Advances in Education 教育进展, 2026, 16(1), 386-392
Published Online January 2026 in Hans. https://www.hanspub.org/journal/ae
https://doi.org/10.12677/ae.2026.161053

Hans汉斯

计算思维融入近世代数课程的教学改革研究

尹富纲

北京交通大学数学与统计学院，北京

收稿日期：2025年12月1日；录用日期：2025年12月28日；发布日期：2026年1月5日

摘 要

近世代数课程因内容高度抽象，传统教学模式往往以理论证明为中心，学生易陷入机械化的“符号演算”，难以真正掌握并灵活运用知识。为此，本文提出在教学中融入计算思维的改革思路，通过增设计算实践模块，引导学生“在做中学”，加深对知识的理解。具体措施包括：利用JavaScript等开发可交互的代数计算网页，引入GAP等专业代数软件，并结合AI辅助编程降低技术门槛，让学生更聚焦于算法设计与数学思想。研究表明，该模式能有效激发学生的学习兴趣与探究动力，促进对代数结构本质的理解，培养其数字化建模和系统性解决问题的能力。

关键词

近世代数，计算思维，GAP，AI辅助编程

Research on Teaching Reform Integrating Computational Thinking into Modern Algebra Courses

Fugang Yin

School of Mathematics and Statistics, Beijing Jiaotong University, Beijing

Received: December 1, 2025; accepted: December 28, 2025; published: January 5, 2026

Abstract

Modern algebra courses are highly abstract in content, and traditional teaching methods often focus heavily on theoretical proofs. This leads students to fall into the mechanistic “symbolic calculation” mode, making it difficult for them to truly master and flexibly apply the knowledge. To address this, this paper proposes a reform approach by integrating computational thinking into teaching. By adding computational practice modules, students are guided to “learn by doing”, thereby deepening

文章引用: 尹富纲. 计算思维融入近世代数课程的教学改革研究[J]. 教育进展, 2026, 16(1): 386-392.
DOI: 10.12677/ae.2026.161053

their understanding of the subject matter. Specific measures include: developing interactive algebraic computation webpages using tools such as JavaScript, introducing professional algebraic software like GAP, and leveraging AI-assisted programming to lower technical barriers, allowing students to focus more on algorithm design and mathematical thinking. Research has shown that this model effectively stimulates students' learning interest and exploratory motivation, enhances their understanding of the essence of algebraic structures, and cultivates their abilities in digital modeling and systematic problem-solving.

Keywords

Modern Algebra, Computational Thinking, GAP, AI-Assisted Programming

Copyright © 2026 by author(s) and Hans Publishers Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).
http://creativecommons.org/licenses/by/4.0/
Open Access

1. 传统近世代数教学的痛点与契机

1.1. 传统教学的困境

近世代数课程中群、环、域等核心概念呈现高度形式化的特点，传统教学长期围绕理论证明展开，课堂以“定义 - 定理 - 证明”为固定结构，作业与考核也主要强调逻辑推导与形式证明。这导致学生在学习过程中易于陷入机械的“符号演算”，难以真正理解并灵活运用相关理论。由此产生两个突出表现：一是学生逐渐感到课程枯燥，学习参与度与兴趣下降；二是课程结束后，由于缺乏深层次理解与结构性联系，相关知识很快被遗忘。

笔者认为，上述现象根植于传统教学往往采用被动灌输、机械接受的学习模式，未能建立以学生为主体的知识建构机制。以“商群”的教学为例，教师通常重点强调商群运算需验证“良定义性”，并解释其原因是陪集代表元的选择不唯一。这种教学模式易使学生将“良定义”简化为需记忆的孤立知识点，停留于“教师讲授 - 学生记录”的表面化认知层面。

我们主张构建一种主动探索、自主建构的学习模式，一个载体即为引入计算思维。例如，可设计一个“商群构造器”实验任务，让学生在编程中亲历概念的生成与验证过程。学生需要主动应对若干关键问题：如何表示陪集——是直接存储陪集对应的子集，还是仅保存一组代表元？如果选择代表元表示，如何系统地选取一组陪集代表元？进一步，在定义陪集之间的运算时，若两个代表元的乘积不在预设的代表元集合中，应如何处理？这一问题将自然引导学生思考：为何必须验证商群运算的“良定义性”——即运算结果是否依赖于代表元的选择。通过解决这一系列递进式问题，学生不再被动接受结论，而是自主建构对商群结构与运算本质的理解。最终，学生不仅掌握了商群的定义，更获得了将抽象结构转化为可计算模型的能力，真正实现了从“听数学”到“做数学”的转变。

