学科规范如何影响数学家对本科数学中编程的看法

Jan-Fredrik Olsen[12] (jan-fredrik.olsen@math.lu.se)

Tor Ole B Odden[1] (t.o.b.odden@fys.uio.no)

编程已深深融入当代数学实践，但其在大学数学教学中的认识论地位仍存在争议。人们对数学家如何理解编程在其专业工作中的正当性，以及这些看法如何影响他们的教学，知之甚少。我们通过对北欧一所大学的 *15* 位数学家开展半结构式访谈，弥补这一研究空白。该校已在科学、技术、工程和数学各学科中系统性地融入编程二十余年。我们借鉴文化历史活动理论和实践共同体的视角，考察数学家如何阐述编程在研究与教学中的作用。

我们归纳出四种认识论原型——经典纯数学、经典应用数学、计算应用数学和计算纯数学，每种原型都体现了一套连贯的规范，用以界定编程的正当使用方式。在不同原型中，轻量编程（例如数值探索）广泛用于研究，却在教学中鲜少显现；在教学中，编程的正当性与更具“实质性”的融入方式相关。我们认为，这一差距反映了不同实践之间认识论上的连续性，以及教学对既定分析结果的侧重；这使计算工作的认识论可见性降低。

鉴于数学家对编程正当用途的阐述，我们的研究结果表明，最广受认可的融入方式是：在专门课程中教授实质性的编程，同时在传统理论课程中以轻量的认知性方式运用编程（例如进行数值探索），以支持学习。更广泛的计算工作通常被认为适合纳入专门的数值或计算数学课程。

关键词：数　学家、编程、数学实践、活动理论、实践共同体

引言

近两个世纪以来，编程这一概念一直影响着数学实践。早在1843年，艾达·洛夫莱斯就指出，只要遵循一系列符号指令，巴贝奇分析机便能够“编织代数图式，就像提花织机编织花朵和叶片”（转载于 Lapham’s Quarterly；Lovelace，未注明日期）。她预见，这类可编程机制能够使数学真理在“实际应用中/变得/ 更加迅速、准确。”

计算机辅助计算在数学中的合法性，既源于其对关注现实世界的数学家有价值，也源于其对关注理论的数学家有价值。事实上，这类计算对于在数学

[1] 科学教育计算中心，奥斯陆大学，Sem Saelands vei 24，Fysikkbygningen，0371 奥斯陆，挪威[2] 数学科学中心，隆德大学，Box 118，22100 隆德，瑞典

模型与现实世界数据之间发挥调和作用至关重要，同时也为探索理论猜想、可视化结构以及检验那些在分析上似乎难以处理的想法提供了机会（Broley 等，2018年；Lockwood 等，2019年）。

然而，计算在数学中的作用一直存在争议。Paul Halmos 曾对 Appel 和 Haken 关于四色定理的计算机辅助证明表示强烈反感，并大声说道："这个证明实际上依赖于一台预言机，我要反对预言机！它们不属于数学"（Hersh，1997年，第54页）。正如 Hersh 所解释的，Halmos 的判断并非基于对证明正确性的怀疑，而是出于对何谓"恰当的"数学论证的顾虑。这与相关研究的发现一致：数学家评价计算的使用时，不仅会考量逻辑依据，也会考量审美和认识论层面的因素，包括透明性、解释深度、美与纯粹性（Burton，2004年；Hamami 和 Morris，2020年）。

这些张力似乎也延续到了高等教育中，而且在这一领域似乎表现得更加明显；现有研究表明，编程在教学中的应用甚至少于在研究中的应用，尤其是在理论导向课程中（Buteau等人，2014年；Broley等人，2018年）。如果要以有意义且可持续的方式将编程融入数学教育，理解这种差异就至关重要。然而，数学家的研究实践与教学实践之间的联系仍未得到充分理解；先前的研究表明，数学家在这两个领域可能会采用不同的认识论（例如，参见 Burton，2004年）。

在本文中，我们访谈了15位数学家，探究他们如何在研究与教学实践中阐述编程的正当性。我们的研究以奥斯陆大学（UiO）为背景；二十多年来，该校一直将编程和科学计算正式纳入数学课程（Malthe-Sørenssen 等，2016年）。在呼吁研究编程在本科数学教育中的应用时，Lockwood 等（2021）明确指出，这一背景是一个关键研究场域，"为我们深入、系统地研究课程层面的协作提供了一个引人入胜的视角"。

在分析中，我们借助社会文化理论框架构建了一个认识论矩阵，根据数学家的主导客体取向（研究什么）和工具取向（如何研究）对其进行刻画。该矩阵有助于我们降低数据的异质性，并识别塑造编程在研究与教学中如何获得正当性的认识论规则。我们运用这一框架追踪一系列反复出现的张力，例如，基于研究的规范在教学中的持续影响，以及当数学活动从知识生产转向传授既有成果时，编程的认识论可见性随之减弱。本研究的研究问题列于"理论框架"一节末尾。

背景

数学家在数学教育研究中的实践

数学家的实践是数学教育研究中一个成熟的探究领域，Burton（2004）和 Nardi（2008）早期做出了重要贡献。Rasmussen 等人（2015）借鉴 Moschkovich（2007）关于专业话语实践的概念，将学科实践定义为"数学家从事其专业工作的方式"（第264页）。有关这一领域研究的概述，请参见 Weber 等（2020）。

