

AI Empowerment and Indigenous Software Integration: A New Paradigm for New-Form Textbook Development in Numerical Computation Courses

Xindong Li^{1,*}, Wenwen Xu², Yongqiang Ren³, Sen Xu⁴

^{1,2,3}School of Mathematics and Statistics, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250353, China

⁴School of History and Culture, Changchun Normal University, Changchun 130032, China

¹xindongli@qlu.edu.cn, ²wenwenxu@qlu.edu.cn, ³renyongqiang2003@163.com

*Corresponding Author

Abstract: *The diversification of academic software accessibility has prompted higher education institutions to explore indigenous alternatives for scientific computing platforms, while national policies on educational digitization mandate the integration of artificial intelligence across all teaching elements. Numerical computation courses, serving as a bridge between mathematical theory and engineering applications, must adapt to these dual imperatives of software diversification and AI integration. This paper proposes a three-dimensional textbook development framework: systematic integration of indigenous software platforms as primary computing engines, embedding of AI-powered case studies that demonstrate how classical numerical algorithms underpin modern machine learning applications, and development of new-form digital textbooks featuring interactive visualizations, executable code snippets, and intelligent assessment capabilities. This paper presents concrete implementation strategies, including algorithm library development, AI-infused case design, and digital resource construction. The framework aims to cultivate students' proficiency in diverse software ecosystems while enhancing their understanding of the "AI for Science" paradigm, thereby serving both pedagogical innovation and the broader goal of educational resource diversification.*

Keywords: Numerical Computation, New-Form Textbooks, Indigenous Software, AI-Enabled Education.

1. Introduction

The landscape of scientific computing education in higher education is undergoing significant transformation driven by two converging developments: the diversification of software accessibility in academic settings and the pedagogical imperative of artificial intelligence integration.

In recent years, changes in the international academic software environment have prompted universities and research institutions to explore alternative platforms for scientific computing education. As one of the most widely used platforms for numerical computation, algorithm development, and data visualization in STEM education, the restricted availability of certain commercial software has posed challenges to course delivery, student training, and research continuity. This situation has accelerated the development and adoption of indigenous software alternatives, creating new opportunities for educational innovation. National policies have emphasized the promotion of indigenous industrial software adoption, signaling a commitment to technological diversification [1]. Concurrently, national education policies have emphasized the integration of artificial intelligence into all aspects of teaching and learning [2]. This directive reflects a broader recognition that AI is not merely an add-on to existing curricula but a transformative force that reshapes what and how students learn across disciplines [3].

Numerical computation courses occupy a unique position at this intersection. These courses teach students how to approximate solutions to mathematical problems that cannot be solved analytically, using algorithms implemented on

computers. The course content ranges from solving linear systems and nonlinear equations to interpolation, numerical integration, and ordinary differential equations. The pedagogical challenge lies in balancing theoretical rigor with practical computational competence. However, existing textbooks for numerical computation face three significant limitations [4]. First, most textbooks remain closely tied to specific commercial software platforms, with limited coverage of alternative or indigenous options. Second, while some recent textbooks have introduced Python, they rarely address the deeper integration of AI concepts—treating AI as a superficial mention rather than as a substantive dimension of algorithmic thinking [5]. Third, the predominant format remains static print, lacking the interactive features, dynamic visualizations, and intelligent feedback mechanisms that characterize new-form digital textbooks [6-7].

This paper addresses these gaps by proposing a comprehensive framework for developing numerical computation textbooks that simultaneously achieve three objectives: (1) systematic integration of indigenous software platforms as primary computational environments, (2) embedding of AI-powered case studies that bridge classical numerical algorithms with modern machine learning applications, and (3) transformation of the textbook from static content to an interactive digital learning platform.

2. Literature Review

2.1 The Evolution of New-Form Textbooks

New-form textbooks integrate multiple media resources, break the limitations of traditional print carriers, and support

interactive learning experiences. According to recent scholarship, these textbooks are characterized by content diversity, support for personalized learning, and interactive communication capabilities. The development path has progressed from simple “print + QR code” appendages to fully integrated “print + digital platform” ecosystems [8].

For STEM courses like numerical computation, new-form textbooks offer particular advantages. The field is inherently visual and computational—algorithms are best understood when students can see them in action, modify parameters, and observe the effects in real time. Interactive visualizations of iterative methods, dynamic demonstrations of convergence behavior, and embedded code execution environments can significantly enhance learning outcomes [9-10].