1.2. 改革契机：计算思维与 AI 技术的赋能

近年来，如何将计算思维引入数学课程，已成为国际数学教育界关注的重要议题。2021 年，第 14 届国际数学教育大会(ICME-14)将“编程与算法教学”列为重点专题，深入探讨了大学数学课堂中编程和算法教学的理论与实践问题[1]。2024 年，中国教育科学研究院孟鸿伟系统研究了计算思维的定义、特征及其与数学思维的内在联系[2]。尽管计算思维虽然没有统一的定义，但其核心实践包括抽象、分解、算法设计、模式识别、调试与迭代五个维度，为近世代数教学改革提供了新路径。

这五个维度在代数学习中体现为：第一，抽象的双重实现——学生需要在数学抽象的基础上进行第二次抽象，将代数结构转化为可计算的数据模型；第二，问题的系统性分解——将复杂代数问题拆解为可执行的子任务；第三，算法的形式化设计——将数学思路转化为明确的步骤序列；第四，模式的发现与归纳——通过编程实例发现数学规律；第五，迭代调试与验证——将程序错误转化为深化理解的契机。

与此同时，人工智能编程助手的快速发展为解决技术门槛问题提供了有效途径。过去，试图将编程融入数学课程的努力都会面临一个根本矛盾：学生需要花费大量时间学习编程语言的语法、库函数的使用、调试技巧等，这些技术性细节往往会喧宾夺主，挤占本应用于数学思考的时间。现在，学生可用自然语言描述数学问题，AI 助手能生成相应代码实现。例如，学生可以描述“请编写判断有限群是否为循环群的 Python 函数”，AI 能够提供完整的代码框架。这种变革具有深远意义：学生不再需要精通编程语法即可实现数学想法，能够将认知资源集中于问题建模与算法设计，快速尝试不同实现方案，同时培养与 AI 协作解决问题的数字时代素养。

计算思维为代数知识向实践能力转化提供了方法论指导，而 AI 工具则为这一转化提供了技术支撑。二者的有机结合，使教学重心从“如何编码”转向“为何这样设计算法”，真正建立起以数学思维为核心、以计算实践为支撑的新型教学模式，既顺应教育数字化发展趋势，又契合培养创新型、实践型数学人才的现实需求。

2. 课程改革框架设计

理论教学在保持理论深度的基础上增强计算视角。课堂讲授仍然以教材为主线，确保学生对群、环、域等基本概念有严格的理论理解。在讲解定义和定理时，有意识地引入计算视角。例如，在讲解商群时，不仅给出定义和性质，还讨论“如何在计算机中表示商群，如何存储商群元素及运算”。此外，理论作业中包含需要思考算法和实现策略的问题，如“设计一个算法，求出一个子群的所有陪集”。

实验教学采用分层递进的设计思路。开设专门的实验课程，按照从基础到复杂的序列设计实验内容：初级阶段聚焦代数结构的基本实现，包括群的构造、子群陪集计算、正规子群判定、置换运算、U(n)群计算、n 阶交换群分类等；中级阶段涉及更复杂的代数操作，包括群同态的实现与验证、多项式的欧几里得算法实现等。在教学过程中，指导学生建立标准化的人机协作工作流程：首先明确数学问题的计算转化，接着设计算法思路与逻辑框架，然后向 AI 描述实现需求，随后评估 AI 生成代码的正确性与效率，最后进行测试调试与算法优化。

专业工具应用拓展数学探索边界。引入 GAP (Groups, Algorithms, Programming)专业代数计算系统[3]，该软件是专为计算离散代数设计的开源跨平台工具，已成为群论及相关领域研究者和教育者的重要工具。通过 GAP 官网提供的丰富教学案例，特别是 Hulpke 编写的《Abstract Algebra in GAP》学生手册[4]，引导学生将自编代码与专业软件的计算结果进行对比分析。在此基础上，指导学生利用 GAP 探索高斯整环的理想、多项式的不可约性等复杂问题，拓展数学视野。

与教学改革相适应，考核体系进行系统性重构。强化过程性评价，将贯穿学期的编程实验与专题作业纳入核心考核范畴，约占最终成绩的 30%~40%，以彰显对学生实践能力与持续学习过程的重视。作业要求中特别强调反思性学习，学生需提交至少两次“AI 工具使用”反思日志，详细记录在借助 AI 工具过程中遇到的挑战、决策调整及学习收获，培养元认知能力。