Burton（2004）发现，数学家的研究实践比其课堂实践所呈现的更加多样，也更具探索性。她分析了七十次访谈，并报告说，数学家会用探索、直觉、合作、提出猜想和创造来描述自己的研究；然而，在教学时，他们却采用传递式模型——这种模型将数学呈现为柏拉图式领域中的逻辑结构，"完全遗漏了最初创造这些结果时人类经历的跌跌撞撞的过程"（Smith & Hungwe，1998年，转引自 Burton，2004年，第87页）。借鉴我们的理论框架，我们将这种现象称为数学教育中的客体-成果转变。

研究表明，数学家对自身认识论的描述（例如，参见 Polya，2014年）只能呈现出丰富多样的实践、信念和规范中的一部分。Burton（2004年）通过一种宽泛的认识论模型来概念化数学认知，该模型纳入了社会关系、审美，以及对不同探究方法的认知。相比之下，Weber 等人（2014年）提出了一种以认识论目标和行动为中心的认识论模型，即数学家实际上如何行动以获得确信。他们的研究结果表明，数学家有时会依赖经验性证据或权威性证据，有时即使有证明也无法完全确信，这凸显了其认识论行为的复杂性。

在此基础上，Weber 等（2020年）指出，研究数学家的实践面临若干方法论挑战。除了前文提到的异质性问题，他们还指出了准确性问题（数学家可能会无意中错误地描述自己的实践）、识别问题（“数学家”的定义因情境而异）、高级内容问题（专业语言可能会令外行难以理解其含义），以及跨学科性问题（要充分分析实践，需要结合数学史与数学哲学）。这些挑战共同表明，该领域的研究得益于与职业数学家的密切合作（另见 Darragh，2022年）。

数学家的实践中的编程

尽管数学教育研究日益关注数学家的实践，但只有少量研究考察了他们对编程的使用。鉴于计算在数学中的认识论地位长期以来一直存在争议，这一研究空白尤为值得关注，正如引言中提到的四色定理争议所示。

经验证据表明，这种张力也体现在数学家日常使用编程的情况中。在一项针对302名加拿大数学家的调查中，Buteau 等（2018）发现，18%的人在教学中使用编程，43%的人在研究中使用编程。在后续访谈中，Broley 等（2018）报告称，编程被认为在纯数学课程中几乎没有立足之地，但在服务课程或应用课程中则更受认可。作者将此归因于由制度与文化因素塑造的“地位差距”。总的来说，这些研究指出，数学家优先考虑的数学对象类型，以及他们认为正当的工具，在两方面都存在系统性差异。

与此同时，研究表明，使用编程的数学家强调其实际和认识层面的益处。Broley 等（2018）和 Lockwood 等（2019）报告称，数学家利用编程进行实验、检验猜想、实现可视化并进行近似计算。Broley 等（2018）还指出了编程被认为具有的教学优势：通过揭开黑箱并构建自己的工具，学生可以“感受到赋能和兴奋，这可能进一步提升他们的学习体验”。

数学教育研究中的编程

数学教育研究中计算的应用有多种理论阐释方式。一种广泛采用的框架是计算思维（CT），它将编程视为计算机科学家的学科实践，并认为这种实践能为科学、技术、工程和数学领域提供可迁移的问题解决技能（Wing，2006年；Weintrop 等，2016年）。数学教育领域有关计算思维的研究主要关注学生如何运用“算法式”问题解决技能（例如，Kotsopoulos 等，2017年；DeJarnette，2019年；Lockwood 等，2020年、2021年）。

另一种框架是计算素养（CL）（diSessa 2000, 2018），它建立在 Papert（1980）的观点之上：计算媒介是表达性工具，使学习者能够将数学思想呈现为可执行制品。用 Papert 的话来说，计算“让我们能够改变具体与形式之间的界线。”（第 21 页）。计算素养采取一种“全局视角”，关注表达性计算媒介的广泛普及如何重塑我们如何学习、为何学习，以及什么才算值得掌握的知识。

尽管出现了这些及其他理论进展，但人们对数学家如何在其专业实践中理解编程的正当性知之甚少。数学教育研究主要关注学生参与编程的情况以及计算工具在教学上的优势，却几乎未曾考察学科规范、身份认同和认识论承诺如何影响数学家在研究与教学中对编程的看法。鉴于数学共同体长期以来对计算持矛盾态度，且制度压力日益增大，要求将编程纳入本科课程，这一缺口尤为显著。

综上所述，本节回顾的研究表明，学科规范和身份认同塑造了数学家对研究和教学中编程的看法。然而，直接考察这种关联的研究还很少。受下一节将介绍的社会文化框架启发，我们采用与计算素养相契合的“宏观”视角：我们将编程视为一种实践，而不仅仅是一项技术技能；这种实践可能正在重塑数学家的知识、价值取向和行为。

理论框架

文化历史活动理论

我们运用 Engeström 的第三代文化历史活动理论，将数学家的研究和教学概念化为集体的、以对象为导向的活动（Engeström，2015）。活动围绕共同的“客体”展开——也就是驱动活动的目的。对数学家而言，客体可能是创造新的数学知识、维护学科规范，或教育学生。这些活动长期延续，彼此部分重叠；人们在不同活动之间转换时，活动也可能成为张力的来源。文化历史活动理论已被用于数学教育研究，以考察教师、学生及其互动中的活动（例如，参见 Núñez，2009；Jaworski，2010；Fredriksen 与 Hadjerrouit，2020）。