2.2 Indigenous Software in STEM Education

The push for software diversification in education has gained substantial momentum in recent years. Among the emerging alternatives, MWORKS has emerged as a leading platform. MWORKS is built on Julia, a high-performance scientific computing language, and Modelica, a multi-domain unified modeling language, providing an integrated environment for scientific computing, data analysis, algorithm design, and system modeling and simulation [11].

Recent studies have documented the successful integration of indigenous software across various disciplines. In scientific computing and simulation courses, researchers have explored these platforms as alternatives to commercial software, noting their technical viability and the importance of cultivating student familiarity with diverse software tools [12-13]. A new textbook, *Scientific Computing and Simulation Applications* (Based on MWORKS), has been published to systematically introduce the scientific computing environment using MWORKS 2025a, covering numerical computation methods suitable for science and engineering students [14]. Similarly, *Control System Aided Design and Simulation Verification* (Based on MWORKS) has been developed with the aim of cultivating user ecosystems for indigenous industrial software [15]. Parallel developments have occurred with other indigenous platforms. Research on the Beita Tianyuan platform has demonstrated its effectiveness in partial differential equation numerical solution instruction, enabling complete teaching implementation from mesh generation to numerical visualization, and summarizing replicable pathways for integrating indigenous software into STEM curricula [16]. A comprehensive textbook, *Beita Tianyuan Scientific Computing Programming and Applications*, has been published systematically presenting the platform’s numerical computing capabilities and data visualization functions [17]. Expert reports have further highlighted the integrated design of numerical computation textbooks that combine theory, case studies, and indigenous software code [18].

2.3 AI Integration in Numerical Computation Education

The integration of AI into numerical computation curricula has been explored from multiple angles. National educational policies emphasize “comprehensively advancing intelligence

and promoting AI-assisted educational transformation,” calling for the construction of AI education models and the development of high-quality educational resources [2]. This policy framework provides strong support for integrating AI into foundational STEM courses.

Recent research has examined the feasibility and effectiveness of AI integration in numerical computation education. Studies on generative AI-assisted numerical iteration method teaching have demonstrated that AI technologies significantly enhance students’ understanding of mathematical principles and programming implementation capabilities [19]. A “teacher-student-AI” ternary interaction teaching model has been proposed for numerical analysis course design, leveraging generative AI as a core driver of instructional innovation [20]. Research on AI-powered personalized instruction for practical numerical methods has proposed dynamic personalized learning pathways and programming tasks based on AI capabilities [21]. Broader reform efforts have constructed a comprehensive “content restructuring, model innovation, and evaluation reform” framework for graduate-level numerical analysis instruction in the AI context [22].

However, the literature reveals a gap: most existing work treats AI as an external addition to numerical computation rather than as a domain where numerical algorithms find deep application. Machine learning models—from neural networks to support vector machines to Gaussian processes—are fundamentally built upon numerical optimization, linear algebra, and statistical computation. Helping students recognize these connections can deepen their understanding of both classical algorithms and modern AI [23].

Furthermore, AI can serve not only as subject matter but also as a pedagogical tool. AI-powered recommendation systems can suggest personalized learning paths, intelligent assessment can provide immediate feedback, and AI-assisted content generation can help instructors develop tailored materials [24].

3. The Framework for Textbook Development

3.1 Theoretical Foundation

The proposed framework is grounded in three educational theories. Outcome-Based Education (OBE) emphasizes designing teaching activities and assessment methods oriented toward learning outcomes. Under the framework, each textbook chapter’s objectives are clearly defined and aligned with final capability outcomes. Constructivist Learning Theory posits that knowledge is actively constructed by learners through interaction with their environment. The framework provides students with multi-scenario learning experiences, from foundational theory to interactive practice, facilitating active knowledge construction. Technology-

Enhanced Learning suggests that digital technologies can transform learning experiences. The framework leverages new-form textbook technologies to create interactive, personalized, and engaging learning environments [25].

3.2 Framework Structure

Algorithm Library Development. The core algorithms of numerical computation—including direct methods for linear systems (Gaussian elimination, LU decomposition), iterative methods (Jacobi, Gauss-Seidel, conjugate gradient), interpolation methods (Lagrange, Newton, spline), numerical integration (trapezoidal, Simpson, Gaussian quadrature), nonlinear equation solving (bisection, Newton-Raphson, secant), and ordinary differential equation solvers (Euler, Runge-Kutta)—are implemented, validated, and performance-tested on the platform. The resulting algorithm library serves as the computational backbone of the textbook. Students can execute, modify, and extend these implementations as they work through the material, building both conceptual understanding and practical proficiency.

