimization, helping them gradually master the core content of numerical analysis algorithms. This teaching model not only focuses on students' understanding of algorithms but also emphasizes verifying the effectiveness and performance of algorithms through practice, thereby enhancing students' ability to apply numerical analysis algorithms in practical problems.

3.1. Problem analysis

In the problem analysis stage, students need to deeply analyze practical problems, clarify their background, conditions, goals, and limitations to better understand the core of the problems. For example, in physics problems, students should master the relationships between different physical quantities; in economic problems, they need to understand the mutual influence among costs, revenues, and market variables. Teachers can use specific cases to assist students in understanding problems. For example, by using the discharge process of an RC circuit, problems in fluid dynamics, etc., teachers can guide students to decompose problems and extract parameters in class. In addition, group cooperation is also an effective method. Students can jointly analyze problems and extract key parameters. Finally, through group reports and discussions, a deeper understanding can be promoted.

3.2. Model construction

Through the analysis of practical problems, students should transform them into mathematical expressions, which usually involves choosing appropriate mathematical tools, such as algebraic equations, differential equations, or optimization models. In this process, students need to master common modeling techniques and make reasonable assumptions and simplifications by combining the specific characteristics of the problems. Teachers can use specific cases to guide students to understand how to abstract mathematical models from practical problems. For example, they can select the heat conduction problem in the engineering field or the investment return analysis in the financial field as examples to help students build models step by step. In addition, teachers can organize group discussions and encourage students to exchange different types of modeling methods to cultivate teamwork spirit and improve their ability to analyze and solve problems.

3.3. Algorithm selection

After completing the modeling, students should select appropriate numerical algorithms based on the mathematical characteristics of the problem and the solution requirements. Commonly used algorithms include: for linear equations, the Gaussian elimination method or LU decomposition method can be used; for differential equations, the Euler method or Runge-Kutta method, etc. can be adopted. When choosing an algorithm, factors such as its applicable conditions, computational complexity, and accuracy requirements need to be comprehensively considered. Students should deeply understand the basic principles and complexity of the selected algorithm, and pay particular attention to convergence, stability, error analysis, and computational cost. For example, in the process of solving non-linear equations, students should understand the differences in convergence between the Newton method and the bisection method and make a reasonable choice according to specific needs. Teachers can explain the applicable scenarios and advantages of common algorithms through examples. For instance, introduce the applications of the Gaussian elimination method and the LU decomposition method in different computational tasks. In addition, teachers should also encourage students to compare the computational efficiency and accuracy of different algorithms for the same problem and guide them to analyze the specific reasons for choosing a certain algorithm.

3.4. Algorithm verification

In the algorithm verification stage, simulation experiments are conducted to test the accuracy, stability, and computational efficiency of the selected algorithm. During this stage, students transform the selected numerical algorithm into programming code and implement it on a computer. Programming not only tests students' programming skills but also helps them gain a deep understanding of the specific implementation of the algorithm. To evaluate the effectiveness of the algorithm, students need to design a reasonable experimental plan, select appropriate test data, set boundary conditions, and determine evaluation criteria. In the process of experimental design, as many scenarios as possible should be covered to comprehensively evaluate the actual performance of the algorithm. Through experimental runs, students can collect key data such as the execution time and error of the algorithm and conduct comparative analysis with theoretical results. This not only helps students recognize the limitations of the algorithm but also provides a reference for optimization. Therefore, teachers should reasonably arrange the programming practice time, encourage students to write code independently and conduct experimental verification to strengthen their numerical analysis capabilities. In addition, students are required to write an experimental report, detailing the experimental steps, data analysis, and optimization suggestions, which will help cultivate their scientific writing skills and critical thinking.

3.5. Algorithm evaluation

After completing the algorithm verification experiment, students need to conduct a systematic analysis of the algorithms they have implemented. Key evaluation indicators include calculation accuracy, execution efficiency, memory consumption, and convergence speed. By comparing the results of different experiments, students can identify the strengths and limitations of the algorithms and gain a deeper understanding of their operating characteristics. A comprehensive algorithm analysis not only helps students better understand the applicability of various algorithms but also contributes to the improvement of their logical thinking and optimization abilities. To this end, teachers should guide students to use programming tools such as MATLAB, Python, and C language to quantify and visually present the experimental results, making the data easier to interpret. In addition, teachers should also assist students in sorting out the advantages and limitations of different evaluation methods and encourage them to record their experiences and reflections during the learning process to further promote the improvement of algorithmic thinking abilities.

3.6. Algorithm optimization

After completing the algorithm evaluation, students should identify its performance bottlenecks and optimize it based on these to improve the efficiency and application effectiveness of the algorithm. According to the evaluation feedback, students can take different measures to improve the algorithm, such as accelerating the iterative convergence speed and adjusting algorithm parameters. The optimization plan needs to be reasonably designed according to the specific bottlenecks. The optimized algorithm should be tested again to ensure that the improvement effectively enhances the overall performance. Through repeated feedback and adjustment, students can not only deepen their understanding of the optimization process but also master more optimization techniques. This process helps students master practical algorithm optimization methods and improve their practical application abilities. Therefore, experimental sessions should be added to the curriculum to encourage students to optimize through methods such as adjusting parameters and improving algorithms. In addition, teachers can organize group discussions to allow students to share optimization experiences and gain inspiration and innovation from others' methods.