Engeström 在维果茨基经典的中介三角形基础上，明确加入了塑造活动的集体因素（见图1）。三角形上方的“维果茨基式”部分呈现了主体如何借助文化工具，以客体为目标采取行动。底部一排的节点（规则、社群和劳动分工）体现了 Engeström 所说的活动系统的隐性课程（第112页，Engeström，2015）。这是活动的集体维度：规范对行动的调节、社群对合法性的赋予，以及塑造参与活动的角色与责任分配。这些因素提醒我们，个体行动始终受到社会中介。

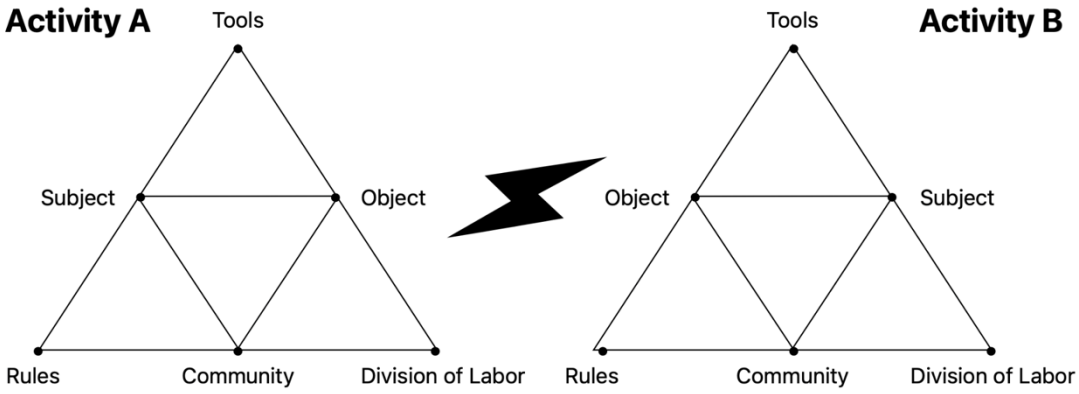

图1：Engeström 第三代文化历史活动理论的分析单位，是一对彼此交织、相互作用且节点部分重叠的活动系统。

活动理论的核心概念是辩证矛盾。这些是活动的内在矛盾，被视为活动的驱动力。它们是张力、冲突、变化和创新的根源。举例来说，Roth 和 Radford（2011）描述了教育的根本矛盾：教师知道学习的客体是什么，学生却永远不知道。这不可避免地造成师生观点之间的不可通约性。这类矛盾是活动中不可消除的特征，而非有待解决的问题。

为分析数学家在研究与教学之间转换时产生的张力，我们借鉴 Engeström（2015）关于相互作用的活动系统的概念。在图1中，我们展示了 Engeström 的分析单位，即两个相互作用且部分重叠的活动系统所构成的一对系统。在我们的研究中，这两个系统分别对应数学家的研究活动和教学活动。

继 Mwanza 和 Engeström（2005）之后，我们将相连的活动三角形的各个节点视为一份核对清单，用来确定应向数据集提出哪些问题，从而识别推动教学与研究活动的深层辩证矛盾。

实践共同体

实践共同体（CoP）补充了文化历史活动理论（CHAT）的系统视角，帮助我们理解嵌入专业社群中的个体如何承接、重新诠释并践行系统性张力（Lave & Wenger，1991；Wenger，1998）。我们的研究受到 Burton（2004）的启发：他运用实践共同体分析了对70位数学家的访谈中的“权力话语”；也受到 Jaworski（2010）的启发：她将实践共同体与文化历史活动理论结合起来，分析自己在大学阶段数学教学中的实践。

实践共同体（CoP）分析中的一个核心构念是合法参与。编程在他们的实践中被定位为充分参与、边缘参与还是边际参与，反映了不同数学家共同体（及其子共同体）如何赋予计算工作以合法性。Burton（2004）阐明了这类判断如何植根于认识论和审美价值，而这些价值塑造着数学家对何为正统数学活动的认知。

实践共同体理论凸显了参与与物化的二元性，共同体借此将抽象观念转化为共享客体，以协商意义。“编程”“纯数学”或“数学工作”等术语都是物化，其含义既反映也塑造共同体规范。要理解“编程不属于数学”这样的论断，就必须关注这些物化范畴如何产生、维持并受到质疑。

最后，实践共同体理论强调，边界是专业化的必然结果。随着数学家不断专攻，物化的含义以及与之相关的合法参与规范都会发生变化。跨越边界，例如在数学的不同子领域之间，或在研究与教学之间，需要重新协商这些含义，这往往会显现出张力，但也能促成新的实践形式。在我们的研究情境中，数学家作为中介者在这些边界和障碍之间周旋，而“编程”则充当一种边界客体，其含义会随情境而变化。