AI as Application Domain. A set of case studies illustrates how classical numerical algorithms find application in AI and machine learning. These include gradient descent and optimization algorithms as the workhorse of deep learning training, neural network weight initialization drawing on numerical linear algebra, physics-informed neural networks (PINNs) for solving differential equations, and AI-driven error correction for traditional algorithms. Each case study follows a consistent structure: theoretical background, software implementation, AI application context, result analysis, and extension exercises. This “algorithm principles → implementation → AI integration → result analysis” cycle helps students see the connections between classical and contemporary computational approaches [19].

Restructured Pedagogical Sequence. Rather than treating AI as an afterthought, the framework restructures the pedagogical sequence to reflect a “problem-driven” philosophy. Each chapter follows a six-module structure: Problem Drive, Algorithm Modeling, Software Implementation, AI Extension, Case Analysis, and Online Practice. This structure breaks from the traditional “theory → examples → exercises” format and emphasizes the connection between algorithms and engineering problems.

Interactive Features. The digital textbook incorporates several interactive capabilities: executable code that students can run directly within the textbook environment, dynamic visualizations showing algorithm iterations and convergence behavior, embedded assessment with automatic scoring, and AI-powered personalized learning path recommendations [21].

Platform Integration. The digital textbook integrates with existing learning management systems to enable seamless data exchange. This integration supports learning analytics, instructor dashboards, and collaborative learning activities.

Resource Ecosystem. The textbook is complemented by a comprehensive digital resource ecosystem: complete script library, AI case study collection, instructional video repository, online question bank with varying difficulty levels, and instructor’s guide with teaching suggestions.

This “print + digital + platform” three-in-one resource system supports both instructor-led teaching and self-directed student

learning.

4. Pedagogical Strategies for Implementation

4.1 Textbook Structure Reconstruction

The traditional “theory → examples → exercises” structure is replaced with a six-module structure per chapter: Problem Drive, Algorithm Modeling, Software Implementation, AI Extension, Case Analysis, and Online Practice. This restructuring strengthens the connection between algorithms and engineering problems, making learning more contextualized and motivating.

The **Problem Drive** module opens each chapter with a real-world problem that motivates the need for specific numerical methods. The **Algorithm Modeling** module presents the mathematical formulation and algorithmic design. The **Software Implementation** module provides code development on the indigenous platform. The **AI Extension** module explores connections to AI applications where relevant. The **Case Analysis** module offers an in-depth examination of a comprehensive example. The **Online Practice** module provides interactive exercises with automated feedback.

4.2 Software Platform Integration

Teaching content is systematically adapted to incorporate indigenous software implementations alongside existing approaches. A teaching script and function library is developed, forming a comprehensive teaching package that supports smooth transition for students. This includes side-by-side syntax comparisons and workflow migration guides.

The integration process follows a three-step approach. First, core algorithmic implementations are adapted to the new software environment with careful attention to syntax differences and performance characteristics. Second, pedagogical materials including lecture slides, lab exercises, and assignment problems are updated to reflect the new environment. Third, assessment methods are revised to evaluate students’ proficiency with the platform alongside their understanding of numerical algorithms [16].

4.3 AI Case Embedding

Typical teaching cases are designed around “AI + Numerical Computation” themes: using neural networks to solve initial value problems of ordinary differential equations, employing genetic algorithms to optimize numerical integration nodes, applying reinforcement learning in boundary value problem solving, and deep learning-assisted approximate computation of matrix eigenvalues. Each case is accompanied by theoretical explanations, code implementation, experimental tasks, and extension questions [22].

4.4 Digital Textbook Development

Digital resources are developed using new-form textbook platforms. Features include online code execution, dynamic visualization of algorithm iteration processes,

AI-recommended learning paths and personalized exercises, embedded testing and automatic grading, and data interoperability with teaching platforms.

The digital textbook development process emphasizes pedagogical usability alongside technical sophistication. Each interactive feature is designed to address a specific learning challenge identified in traditional instruction. For example, dynamic visualizations of iterative methods help students understand convergence behavior that is difficult to convey through static text and equations alone.

5. Expected Outcomes

Student Learning Outcomes. Students participating in the reformed curriculum are expected to achieve high proficiency in the adopted software platform. Their AI-integrated innovation capabilities should show significant enhancement, with comprehensive project completion quality improving markedly and learning interest in numerical computation increasing substantially.