期末考核采用理论考查与综合实验相结合的方式。期末考试仍以闭卷笔试形式进行，重点考察学生对基本概念、定理及其证明的理解与掌握。同时设置独立的期末综合实验项目，要求学生在集中时间段内完成一个具有完整性的代数计算或可视化课题。这既是对其整个学期所学计算思维与编程能力的综合

检验，也模拟了解决真实问题的完整流程。这种“日常实验 + 期末笔试 + 综合项目 + 过程反思”的多元评价体系，力求更全面反映学生在知识、能力与素养三个维度的发展水平。

3. 具体举措

3.1. 基于 Web 技术的交互式代数对象构建与计算

传统教学中，学生是代数概念的被动接受者；而通过构建自己的代数计算工具，他们成为数学对象的主动创造者。浏览器环境(HTML/CSS/JavaScript)提供了理想的平台：MathJax 库支持高质量的 LaTeX 数学公式渲染；现代 JavaScript 的交互能力使得创建动态可视化的代数对象成为可能；轻量级的特性便于分发和分享。

在实施层面，首先由教师主导搭建网站框架，统一规划整体 UI 风格与模块化设计的接口规范，并预先定义代码提交的数据格式及参数要求。随着课程推进，学生在完成特定实验作业(例如 U(n)求解器的实现)后，需按照教师规定的格式，须按规范以 JSON 等格式提交函数实现及相关配置。教师团队通过自动化脚本将学生提交的模块集成至主站，实现功能的持续扩展与系统维护的集中管理。同时，利用自动化测试脚本对提交代码进行统一的功能验证与效率评估。

案例：U(n)求解器(见图 1)

1) **实验目标**：输入**正整数** n，输出群 U(n)的所有元素，阶和所有的生成元。
2) **问题分析**：U(n)定义为所有小于 n 且与 n 互质的正整数在模 n 乘法下构成的群。需理解以下关键问题：如何表示和存储 U(n)的群元素？如何计算每个元素的阶(即满足 $a^k \equiv 1 \pmod{n}$ 的最小正整数 k)？如何找出生成元？
3) **算法设计与实现**：遍历 1 至 $n-1$，筛选出与 n 互质的数可得 U(n)。对每个元素 a，迭代计算 $a^k \bmod n$，直至结果为 1，首次出现的 k 即为 a 的阶；计算欧拉函数 $\phi(n)$，并找出阶为 $\phi(n)$ 的元素就得到了所有生成元。
4) **AI 辅助编程**：学生可向 AI 助手提出如下请求：“请帮我实现一个 *JavaScript* 函数，输入正整数 n，输出 U(n)群的所有元素、每个元素的阶、即所有的生成元。算法是：遍历 1 至 $n-1$，筛选出与 n 互质的数可得 U(n)。对每个元素 a，迭代计算 $a^k \bmod n$，直至结果为 1，首次出现的 k 即为 a 的阶；计算欧拉函数 $\phi(n)$，并找出阶为 $\phi(n)$ 的元素就得到了所有生成元。要求函数返回结构化的结果对象，适合在网页中展示。”
5) **提交程序**：用 JSON 文件提交程序，字段有函数名，函数参数和函数内容等。

Figure 1. U(n) group calculator
图 1. U(n)求解器

在学生构建 U(n)求解器的过程中，计算思维的五个核心维度得到了系统的训练。“抽象与建模”首先体现为将“模 n 乘法可逆元集合”这一数学定义，转化为程序中整数列表与互质判断函数的具体数据结构与逻辑；接着，“问题分解”引导学生把复杂任务拆解为生成整数序列、逐元素判断互质性、收集有效结果等可执行的子步骤；随之而来的“算法设计”阶段，学生必须为判断互质这一关键操作选择或实现具体算法(如欧几里得算法)，将数学思路形式化为明确的指令序列；程序运行后，“模式识别”能力在观察不同 n 值下群的阶、生成元分布和循环结构等输出规律时被激活，促使学生从具体计算中归纳数学性质；而贯穿始终的“调试与迭代”则要求学生在代码报错或结果异常时，不断回溯并厘清是数学理解偏差、逻辑设计缺陷还是技术实现错误，将每一次修正转化为深化理解的契机。这一完整流程，实质上构建了一条从数学概念到可计算模型，再通过计算实践反哺数学洞察的闭环学习路径。