研究问题

在下一节中，我们将通过以人为本的实践共同体视角，把文化历史活动理论对矛盾的系统性探寻转化为可操作的分析框架，以分析数学家如何描述编程在其研究和教学中的作用。

请注意，我们所说的“计算工具”（或简称“计算”），是指依赖于对表达性计算媒介的运用能力来开展数学工作的方式（diSessa，2018年）。这类工作可能涉及实际计算机，也可能不涉及；它也可以表现为受对这类媒介的熟悉程度影响的推理。我们所说的“理论工具”，是指不借助这类媒介的工作方式。与我们的社会文化理论框架一致，我们不把这种区分视为客观的，而是认为它反映了计算媒介如何影响参与者对可能采取何种行动的看法。

这一理论框架引出了以下研究问题。

> 1. 从客体取向（理论性的还是面向现实世界的）和工具取向（理论性的还是计算性的）来理解数学家的陈述，如何帮助我们在一个异质性社群中辨识出连贯的认识论原型？2. 在每种原型中，哪些认识论规则决定了研究活动和教学活动中编程的正当使用？3. 研究和教学在塑造数学家对编程正当性的看法方面有何不同？

方法论

本研究旨在理解数学家对其在研究和教学情境中使用编程的反思。为此，我们开展了半结构式访谈，并借助CHAT和实践共同体（CoP）对他们的叙述进行分析（例如，参见Jaworski，2010）。我们采取解释主义立场，将访谈视为参与者对其认识论取向、身份认同以及对职业规范的理解所作反思的情境化共同建构的叙述。

背景与参与者

这项研究在奥斯陆大学数学与自然科学学院开展。该学院长期以来实行全校范围的政策，在科学、技术、工程和数学

学科中推广编程与科学计算，至今已有 20 多年（见 Malthe-Sørenssen，2016年）。这一背景使我们能够考察数学家在编程得到制度认可并受到公众赞誉的环境中所持的看法。

参与者通过目的性抽样和滚雪球抽样招募，以收集关于编程的多元且相互对照的观点，并确保广泛代表不同的制度背景。我们访谈了十五位数学家，他们处于职业生涯的不同阶段，专长领域也各不相同（纯数学、应用数学、计算数学和数值分析）。

数据收集

访谈于2023年九月至十月间逐一进行，每次持续90分钟至两小时。每次访谈均以半结构化访谈提纲为指导，围绕以下三个方面展开：

- 研究实践
- 教学实践
- 制度背景

随着访谈的推进，访谈提纲保持灵活，使参与者能够进一步阐述此前交流中出现的主题。所有访谈均录音、转录并匿名处理。论文中引用的节选均已译成英文。

访谈由第一作者主持；第一作者是一位活跃的数学研究者，因此参与者可以自由使用专业术语。在讨论证明、建模或计算启发式方法时，这使交流具有更深的认识论和技术层面。 然而，这种内部人士身份也带来了偏见以及过度认同参与者假设的风险。为降低这些风险，我们将访谈视为共同建构的对话，由访谈者和参与者共同表达意义与阐释。第二作者在分析过程中担任外部批判性声音，并帮助抵消潜在的内部人偏见。

分析

遵循 Mwanza 和 Engeström（2005）的做法，我们将 Engeström 扩展活动三角模型的各个节点操作化为一份分析清单，并系统地运用该清单，在转录后的访谈数据中寻找矛盾的表现。为了使这份清单扎根于对人类生活经验的叙述，我们借鉴了受实践共同体启发的话语指标（表 1）。

表 1：作为“做数学”的专业活动——数据中的代表性引文
set.

理论构念	理论构念作为分析视角的关注点及其操作化（运用实践共同体）
客体	作为数学家工作的集体目的和目标。 “数学关乎探索数学上平凡与不可能之间的薄膜。”
主体	数学家行事的身份认同和个人动机。 “我就是喜欢定理。”
规则	学科规范以及开展数学工作的方式的合法性。 “我不认为实现算法属于我的数学工作。”
工具	数学家在工作中使用的有形与无形工具。 “我用计算机对猜想进行数值检验。这让我能够开展理论工作，更多信心和活力。”
社群	参与、合法性与边界的专业格局图。

表 1：作为专业活动的“做数学”及从数据中选取的示例引文 set.	
	“我觉得研究与现实毫无关联的抽象问题既无聊又愚蠢。”
分工 of 劳动	运用能动性，在数学共同体中协商并游刃有余地行动。 “我会用计算来检验猜想，但我的大多数同事不会——即使他们知道他们可能会通过这样做获得一些新的认识。
成果	专业数学实践所产生的有形与符号性成果。 “几乎所有期刊都希望/……/ 有一幅基于真实数据的图示。”

分析以迭代方式展开。在初步分析中，第一作者运用我们对文化历史活动理论的操作化进行框架分析，以整理数据；随后，作者团队持续开展讨论，不断质疑和完善各种解读、主题及逐渐形成的假设。这些解读又通过在研讨会和学术会议上的报告得到进一步巩固（Olsen 等，2025a、2025b）。在这一过程中，我们构建了一个认识论矩阵，用以呈现参与者对客体和工具的取向（见下文表 2）。

我们的认识论矩阵指导了第二阶段的演绎式编码，使我们能够有针对性地比较研究与教学中的对象与工具取向。在反复分析的过程中，作者团队对不断演变的解读和理论承诺进行了批判性审视。必要时，第一作者作为数学家的身份使其能够结合参与者的发表记录和研究成果进行三角验证，以澄清相关解读。为保护匿名性并凸显更广泛的模式，研究发现以原型而非个人为单位进行报告。

