ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 8, Issue 4, April 2022

Development of Thinking Skills in Teaching Numerical Analysis Courses

Jing Kong[a], Liang Fang[a,*]

[a] College of Mathematics and Statistics, Taishan University, 271000, Tai'an, Shandong, China

[*] *Email:fangliang3@163.com*

***Abstract*—This paper focuses on the computational and engineering ideas included in the teaching of numerical analysis from the perspective of the course characteristics of numerical analysis. We discussed how to develop students' thinking skills in numerical analysis course from the aspects of teaching contents, teaching methods and teaching process. The actual teaching results show that these improvements have improved students' interest in learning and their ability to use computers to solve problems**

***Index Terms*—Numerical analysis, Teaching content, Computational thinking, Engineering thinking.**

I. Introduction

Numerical analysis is a major part of computational Mathematics, which focuses on numerical methods and their software implementation for solving various mathematical problems by computer. Almost all branches of mathematical science are linked to computational mathematics, which uses the results of research in mathematics to develop new and more effective algorithms and theories, and in turn many branches of mathematics need to explore and study numerical methods suitable for computer use. The two complement each other and reinforce each other [1-3].

Numerical analysis course is based on the basic content of advanced mathematics, linear algebra and differential equations courses, using programming language as a means, using the computer as a tool to solve problems, to introduce the numerical methods and theory of solving common mathematical problems in engineering and scientific experiments. The main contents include error analysis of numerical computation, numerical approximation and curve fitting, solution of linear equations, roots of nonlinear equations, numerical differentiation and numerical integration, numerical solution of ordinary differential equations and partial differential equations, etc.

II. Features of Numerical Analysis Courses

Numerical score course was originally a compulsory course for undergraduates in mathematics and computer science, and with the development of computers, the importance of numerical computation has become more and more prominent, and corresponding computational courses have been offered in other majors, such as computational physics, computational chemistry, computational biology, computational economics, etc. Numerical computation has become the third important scientific research method after scientific experiments and theoretical studies. The characteristics of numerical analysis courses can be summarized as follows.

A. Provide computer-oriented theoretically reliable numerical algorithms of good computational complexity

The core of numerical analysis is the study of various numerical computational methods for the application of computers to solve mathematical problems, and the associated theoretical analysis of each algorithm. The convergence and numerical stability of the approximation algorithm must be ensured, and the error must be analyzed. For approximation problems it is important to ensure that the required accuracy is achieved, and it is also necessary to ensure that the algorithms provided are practically feasible on the computer. This includes the requirement that the algorithm has good time complexity and space complexity.

B. Emphasis on completing the whole process from theory to practice.

Numerical analysis is a highly practical mathematics course. Each algorithm should be proven to work through numerical tests in addition to being theoretically correct and feasible. After learning each algorithm students should aim at solving practical problems and complete the training of numerical computation by programming or with the help of mature mathematical software. Not only must the solution to the problem be known to exist, but also specific results must be derived. Some methods, although not yet theoretically rigorous, can also be used if they prove to be effective by means of actual computational comparisons, analysis, etc.

C. The calculation formula is long and difficult to remember

Numerical analysis courses have the following distinctive features in the approach to problems: (1) Adopting a constructive approach, the proof of the existence of many problems are completed by the specific construction of the problem formula, not only to prove the existence of the problem, but also to provide a specific formula. (2) Discretization method is used to transform a mathematical problem of continuous variables into a problem of discrete variables. For example, the differential equation is discrete into a difference equation, etc. (3) Using recursive methods, a

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

IJARMATE
Your ult-MATE Research Paper !!!

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 8, Issue 4, April 2022

complex computational process is reduced to multiple iterations of a simple computational process in order to write a computer program to compute it. (4) Approximate substitution method is used. Because the computer must stop after a finite number of operations, so the numerical method is often expressed as an infinite process of truncation, the mathematical problem of an infinite process into a finite step operation to meet certain error requirements to approximate substitution. These basic features make the numerical analysis course appears in the computational formulae are many and complicated, not easy to memorize.