通过构建 U(n)求解器的实践，学生能够有效克服近世代数的认知障碍，其深层机制源于多重认知与教育学原理的共同作用。首先，从情境认知与认知学徒制的视角看，该任务将学生置于“数学工具开发者”的真实角色中，使抽象的 U(n)群概念锚定于一个可操作、可检验的实践情境，有效对抗了传统教学中知识与应用脱节导致的“惰性知识”问题。其次，这一过程通过“生成效应”与“多重编码”深化了理解：学生并非被动接收信息，而是主动生成代码、调试错误，并同时处理数学符号、算法逻辑和可视化结果，这种深度加工显著强化了记忆痕迹，使关于互质判断、群运算封闭性等核心知识得以内化。更为关键的是，项目驱动的探索激发了学生的内在动机与系统化思维。求解器从特殊数字(如 U(8))到一般规律(如 U(35)的结构分解)的演进过程，自然引导学生发现并验证“$U(mn) \cong U(m) \times U(n)$ (当 m，n 互质)”等抽象定理，将孤立知识点整合为有机的知识网络。这种从直观体验到模式归纳，再到理论确证的认知路径，恰好吻合了从具体到抽象的人类认知规律。最终，可交互的求解器本身成为学生认知的“外化延伸”与“思维实验场”，使高度形式化的代数结构获得了具身化的认知锚点，从而在深层次上完成了从“符号演算”到“意义建构”的转变。本研究设计的教学范式，其有效性根植于坚实的教育心理学基础。根据建构主义观点，学生通过编程“建造”代数对象的过程，正是在主动建构知识意义。

3.2. 专业代数计算工具 GAP 的探索与应用

在学生通过 Web 编程初步掌握代数对象的计算实现后，教学进一步引入专业的代数计算系统 GAP。这一环节旨在引导学生从自主实现走向专业工具应用，通过系统的实践训练，培养其运用现代计算工具探索代数学前沿问题的能力。

教学从熟悉与验证开始。通过 1 次专门的实验课，学生掌握 GAP 的基本操作环境、核心语法和常用命令。重点学习使用“CyclicGroup”、“SymmetricGroup”等命令构造各类群，并用“Size”、“Elements”、“IsAbelian”等函数验证群的基本性质。这一过程让学生初步体验专业计算工具的高效与严谨，为后续深入学习奠定基础。

随着教学内容的推进，教师在课堂上结合具体知识点的讲解，及时引入相关的 GAP 命令，帮助学生建立理论概念与计算实践的直接联系。例如，在讲授群同态时，介绍“GroupHomomorphismByImages”等命令，演示如何构造同态、计算核与像；在讲解子群结构时，引入“AllSubgroups”等函数，展示如何系统获取和分析群的子群格。这种同步教学方式使抽象的理论概念变得具体可操作，增强了学生对代数结构的直观理解。

为了深化学生对 GAP 的掌握并培养其算法设计能力，课程作业特别设计了需要创造性思考的 GAP 实验任务。这些任务往往无法直接用 GAP 的标准命令完成，而是要求学生基于已学命令设计组合算法。例如，“找出子群的所有左陪集代表元”这一任务，虽然 GAP 提供了陪集计算的基础功能，但需要学生思考左陪集代表元和右陪集代表元的关系，如何设计算法生成左陪集代表元集合。又如，“判断两个群

是否同构”的作业，学生需要综合运用“StructureDescription”、“IsomorphismGroups”等命令，并考虑特殊情况下的判断策略。这类作业促使学生不仅学会使用工具，更要理解工具背后的原理，并能够为解决新问题设计合适的计算流程。

通过这种“课堂讲解－命令演示－作业实践”的循环，学生逐渐从 GAP 的简单使用者成长为能够灵活运用工具解决复杂问题的探索者。他们不仅掌握了 GAP 这一专业工具的使用技能，更重要的是培养了将代数问题转化为可计算任务、并设计有效解决方案的计算思维能力，为其未来的学术研究或专业发展奠定了坚实的基础。