研究发现

为回答研究问题，我们首先介绍在分析过程中形成的认识论矩阵。该矩阵呈现了数学家对其工作客体以及他们认为正当的工具所表现出的主导取向。随后，我们借助该矩阵，按照矩阵所代表的四种原型实践，描述参与者如何看待编程在研究和教学中的作用。最后，我们比较这些陈述，考察计算工具在研究和教学中的使用有何差异。

认识论矩阵：四种原型数学实践

在访谈中，参与者在研究客体（目的）以及他们认为适用于研究的工具方面，呈现出系统性的差异。通过迭代分析，我们发现，这些差异可以沿着两个相交的维度加以归纳：

- 他们的研究客体是面向理论问题，还是面向兼具理论问题与现实世界问题的议题；• 他们偏好的工具使用是面向理论工具，还是兼用理论工具与计算工具。

如前所述，我们遵循 diSessa (2018)，将理论工具与计算工具区分开来；后者被理解为运用表达性计算媒介的素养。在谈及认识论规范时，我们与 Weber 等人（2014）的用法一致，指的是支配数学家寻求何种知识，以及他们通过何种方式为这些知识提供依据的规则（CHAT 意义上的规则）。

由此形成的认识论矩阵（表 2）有助于我们识别参与数学活动的四种原型方式。遵循我们的社会文化框架，我们将这些原型视为参与模式——数学家对客体和工具采取的反复出现的取向方式；随着时间推移，这些方式会成为可辨识的实践形式。

几位数学家描述了自己在职业生涯的不同阶段参与了与多个象限相关的实践。然而，我们认为，除一人之外，其余数学家都清楚地表达了一种取向，而这种取向在以下方面始终更为突出：

他们如何描述日常工作、偏好的问题，以及他们对何为“数学”的理解。因此，我们根据访谈中反复提及的主要观点，将其余数学家归入最相符的原型。为使叙述清晰，我们将个别数学家称为“经典纯数学”或“计算应用数学”等；这些标签是分析性构念，并非他们自我认定的身份。它们反映了参与者通过话语如何定位自己的实践，而不是学科类别。

表 2：认识论矩阵（通用版本）

	理论对象取向	实用对象取向
理论 tool 取向	“经典纯数学” 偏好具有理论兴趣的问题，并且希望借助理论工具来研究这些问题。如果工具不起作用，他们往往会改变问题。 频次：*5* 次访谈	“经典应用数学” 偏好理论和现实世界中的问题感兴趣的问题，并希望借助理论工具来解决。如果工具不起作用，他们往往会改变问题。 频次：*2*次访谈
务实型 tool 取向	“计算纯数学” 偏好具有理论兴趣的问题，并且希望运用理论和计算工具来研究这些问题。如果工具不起作用，他们愿意扩充工具箱。 频次：*3* 次访谈	“计算应用数学” 偏好理论和现实世界的问题感兴趣，并希望运用理论和计算工具来研究这些问题。如果工具不起作用，他们准备改进工具箱。 频次：*5* 次访谈

数学家对编程应用的看法

现在，我们运用认识论矩阵来指导分析数学家对于在研究和教学实践中使用编程的认识论观点。这些发现总结于表3。

表3：认识论矩阵（研究与教学实践）		
	理论对象取向	实用对象取向
理论 tool 取向	“经典纯数学” **研究中的编程：** 很少使用或完全不使用计算工具。 不认为编程属于数学工作。 对于“前认识论”层面的工作，适度使用是合理的 。计算输出基本不会出现在 出版物中。 **教学中的编程：** 教学中的编程参与度较低。 实质性且有理论框架支撑的融入被视为 合理，但不应扰乱分析学课程的 身份认同。轻度支持被认为 有所帮助，但其认知价值较低 可见性。编程在 评估。	“经典应用数学” **研究中的编程：** 使用计算的熟练程序员 将其作为概念验证来 理论结果。不认为 编程属于数学工作。 轻度使用适用于“前 认识论”工作。计算输出 在出版物中具有很高的可见度。 **教学中的编程：** 对编程的参与度较低 在专业方法课程之外。 要实现真正的融合，需要实质性的 应用和理论框架。轻度支持 被认为有帮助，但其 认识论可见性较低。编程的 在评估中的可见度。
务实 tool 取向	“计算纯数学” **研究中的编程：** 对将编程作为.....的参与度虽高但各不相同 合法的数学工作。所有人都讲述了 自己对.....的认识论影响着迷 计算。计算产物具有充分的 与理论结果并列的合法性。 **教学中的编程：** 对教学中的编程参与度不高， 但也有一些值得注意的例外。合法性与 人们认为其对理解理论 概念的影响有关。轻度支持的认识论 可见性较低。编程在 评估。	“计算应用数学” **研究中的编程：** 高度投入于将编程视为 现实问题解析工作的合法伙伴。 计算产物 与理论结果并列，具有充分的合法性。 结果。 **教学中的编程：** 广泛参与编程，但 也存在一些值得注意的例外。 编程被视为具有内在 正当性，但这种看法受到人们所认为的 其对理论理解的影响的制约 概念。编程的可见度较低 在评估中。