The numerical analysis method not only provides a way to solve mathematical problems by approximation using computers and other tools, but also deepens students' understanding of related contents in advanced mathematics, and opens up their horizons and enlivens their minds. It is also wrong to think of numerical analysis as simply doing mathematical exercises. Numerical analysis methods, compared to other basic mathematics courses, have significantly changed students' perceptions of university mathematics education by introducing new ideas and perspectives. From the point of view of mathematical applications, these ideas are exactly what is needed for mathematical applications and what is lacking in basic university mathematics education. Therefore, emphasizing the teaching of numerical methods courses can have a great effect on students' understanding of mathematics and on their ability to apply it.

III. Computational and Engineering Thinking in Teaching Numerical Analysis

A. Computational thinking in numerical analysis teaching content

Computational thinking was first proposed in 2006 by Professor Ezin Zhou, a senior vice president at Microsoft and a professor at Carnegie Mellon University [4-5]. Since the concept of Computational Thinking was introduced, the computer education community at home and abroad has conducted in-depth research and discussions on the essential features and core methods of computing. So what is computational thinking? In an interview, Professor Yi-zhen Zhou gave a minimalist explanation that computational thinking is thinking like a computer scientist. From this statement, we can see that computational thinking represents a way of thinking. This type of thinking uses computers to formulate and propose solutions to problems in a way that computers can effectively perform. In other words, computational thinking is essentially a way of thinking that uses computers to solve real-world problems. In addition, in order to better understand computational thinking, one must first figure out what computation is. Usually, when we talk about calculation, people's minds immediately think of various arithmetic operations in mathematics. Such as addition, subtraction, multiplication, division, integration, differentiation, and so on. In computer science, all objects including text, sound, image, and video are represented in binary in a computer. And computation in computers includes arithmetic operations, relational operations, logical operations, etc. Moreover, the values "0" and "1" can represent both numerical values and logical "true" and "false", so that the logical judgment process can also be realized by calculation. So, what is called computation in computer science is much broader than our usual understanding of computation. Once you understand the nature of computation, you will find that many seemingly incalculable things become calculable, and it will be easier to understand the universality of computational thinking. Therefore, some real-life problems, after some abstraction, can be described in a specific rigorous mathematical language to describe the problem solving process, and then, we can solve the problem in a computational way. For a particular problem, a program is designed to solve the problem and a computer executes this program, which is known as computation in computer science.

In general, the most important processes in computational thinking are abstraction, decomposition, and combination. “Abstraction” means ignoring all the irrelevant and complex details and focusing only on the key parts of the problem. “Decomposition” means dividing the problem into smaller parts, breaking them down one by one, and finally “combining” the parts to form a concrete solution.

The solution of many problems in this course of numerical analysis reflects the use of computational thinking. Therefore, in learning this course, from the perspective of cultivating students' computational thinking, first of all, students can grasp the main ideas of algorithms from a macroscopic point of view, without being bound to the derivation of details, so that the level of all kinds of students can also be taken into account. The following is a compilation of the main methods in the computational methods course and the typical computational thinking embodied in the computer implementation of the method, the results of which are shown in Table 1.

TABLE I: THE MAIN CONTENTS TAUGHT IN NUMERICAL ANALYSIS COURSES

Main teaching content	Computational Thinking
Gaussian elimination method	Simplification, transformation, recursion
Direct triangular decomposition method	Decomposition, transformation
Lagrange interpolation method	Discretization, recursive
Least squares method of data fitting	Transformation, separation of concerns
Main teaching content	Computational Thinking
Compound Integral	Discretization, transformation, iteration, stepwise improvement

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 8, Issue 4, April 2022

Numerical solution of ordinary differential equations	Simplification, transformation, recursion
Main teaching content	Computational Thinking
Successive approximation method	Iteration, transformation

B. Engineering thinking skills in teaching numerical analysis content

The methods of numerical analysis have a strong engineering background, and each of them is directly or indirectly related to engineering applications. Function interpolation and curve fitting have been widely used in experimental data processing and appearance design problems. The spline function is one of the foundations of image processing techniques. FFT technology is able to process data quickly and has a pivotal role in real-time signal processing in mechanical, electronic, information and automation engineering. Many models in physics and chemistry boil down to various differential equations. Therefore, the numerical solution of differential equations plays a great role in solving various practical problems.

(1)The idea of approximation and approximate approximation.