4. Teaching Cases

This section uses an engineering example of an RC circuit to demonstrate how to integrate algorithmic thinking into the chapter "Numerical Solutions of Ordinary Differential Equations".

Case: In actual engineering applications, RC circuits are widely used in many fields such as signal filtering and power supply voltage regulation. Consider a classic series RC circuit, in which the resistor R and the capacitor C are connected in series [8]. At the initial moment, the capacitor is fully charged with electrical energy and then discharges through the resistor. During this process, the voltage and current exhibit specific dynamic characteristics as they change with time. Studying the dynamic behavior of RC circuits is crucial for understanding the evolution laws of voltage and current over time.

The following steps demonstrate how to teach this topic using a teaching mode based on algorithmic thinking.

Step 1: For the classical RC circuit, let $V(t)$ denote the voltage across the capacitor, and suppose the initial voltage is $V(0) = V_0$. R represents the resistance, and C the capacitance. According to physical laws, one can derive the relationship among these quantities.

Step 2: The purpose of this step is to transform the dynamic behavior of the RC circuit into a mathematical model and clearly define the solution objective. According to Kirchhoff's voltage law, the governing equation is

$$\frac{\mathrm{d}V(t)}{\mathrm{d}t} = -\frac{1}{RC}V(t). \tag{1}$$

This is a first-order linear ordinary differential equation. Although it has an analytical solution

$$V(t) = V_0 \exp\left(-\frac{t}{RC}\right), \tag{2}$$

to better cultivate algorithmic thinking, we will use different numerical algorithms to solve it.

Step 3: In the dynamic analysis of RC circuits, students need to select appropriate numerical algorithms based on the RC circuit model. For the RC circuit model, combining various numerical algorithms for differential equations taught by teachers, we will choose three common numerical algorithms, namely the Euler method, the improved Euler method, and the fourth-order Runge-Kutta method. According to the textbook content and classroom explanations, these three methods each have their own characteristics and applicable scenarios. The Euler method is a basic numerical calculation method that predicts the next value through the slope of the current point. This algorithm is simple to implement and fast in calculation, but has relatively large errors. The improved Euler method combines the slopes of the current point and the predicted point, and improves the accuracy through one correction. Compared with the Euler method, the improved Euler method significantly reduces errors and enhances stability, but also brings more computational burden. The fourth-order Runge-Kutta method is a numerical algorithm with relatively high accuracy. It estimates the next value through the slopes of multiple points. This algorithm has high accuracy and stability, but the computational complexity is relatively high. Next, we will verify the accuracy of these conclusions through experiments to ensure their scientific nature and reliability.

Step 4: To verify the effectiveness of the three selected numerical algorithms, we designed a specific simulation scenario. Assume $R = 1\,\mathrm{k}\Omega$, $C = 1\,\mu\mathrm{F}$, the initial voltage $V_0 = 5\,\mathrm{V}$, and the time step $\Delta t = 10^{-4}$s. Under this scenario, different numerical algorithms are used to simulate and obtain the numerical solution of the RC circuit model so as to verify the validity of the algorithms. To compare the accuracy and computational cost of different methods, we record the error curves and the computation time of each algorithm. Figure 2 shows the error curves of the three algorithms. Meanwhile, the computation times for the Euler method, the Improved Euler method, and the fourth-order Runge-Kutta method throughout the entire process are 1.31×10^{-6}s, 1.73×10^{-6}s, and 2.98×10^{-6}s, respectively.

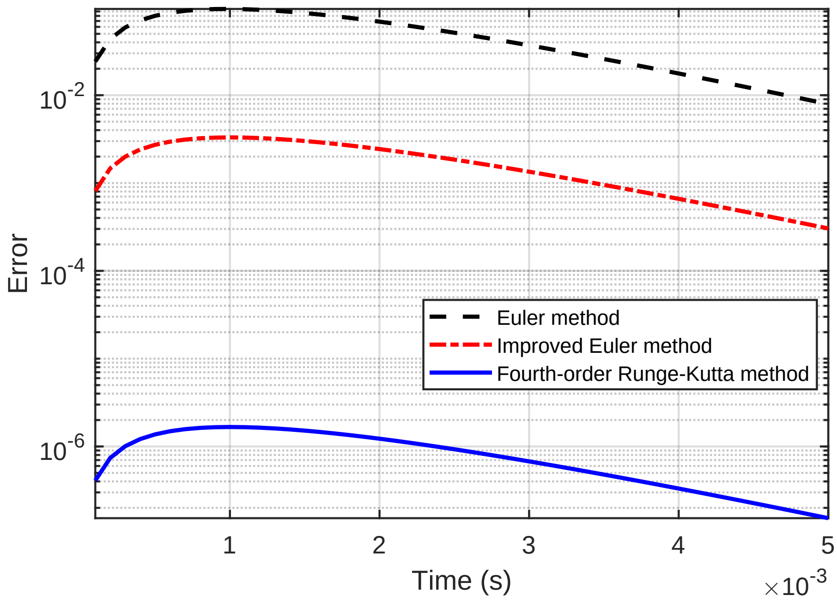

Fig. 2. Error curves of the three numerical algorithms.