经典纯数学家

研究实践

经典纯数学家将自己的研究描述为以抽象数学对象为中心，这些对象同时是研究对象、探究工具和研究成果。正如一位资深数学家所描述的那样，这种紧密的契合几乎没有给编程留下空间：

> “.....我的研究中几乎没有什么会用到编程，或者说，真的会用到任何计算机工具。[.....] ‘我常说，数学上有趣的东西，就处在平凡与不可能之间那层极薄的膜上——只有这部分问题能用数学解答。’ [.....] 毕竟，你得寻找自己认为能用现有工具解决的问题。而且，也要能借助这些问题来稍微扩充一下工具箱，对吧？”

尽管经典纯数学家表示自己很少参与编程，但他们承认使用编程具有认识论潜力。正如一位数学家所说：“大家都认为，让人用[数值方式] 检验这类事情是件好事。但这并非关键；关键在于你能证明的定理.....”

尽管所有人都对编程有所了解，而且几乎所有人都尝试过一些探索性的数值工作，但只有一人表示，他会定期将其作为解析工作的“轻量级”辅助——偶尔检验猜想和例子。他不太愿意把这称为编程，因为他使用的始终只是简短的代码片段。

编程的成果只会在出版物中作为支持猜想结果的依据而显现。有趣的是，他们似乎很少与同事谈论计算。事实上，有些人甚至说不清长期合作者是否会编程：“我肯定他会编程——他什么都会——但我不觉得他会费心去做这件事。”

教学实践

经典纯数学家在教学中很少涉及编程，而是专注于抽象数学对象，并使用定理和证明等解析工具进行研究。他们通常赞成学生在学习期间学习编程，但不认为理论数学课程是教授编程的合适场所。

他们认为，要将编程合理地融入基础课程，就必须实质性地使用编程，并为其建立理论框架。一位没有在课程中融入计算的数学家承认，通过一个简单的 for 循环可以学到很多东西，这“令人惊叹”。然而，他不认同将轻量介入方式作为有效融合的基础：“我有点不好意思，因为我觉得我们在这里做的编程就是 [...] 基本上是一个循环，之前做些设置，之后做些收尾。[...] 这并不能真正培养出扎实的编程能力。”相反，他认为有必要将数值数学中的理论引入经典理论课程，以避免对计算的使用流于“琐碎”。

然而，其他参与者批评这种“强硬”的做法扰乱了传统的课程特性。一位参与者代表普遍观点，批评道：“我觉得，像我这样的学习较慢的学生，[...] 更受益于内容连贯、易于理解的课程 [...] ，而不是这种跨界做法；它似乎只对那些已经什么都懂的人有帮助。”

尽管经典纯数学家并不认为这具有实质性意义，但他们对以“轻触式”的编程辅助分析学习持积极态度。一位参与者指出，经过少量数值探索的“预热”后，“基础较弱的学生”似乎表现得更好。另一位则谈到，仅仅引入编程这一概念，就有助于学生跨越学科边界建立联系：“效果简直棒极了”。此外，那些没有参与编程的人也表示，他们“基本上支持”让学生“就在一旁编写一点代码，你知道，只是为了培养直觉”。至少在学生已经会编程的情况下如此。不过，正如一位参与者补充的那样，“也许只要让[学生] 坐在那里，用笔和纸就够了”。

在这一原型中，编程在评估中几乎隐而不见。一位受访者说：“不过，我们希望他们能读懂一些简单的 Matlab 输出。但即使不会，他们也能解题，只不过可能得用手工[计算]一点。”

经典应用数学家

研究实践

经典应用数学家将自己的研究实践描述为主要专注于数学模型数值算法的理论工作，而这些模型与现实世界有一定关联。尽管他们是熟练的程序员，会在工作中明显地运用计算，但他们不太愿意将其描述为数学实践的一部分。正如一位受访者所解释的：

> “研究工作包括开发数值方法，然后运用泛函分析和其他工具分析这些方法。编程实现这些方法并不是最重要的[...] 这

主要是为了说明这并非空谈。[...] 我认为，数值方法的理论及其分析才是真正属于数学的部分。”

尽管一些经典应用数学家起初淡化了编程的认识论作用，但在更详细地讨论编程的用途时，他们的表述显得更为细致。他们都表示，会将编程作为解析工作的辅助性认识论支持，不过正如一位受访者所说，这种情况“并不常见”。与此同时，他们强调，计算的作用不只是提供表面上的说明——事实上，省略计算反而会显得可疑：“在我看来，这似乎表明他们有些事情想要隐瞒，说明这种方法在实践中其实行不通。”

有趣的是，尽管有人明确表示，出版物必须以理论为核心，并辅以计算验证，但这一规范也有同样明确的例外。一端是，对于那些难以（甚至根本无法）实现的数值方法，其相关结果无需计算支持。一位受访者说：“[有时] 我们懒得去看全离散情形”。另一端是，在几乎没有或完全没有已知理论的情况下研究物理模型时，他们认为仅基于计算结果发表论文也合乎规范——他们将其称为“物理学家的方法”。由此，他们指出自己的实践与经典纯数学家和计算应用数学家的实践有交集。