Function approximation is one of the main elements of numerical analysis methods, and many numerical methods rely on the idea of function approximation. In advanced mathematics, the emphasis has always been on the rigor of logic and the accuracy of mathematical calculations, and the approximations often taken in function approximation are quite different from the previous practice. For students, they often ask the question: Does this approach make sense? Of course, convergence analysis and error analysis in numerical analysis courses also answer these questions. The numerical approximation using interpolation functions makes students realize that the numerical analysis course is not simply a mathematical exercise, but a training in how to use existing mathematical knowledge and tools to approximate the solution of the original problem through the analysis of the original problem. As a new way of thinking, approximation and approximation make students realize that it is not scary to be unable to solve a problem analytically or precisely, but scary to be unable and afraid to use the mathematical knowledge they have learned to approximate and simplify the original problem so as to obtain an approximate solution to the original problem.

(2) Iterative thinking.

Iteration is an important concept in computing and also in numerical analysis methods. The basic idea is as follows: first, students are asked to dare to guess a solution, but of course a good guess is in the vicinity of the real solution. Secondly, we examine whether the guess satisfies the requirements of the solution, and if it does, the guess is successful; if not, we proceed to the next guess. This process is considered as one iteration [6-7]. So, how can we guess the next solution based on the previous one? Different iterative methods have different guesses. The idea of iteration is to encourage guessing and correcting guesses based on the results. This idea is not only applicable to numerical methods, but also to scientific research. In the process of mathematical modeling, guessing the possibility of the result can help us to a large extent in the choice of modeling direction, so that we can reduce many detours. Since most of the iterative methods have only a finite convergence interval, it is important to use the available information to make guesses about the solution, which relies on the students' experience in applying mathematical analysis theory and various methods in practice.

(3) The idea of discrimination of continuous problems.

In engineering calculations, we often need to solve continuity problems, such as the solution of differential equations. In general, it is difficult to find analytical solutions to differential equations, so solving differential equations numerically is an important element in computational methods. Numerical solution of a differential equation does not rely on the computer to give the analytical form of the differential equation, but relies on it to approximate the value of the differential equation as a function of a specified point. Therefore, the idea of discretization is introduced into the numerical method. The idea of discretization requires partitioning the solution into isolated lattice points in the variable space, and approximating the value of the solution at the lattice point as a lattice function. After introducing the idea of discretization to discretize the problem, we can use various numerical methods to solve the value of the lattice function. By discretization, the original continuity problem becomes a discrete problem. The idea of discretization is one of the fundamental ideas of numerical computation, and the existing numerical computation depends almost entirely on the discretization of the problem. Therefore, the discretization of the problem and the generation of the corresponding mesh has always been an important aspect of numerical analysis research.

IV. THE DEVELOPMENT OF THINKING SKILLS IN TEACHING NUMERICAL ANALYSIS

In the actual teaching process, it is worth studying how to make students master the basic methods and the fundamental principles of algorithms in the numerical analysis course in the limited classroom teaching. In this limited number of hours, even if we simply teach the basic principles of algorithms in the textbook, the hours are very tight, not to mention the implementation of programming. Moreover, simply teaching the basic principles of algorithms and the derivation of computational formulas will make students feel that the course is no different from a mathematics class, and students will have little interest and enthusiasm in learning. Therefore, in the classroom teaching, we should focus on teaching the basic ideas of numerical analysis and analyze the thinking methods used in the methods, so as to improve the

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 8, Issue 4, April 2022

students' computational thinking ability and be able to refer to similar problems by analogy.

In actual teaching, the development of thinking skills can be incorporated in several aspects, such as teaching content, teaching methods, and assessment of teaching effectiveness.

A. In the arrangement of teaching content, to focus on the content of the lecture

The original focus on the mathematical principles of numerical analysis, the mathematical derivation of specific algorithmic processes, should be transferred to explain the basic principles of numerical analysis, focusing on the implementation of the computer, so that students focus on the use of computers to solve mathematical problems in general methods and steps. At the same time, in order to increase students' interest in learning, some tools (such as MATLAB) can be used for demonstration. MATLAB software has efficient numerical calculation functions and complete graphic image processing functions, which can visualize the calculation results easily and quickly, which is unmatched by general programming languages. If students use programming language to implement the calculation process, they may have to consider too many details and the result may be incorrect, which may cause frustration in the learning process. At the same time, it is easy to ignore the essence of the algorithm in the computation process because of the excessive focus on the details of the solution process. Once students have mastered the main idea of the algorithm, it is not difficult to implement the program. Therefore, the teaching content focuses on the basic principles of algorithms. For students with poor programming skills, they can use the tool software to implement the calculation results and specifically appreciate the role and principles of the algorithm. For students with strong programming skills, they are encouraged to implement the algorithms programmatically. Of course, the depth of understanding is completely different when comparing implementation through programming and using tool software. Students who can implement the algorithm by programming have a deeper understanding of the algorithm. Students who use tool software, on the other hand, understand the algorithm mainly at the principle level, and some details of the specific implementation cannot be taken into account.