Step 5: After completing the numerical solution experiment of the RC circuit model, students are required to evaluate three algorithms. The purpose of this process is to conduct an in-depth analysis of the experimental results, compare the advantages and disadvantages of each algorithm, and help students understand their characteristics more clearly. Based on the experimental data, the following conclusions can be drawn: First, the fourth-order Runge-Kutta method provides the most accurate solution and can approximate the true solution of the RC circuit model very well, but its computational cost is relatively high. Second, the improved Euler method has better accuracy than the Euler method, but its accuracy is lower than that of the fourth-order Runge-Kutta method. At the same time, the computational cost of the improved Euler method is lower than that of the fourth-order Runge-Kutta method but higher than that of the Euler method. Finally, the Euler method has the lowest accuracy and is suitable for applications with low accuracy requirements, but its computational cost is the lowest among the three algorithms. In conclusion, the experimental results are highly consistent with the theories learned in textbooks and the content explained by teachers in class, further verifying the accuracy and reliability of the relevant knowledge.

Step 6: When solving the RC circuit model, the choice of time step size plays a crucial role in accuracy and computational effort. A larger time step size helps improve computational efficiency, but it may lead to accuracy loss and numerical instability. To address this issue, an effective strategy is to use an adaptive time step size: increase the step size in regions with small errors, and decrease it in regions with large errors or rapid changes. This can improve computational efficiency while ensuring the accuracy of the results. In addition, controlling errors is of great importance in optimizing numerical methods. By tracking the error of each calculation and adjusting the calculation parameters accordingly, the accuracy and stability of the results can be significantly improved.

5. Conclusions

This paper explores the teaching mode of the numerical analysis course integrated with algorithmic thinking, highlights the importance of algorithmic thinking for this course, and puts forward the specific steps and cases for implementing this mode. Through this teaching method, the numerical analysis course not only focuses on the teaching of numerical algorithms and theories but also endeavors to cultivate students' systematic algorithmic thinking, helping them apply these algorithms more effectively to solve real-world problems. Looking ahead, with the continuous progress of technology and education, the teaching mode based on algorithmic thinking will play an even more crucial role in cultivating innovative talents and promoting the development of science and technology.

Conflict of Interest

The authors declare that there is no funding and no confffict of interest.

Acknowledgments

This work was supported by the Undergraduate Teaching Reform Projects of Fuyang Normal University (Grant Nos. 2023JYXM0039, 2022KCSZSF16, 2024JYXM0018, and 2025JYXM0011); the Teaching Research Project of Higher Education Institutions of Anhui Province (Grant No. 2022jyxm1161); the Undergraduate Teaching Reform and Practice Project of Henan University (Grant No. HDXJJG2023-128); the New Undergraduate Program Quality Improvement Project of Anhui Province (Grant No. 2024xjzlts035); the Natural Science Research Projects of Higher Education Institutions of Anhui Province (Grant Nos. 2023AH050429 and 2023AH050413); and the Doctoral Research Foundation of Fuyang Normal University (Grant Nos. 2021KYQD0034 and 2025KYQD0031)

References

1. Q. Li, N. Wang. Numerical analysis. Beijing: Tsinghua University Press, 2008.
2. Ministry of Education of the People's Republic of China. Notice on Issuing the "Innovation Action Plan for Artificial Intelligence in Higher Education", Gazette of the Ministry of Education of the People's Republic of China (2018) 127-135.
3. Z. Xie. Theory of Numerical Analysis and Its Thinking and Teaching, Journal of Qiannan Normal College of Nationalities 9(2006) 27-31.
4. C. Wei, K. Liao. Research on Algorithm Design Teaching in Numerical Analysis Course, Journal of Luoyang Normal University 32(2013) 140-142.
5. X. Wu, Z. Xiong, Y. Tang, J. Liu. Teaching of Algorithm Design in Numerical Analysis Curriculum, Journal of Hunan University of Technology 24(2010) 68-71.

6. Y. Yu, R. Song. Teaching Algorithmic Thinking in the Digital Era: A Backward Design Framework and Action Practices, Journal of Distance Education 42(2024) 86-94.
7. M. F. Byrka, A. V. Sushchenko, A. V. Svatiev, V. M. Mazin, O. I. Veritov. A New Dimension of Learning in Higher Education: Algorithmic Thinking, Journal of Educational Psychology-Propositos y Representaciones 9(2021) e990.
8. Q. Wang, Y. Li, X. Huang, G. Zhang. Based on the mathematical principle of RC circuit response analysis, Computer Knowledge and Technology 18(2022) 115-117+124.