教学实践

经典应用数学家在基础课程中对编程的参与程度，同经典纯数学家一样低。他们觉得教编程很困难，因此更倾向于将其局限在专业方法课程中。

他们认为，要在学习项目中合理地融入编程，就必须以数值理论为理论框架；但他们批评那些通过把数值数学理论移至其他课程来融入编程的做法，认为这种做法很随意：“在我看来，似乎有人让一个对编程或数值分析一窍不通的人，在课程中插入一些编程内容 [.....] 感觉有点像硬塞进去的。”

当被问及较为宽松的做法时，有些人表示怀疑：“我认为，必须先真正理解数学理论，才能编程。[.....] 反过来学肯定不容易.....也就是在某种程度上更多地依赖计算机上的数学计算，并理解其理论。”这与他们从研究中形成的看法相呼应：编程主要是一种工具，用来阐释已经获得的数学结果，而不是用来学习数学的工具。

与经典纯数学家一样，应用数学家也边缘化了编程在评估中的作用；他们甚至会说，如果学生能完成课程作业中的分析部分，就算是放了他们一马。

计算纯数学家

研究实践

计算纯数学家将自己的实践描述为以抽象数学对象为中心，并通过结合分析工具与计算工具来研究这些对象。他们是群体中异质性最大的一群，但共同点是，他们都讲述了自己如何着迷于计算工具所带来的可能性：借助这些工具，他们能够对纯数学对象获得洞见，其深度远超纸笔计算所能达到的程度。正如其中一人所说：“能够使用这些专用程序来创建、计算并理解更多例子，对我来说意义重大。”

计算纯数学家以多种方式运用编程，从轻量级的数值探索到深入的算法实现。有些人像经典纯数学领域的同事一样进行计算——检验猜想或生成例子——但他们会公开这样做，并在出版物中附上符号脚本或数值脚本。正如其中一人直截了当地说：“你应该始终解释自己做了什么。”另一些人则进一步运用编程，通过实现算法来探究抽象对象的结构，并将计算视为一种互补且正当的洞见来源。

他们超越了经典纯数学家所提供的轻量级计算支持，因为他们都是编程技能高超的程序员，必要时能够利用超级计算机。少数人认为，计算提供了全新的认知方式，并主张生成式人工智能等发展揭示了纯粹解析方法的局限，使解析与计算方法的结合不可或缺。否则，有人指出，理论数学就有被简化为“哲学”的风险。

教学实践

尽管计算纯数学家高度认可利用计算来学习数学的合法性，但他们在教学中的编程参与程度并不均衡。有趣的是，他们都将学习分析数学时教学法方面的考虑作为其教学中高程度或低程度参与编程的理由。

他们认为，在教学中使用编程是否合法，取决于他们认为什么是学习解析概念所必需的。例如，其中一人解释自己在基础课程中很少使用编程，是因为“这些题目可以手算，所以我没有告诉他们[使用编程]”。在更高级的课程中，他则描述了他们如何展示计算工具，让学生借助这些工具以计算方式探索那些仅靠解析方法难以处理或无法处理的例子。

这也表明，与其他原型一样，轻度使用编程来支持解析概念的学习并不显眼。另一项证据来自另一位受访者的观察：在解析方面有困难的学生，在同时被要求使用计算工具时会更加吃力。因此，他认为自己的职责是开设让学生能够专注于解析材料的课程。他说：“我的教学始终[.....] 以理论部分为重点。这并不是要[排斥] 另一面，而是要与计算相辅相成。我认为两部分都学到很重要。”不过，也有人认为，最大限度地将计算融入教学，可能有助于理论学习吃力的学生：“我认为.....对于编程能力相当强的人来说，就他们学习理论的能力而言，我觉得编程会有所帮助。他们更容易看明白；通过代码，他们比通过数学证明更能理解[理论] 。”然而，他并未真正贯彻这一理念。他表示，自己很少在讲座中提及编程，而是主要将其作为课程材料中的一种补充视角。他说：“在我们所有的[数学] 课程中，不应该让学生接触太多代码，但这对他们是有益的。”

在这一原型中，编程在评估中几乎不可见。有人说：“唯一要求他们编程的地方，就是家庭作业。”

计算应用数学家

研究实践

计算应用数学家将自己的研究实践描述为：通过解析工具、计算实现以及与外部合作伙伴的合作相互交织，开发并实现现实世界现象的数学模型。他们显然是熟练的程序员，并且真心热衷于在现实世界数据、计算与理论数学的交界处开展工作。

> “可以说，正是在那个交汇点上工作让我感到愉快——在那里，你能对问题的样貌形成某种几何直觉[.....] 然后，基于这种理解，运用恰当的数值操作，将问题推进到足够深入的程度，从而在特定情境下找到一个具体答案。”

他们认为，计算与解析工具可以构成一种正当的伙伴关系，彼此相辅相成。有人说：“对我来说，并不存在明确的区分，仿佛有些问题需要分析，有些需要建模，还有些需要编程。”进一步说明这一点时，有人谈到一些同事：他们在没有考虑计算的情况下，对数值算法做了理论工作：“可以说，他得出的公式，就其适用范围而言，完全无懈可击，但几乎无法用于计算。”