B. Adopt task-driven teaching in the teaching method.

The Numerical Analysis course addresses a different problem in each chapter, which makes it ideal for using task-driven instruction. In the teaching process, prepare teaching cases in advance, preferably from real-life examples, to increase students' interest in learning. After the task is clarified, the scope of knowledge involved in the problem is clarified by analyzing the problem. Through discussion, possible solutions to the problem and steps are identified. In this way, students not only improve their problem-solving skills, but also develop the ability to think independently.

C. Pay attention to the order of teaching contents in the teaching process

In the teaching process, theoretical lectures should be combined with practical applications, software demonstrations and student exercises. First, choose good cases, such as from real-life practical problems, for example, explain the interpolation problem, you can relate to the real-life relationship between the change of house prices and time in a city, you can get the change curve of house prices through interpolation, through the relationship curve of house price changes, you can also simply predict the future trend of house prices. In the process of explaining the principles of algorithms, appropriate animations can be used to give students a more visual and concrete experience of the thinking process. After the explanation, some simple problems can be assigned for students to practice solving the algorithm in class, so that students can really feel the practical value of the algorithm.

V. CONCLUSION

Numerical analysis course as part of the basic mathematics course, it is very different from other mathematical analysis courses, it emphasizes the emphasis on computer applications, methods and solutions to practical problems of computational and engineering ideas, it is these methods and ideas make it play a fundamental role in mathematics laboratory courses and mathematical modeling teaching. Of course, the numerical analysis course is not a panacea, for example, the discrete problems in numerical analysis course are very different from the discrete problems in discrete mathematics, and a considerable part of the content in operations research is not covered in numerical analysis course. Since numerical analysis courses teach students many new ideas and methods, they do not feel a huge obstacle in learning other new things. If teachers can emphasize the practical aspects of numerical analysis and the engineering aspects of numerical analysis in their courses, they will get twice the result in terms of educating students in the application of mathematics.

REFERENCES

[1] M.-Y. Huang, B. Liu, T. Xu, *Numerical calculation method*. Beijing: Science Press, 2012, pp. 37-242.
[2] Q.-Y. Li, N.-C. Wang, D.-Y. Yi, *Numerical analysis*. Beijing: Tsinghua University Press, 2008, pp. 22-277.
[3] F.-S. Bai, *Introduction to Numerical Computation*. Beijing: Higher Education Press, Inc., 2012, pp. 23-236.
[4] G.-L. Chen and R.-S. Dong, “Computational thinking and basic university computer education,” *Teaching at Chinese Universities*, vol. 1, pp. 21-23, April 2011.
[5] Y.-Z. Zhou, “Computational thinking,” *CCA Newsletter*, vol.3, pp. 76-78, November 2007.
[6] Y.-Z. Zhu, “On Computational Thinking - Scientific Positioning, Fundamentals and Innovative Paths of Computational Thinking,” *Computer Science*, vol.4, pp. 52-56, July 2009.
[7] Z.-H. Yi, L.P. Jiao, W. Ou, “Exploration of teaching models based on computational thinking,” *Computer Knowledge and Technology*, vol.9, pp. 4276-4278, October 2013.

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 8, Issue 4, April 2022

Author's' biography with Photo

Jing Kong was born in August 1977 in Qufu, Shandong Province, China. She is a lecturer at Taishan University. She received her PhD degree from Shanghai Jiao Tong University in December 2011. Her research interests are in the areas of discrete mathematics, numerical analysis, and optimization problems.

Liang Fang was born in December 1970 in Feixian County, Linyi City, Shandong province, China. He is a professor at Taishan University. He obtained his PhD from Shanghai Jiaotong University in June, 2010. His research interests are in the areas of cone optimizations, numerical analysis, and complementarity problems.