Teaching Quality Outcomes. Teaching evaluation satisfaction is expected to improve substantially. The model should generate a replicable new-form textbook teaching application pattern that can serve as a reference for similar course reforms.

Strategic Outcomes. The project aims to stimulate student interest in learning, strengthen practical engineering capabilities and independent innovation awareness, and cultivate potential users and developers for the indigenous software ecosystem.

6. Discussion

6.1 Implications for Practice

The proposed framework has several practical implications for educators and curriculum developers. First, it demonstrates that the transition to indigenous software platforms can be accomplished without sacrificing pedagogical quality, provided that adequate transition support and teaching materials are developed. Second, it shows that AI integration need not be superficial; by embedding AI applications throughout the curriculum, students can develop both computational skills and AI literacy simultaneously. Third, it illustrates how new-form digital textbooks can serve as more than just digital versions of print materials, functioning instead as interactive learning environments that support personalized and active learning.

6.2 Limitations and Future Directions

This study has several limitations. The framework has been developed based on experiences at a single institution, and its generalizability to other contexts requires further investigation. The effectiveness of the proposed pedagogical strategies has not yet been rigorously evaluated through controlled studies. Future work should focus on empirical assessment of learning outcomes, comparative studies with traditional approaches, and adaptation of the framework to different institutional contexts and disciplinary settings.

Additionally, as AI technologies continue to evolve rapidly, the case studies and examples embedded in the textbook will require periodic updating to remain current. Establishing mechanisms for ongoing content review and revision will be essential for sustaining the framework's relevance.

7. Conclusion

This paper has proposed a comprehensive framework for developing numerical computation textbooks that simultaneously addresses three pressing challenges: the need for software diversification in academic settings, the imperative to integrate AI across STEM curricula, and the opportunity to leverage new-form digital textbook technologies.

The framework's three dimensions—indigenous software infrastructure, AI-infused content design, and new-form digital delivery—offer a coherent response to these challenges. By integrating indigenous software throughout the curriculum, the framework cultivates student proficiency in diverse software ecosystems. By systematically connecting classical numerical algorithms to AI applications, it helps students understand the deep relationships between foundational computational methods and modern machine learning. By transforming the textbook from static content to interactive digital platform, it leverages technology to enhance learning outcome.

Acknowledgements

This work was supported the Jinan Municipal City-University Integration Project – Artificial Intelligence High-Level Talent Training System and Innovation Platform Construction Project (Grant No. JNSX2024044), the Undergraduate Teaching Reform Key Project of Qilu University of Technology (Shandong Academy of Sciences) (Grant No. 2025zd12), and the “AI+” Classroom Teaching Reform Project – Numerical Analysis (Grant No. 452401130123).

References

- [1] Ministry of Industry and Information Technology of the People's Republic of China. Guidelines for Equipment Renewal in Key Industrial Sectors. Beijing: MIIT, 2024.
- [2] Ministry of Education of the People's Republic of China. Opinions on Accelerating the Digitalization of Education. Beijing, MOE, 2025.
- [3] W. Zhao, G.Z. Song, X.Z. Du, “On Strategies to Construct an Evaluation System for Talent Training in Applied Colleges and Universities,” *The Science Education Article Collects*, 310, pp. 25-26, 2015.
- [4] M.X. Liao, “Research and Practice on the Construction of Quality Evaluation System in Applied Talents Training,” *Vocational & Technical Education Forum*, 23, pp. 55-59, 2016.
- [5] B. Jia, X.D. Chen, H. Li, “Construction and Practice of the Evaluation System of Professional Learning Outcomes under OBE,” *The Guide of Science & Education*, 8, pp. 42-44, 2017.
- [6] Y. Lu, Y.J. Zhang, H.Q. Han, “Construction of Quality Evaluation System of Talent Cultivation of Human Geography and Urban-rural Planning Majors Guided by