3.3. AI 编程助手的教学融合：策略、挑战与应对

将 AI 编程助手引入近世代数教学，在带来显著机遇的同时也伴随着特定挑战，需要系统性的教学设计予以应对。这一融合也面临三重挑战：一是代码可靠性与“黑箱”风险，AI 生成的代码可能包含隐蔽的逻辑错误或非最优实现，若学生不加批判地接受，可能形成错误认知；二是思维惰性与过度依赖，学生可能倾向于直接索取最终代码，削弱了自主进行问题分解与算法设计的关键训练；三是学习的表面化，学生可能仅满足于获得能运行的代码，而忽略了对背后数学原理的深度挖掘。

为应对这些挑战，我们设计了更具操作性的三层教学策略。首先，在作业中引入“AI 代码辨析”任务，教师提供包含典型错误或可优化片段的 AI 生成代码，要求学生分析其逻辑漏洞、验证其正确性并提出改进方案，以此训练其批判性思维与调试能力。其次，建立常态化的 “同伴代码评审”(Peer Review)机制，学生以小组形式定期交换实验代码，依据明确的标准清单(包括算法正确性、效率、可读性等)进行互评并提供书面反馈，这一过程既能促进知识共享，也能通过审查他人代码来反观自身不足。最后，制定并实施结构化的“反思日志”评价量规，将反思内容具体化为可评估的维度，例如：问题向 AI 描述的清晰度、提示词迭代优化的过程、对生成代码的测试验证方法、从错误中获得的数学见解等。这套“辨析－互审－反思”的组合策略，旨在引导学生从被动的工具使用者，转变为主动的、有批判意识的 AI 协作学习者，确保技术赋能真正服务于数学思维与计算思维能力的深度融合。

4. 结论与展望

本研究的核心创新在于建立了计算思维与近世代数教学的有机联系，将高度抽象化的代数概念转化为可编程、可验证的算法实现。传统教学中学生面临的理论与实践脱节问题，通过“做中学”的教学设计得到有效解决——当学生动手实现 U(n)群计算器或商群构造器时，他们不仅理解了定义，更在代码调试中把握了概念的本质特征。这一方案重构了教学关系：教师成为学习引导者，学生成为知识建构者，AI 编程助手作为“思维脚手架”降低了技术门槛，GAP 等专业工具则拓展了探索边界。这种转变使数学学习从被动接受转变为主动创造，符合数字化时代对人才培养的新要求。

从认知层面分析，编程实践让抽象概念获得具象载体。学生在实现代数结构的过程中，能够直观观察数学规律，如通过 U(n)群计算器发现生成元分布、循环性质等特征，这种基于实证的学习方式建立的认知更为深刻和牢固。能力培养方面，完整的算法设计流程系统训练了计算思维——从问题分解、算法构思到代码实现和调试优化，这一过程培养了将复杂数学问题转化为可执行步骤的关键能力。教学实施层面，基于 Web 的技术平台支持分层教学，既能满足基础学生的验证需求，也为能力较强的学生提供深入研究的空间。

尽管方案优势明显，实施中仍需面对现实的制约条件。首先教师应具备一定编程与计算思维教学的系统训练，需通过持续的专业研修填补这一能力缺口。教学资源的建设同样关键，高质量、模块化的实验代码库、详尽的 GAP 操作案例以及自动化评估工具的开发，都需要投入大量时间与精力。课时安排也

是现实难题，如何在有限的教学时间内合理分配理论学习、编程实验与 GAP 探索的比例，需要精细的设计与调整。

从长远发展看，这一教学模式的潜力尚待充分挖掘。未来可围绕具体教学场景开展实证研究，例如对比传统班与改革班学生在概念理解深度、问题解决能力等方面的差异，用数据验证改革成效。在应用广度上，可将“计算实现 + 专业工具”的模式迁移至其他抽象数学领域。这种融合了经典理论与现代技术的教学模式，不仅有望提升单门课程的教学质量，更可能为整个数学教育体系的现代化转型提供可复制的范式。

参考文献

[1] Buteau, C., Rafalska, M., Chen, X. and Matkarimov, B. (2024) Teaching and Learning of Programming and Algorithms. *Proceedings of the* 14*th International Congress on Mathematical Education*, 375-378. https://doi.org/10.1142/9789811287152_0038

[2] 孟鸿伟. 面向数字化未来的“计算思维” [J]. 中国教育信息化, 2024, 30(2): 3-12.

[3] The GAP Group (2024) GAP—Groups, Algorithms, and Programming (Version 4.14.0). https://www.gap-system.org

[4] Hulpke, A. (2013) Abstract Algebra in GAP. https://www.math.colostate.edu/~hulpke/CGT/howtogap.pdf