- OBE,” Journal of Heilongjiang Vocational Institute of Ecological Engineering, 31, pp.135-137, 2018.
- [7] Y.W. Gao, “Morphological Evolution, Construction Status and Impact of University Mathematics Textbooks,” Higher Education of Sciences, (1), pp. 52-59, 2026.
- [8] K.F. Liu, Y.P. Zhu, J.H. Wang, “Research on Quality Evaluation System of Mathematics Application Talents Training,” Journal of Higher Correspondence Education (Nature), 23, pp. 16-18, 2010.
- [9] Z.Q. Fu, J.X. Zhao, P.P. Yan, “Wisdom Teaching Context Graduate Numerical Analysis Course New-Form Textbook Construction Analysis,” University Education, 23, 2025
- [10] D.L. Li, X.J. Wang, W.Z. Yang, “Reflections on New-Form Textbooks for Numerical Analysis,” Higher Mathematics Research, 04, pp. 53-55, 2024.
- [11] Tongyuan Soft Control. MWORKS: Scientific Computing and System Modeling and Simulation Platform. Suzhou: Tongyuan Soft Control, 2024.
- [12] X.D. Li, Computational Methods Based on Python, Electronics Industry Press, Beijing, 2023.
- [13] M. Zhou, “On Quality Evaluation System for Training Financial Management Talents in Application-Oriented Undergraduate Colleges,” Journal of Heihe University, 11, pp.104-105, 2019.
- [14] Y.L. Wang, G.H. Zhang, C.X. Du, J.W., Zhou, Z.Z. Zhang, H. Yuan, Scientific Computing and Simulation Applications (Based on MWORKS). Publishing House of Electronics Industry, Beijing, 2026.
- [15] X.F. Chang, Yan, T., X. Guo, et al. Control System Aided Design and Simulation Verification (Based on MWORKS). Publishing House of Electronics Industry, Beijing, 2025.
- [16] B. Huang, T. Lu, W.Q. Yao, “Innovations and Practices in Teaching of Numerical Solutions of Partial Differential Equations Based on Baltamatica,” Journal of Mathematics, 46(4), pp. 229-242, 2026.
- [17] R. Li, T. Lu, Beita Tianyuan Scientific Computing Programming and Applications. Peking University Press, Beijing, 2025.
- [18] A.W. Wang, From Classroom to Research: Beita Tianyuan Empowers Talent Cultivation, 2025.
- [19] Y.H. Xu, L.P. Zhang, “Teaching Design and Case Study of Numerical Iteration Methods Based on Generative AI—A Case of the Numerical Analysis Course,” Advances in Education, 15(8), pp. 587-596, 2025.
- [20] J. Zhang, “Teacher-Student-Machine Ternary Interaction: Research on the Innovative Teaching Model of Numerical Analysis Course Design Empowered by AI,” Creative Education Studies, (11), pp. 292-299, 2025.
- [21] C. Yu, Y.H. Feng, “Research on Personalized Teaching and Programming Ability Improvement of Practical Numerical Methods Based on AI Large Models,” The Guide of Science & Education (Electronic Edition), (31), pp.293-295, 2025.
- [22] L. Li, Y.G. Zhang, Y. Wang, S.L. Li, D.D. Lei, “Challenges and Breakthroughs in Graduate Numerical Analysis Instruction under the Background of Artificial Intelligence: Reform and Practice,” Education and Traditional Culture, 2(4), Article 7335, 2025.
- [23] X.X. Hu, P. Zhao, D. Yan, “Study of Construction on Education Quality Evaluation System Based on Teaching Department,” Journal of Liaoning Provincial College of Communications, 16, pp. 48-51, 2014.
- [24] D. Luo, Y. Xi, “Research on Quality Evaluation System of Talent Cultivation for Applied Mathematics Major in Applied Technology Colleges and University,” Dang Dai Jiao Yan Lun Cong, pp. 8-9, 2017.
- [25] H.D. Xu, N.Z. Chen, G.Q. Xu, H.T. Di, “Research on Quality Evaluation System of Forest Engineering Talents Cultivation Under the Background of Double First-Class,” Journal of Social Science of Jiamusi University, 39, pp.187-191, 2021.

Author Profile



Xindong Li received the Ph.D. degrees in Computational Mathematics from Shandong University, China in 2014. He is currently an Associate Professor in mathematics and artificial intelligence at the Qilu University of Technology (Shandong Academy of Sciences), China. His research interests include numerical solution of partial differential equations and reservoir numerical simulation.



Wenwen Xu received the Ph.D. degrees in Computational Mathematics from Shandong University, China in 2016. She is currently an Associate Professor in mathematics and artificial intelligence at the Qilu University of Technology (Shandong Academy of Sciences), China. She is mainly engaged in the research of numerical methods of partial differential equations, including mixed finite element and multi-point flux mixed finite element decoupling algorithm.



Yongqiang Ren received the M.S. and Ph.D. degrees in Computational Mathematics from Shandong University, China in 2009 and 2015, respectively. He is currently an Associate Professor in mathematics and artificial intelligence at the Qilu University of Technology (Shandong Academy of Sciences), China. His research

interests include numerical solutions of differential equations and
reservoir numerical simulation.