编程在他们的工作成果中具有很高的可见度，其中甚至包括在商业层面开发和维护软件。

教学实践

总体而言，计算应用数学家在教学中对编程的参与程度远高于其他原型，其中大多数会在一定程度上融入计算。然而，他们对计算的参与似乎受到对学生分析能力的担忧所制约。其中一人解释了为什么他在所教授的一门基础课程中不使用任何编程：“对于我所教授的第一年课程来说，学生能否顺利完成课程的挑战并不在于他们使用计算机的能力，而在于他们理解[分析数学]的能力。”

几乎所有人都很快就能描述他们如何以各种方式在学习理论中运用轻度编程支持：“里面有一些 for 循环；这不是高级编程。我们使用它，既是为了让他们真正得出结果，也是因为稍微编一点程有助于他们更好地理解这些方法。”然而，有些人对此感到沮丧，因为他们认为学生获得的实践经验仅限于这种轻度支持，并指出学生的学习计划中需要

更多实质性的计算项目：“我是说，谁都能做到。如果你想做好计算，就必须做一些超出[使用代码片段]的事情。比如……你得亲自动手编一些程序……”

纵观这些原型，一种辅助编程或评估并几乎无足轻重。有人说：“我们认为编程和编

不是我们认为测试中必不可少的内容。”

研究实践与教学实践的比较

在四种原型中，数学家似乎都将相同的深层认识论规则从研究延续到教学，表现出不同情境之间的认识论连续性。纯数学家坚持“分析排他性”；应用数学家在坚持分析排他性的同时做出实用主义让步；计算应用数学家依赖分析与计算的协作；计算纯数学家则将计算视为正当的分析方法。这些规则构成其研究实践的根基，并在他们转向教学时依然保持不变。

与此同时，Burton（2004）所描述的从研究到教学的对象—结果转变，重塑了编程的使用方式。教学侧重于数学既有的成果——定义、定理和分析技术——而非产生这些知识的实践。由于基础课程突出分析内容和传统评估形式，计算的认识论作用往往不那么显而易见。即使是研究中高度依赖编程的数学家，也会限制编程在教学中的使用，理由是学生的分析基础较弱，且需要维护课程特性。

这些动态引发了编程整合中强势整合方式与轻量整合方式之间反复出现的张力。强势整合方式试图以理论上合理的方式进行整合——通常是将数值分析内容或编程教学引入纯数学课程——但这被认为会扰乱课程特性，并使分析能力不强的学生处于不利地位。相比之下，轻量的计算应用（例如探索性脚本、数值检验、可视化，或不涉及实际代码的算法化框架）普遍被视为有助于教学，尤其对分析能力较弱的学生而言。然而，这类实践仍处于边缘地位，因为它们很少被认可为“真正的”编程，因此认识论可见性较低。

最后，编程在评估中仍基本不可见。无论属于哪种原型，数学家都避免评估编程，认为编程难以评分，或与“核心”数学能力关系不大。即使是那些在研究中高度依赖计算的数学家，有时也会忽略学生未完成的计算工作，只要学生展现出分析熟练度。我们将此解读为一个强烈信号：在教学中，所有原型都认为计算的认识论合法性较低；这与许多参与者的研究出版物中计算贡献高度可见形成了鲜明对比。

讨论

概要

在本研究中，我们借助文化历史活动理论和实践共同体理论，分析了对15位数学家的访谈，探究他们如何阐述编程在其研究和教学中的正当性。 现在，我们将结合理论框架部分末尾提出的三个研究问题，对分析结果展开讨论。

研究问题1： 从客体取向（理论型与现实世界型）和工具取向（理论型与计算型）的角度来理解数学家的叙述，如何帮助我们在一个原本异质多元的社群中辨识出连贯的认识论原型？

尽管长期以来都得到制度支持，数学家对于将编程定位为认识论工具的看法仍存在显著差异。我们的分析表明，参与者的话语中的这种差异并非纯粹出于个人特性，而是呈现出一定规律。为此，我们依据数学家的客体取向和工具取向对其进行分类，归纳出四种认识论原型：经典纯数学、经典应用数学、计算应用数学和计算纯数学。在每种原型中，数学家对何为有意义的问题、可接受的工具和正当成果都表达了内在一致的看法，从而使我们能够梳理既有文献所述的显著异质性。我们发现，与经典纯数学家和计算应用数学家相比，纯应用数学家和计算纯数学家表现出更多跨越边界的模式。

研究问题*2*：在每种原型中，哪些认识论规则分别规范着编程在研究活动和教学活动中的正当使用？

在研究中，所有原型都遵循某种形式的分析工具与计算工具协同使用方式——从严格排斥分析以外的方法，到分析与计算的协作。这些取向决定了编程会被视为探索性的“前认识论”支架、数值验证，还是建立合法洞见的工具。在教学中，数学家的认识论规则在很大程度上延续了研究中的规则；然而，由于基础课程着重呈现既定知识的成果，而非塑造研究的探究过程，计算工作在认识论上的可见性明显较低。这种从客体到成果的转变系统性地限制了编程在教学中所能发挥的作用——即使对那些在职业工作中依赖计算的人来说也是如此。

研究问题*3*：研究与教学在塑造数学家对编程正当性的看法方面有何不同？

一项核心发现是，对计算采取强势介入与轻量整合方式之间存在张力。强势整合是指在传统上纯粹以分析为主的课程中加入数值分析理论；这种做法赋予计算认识论合法性，却打破了既有的课程特性，而且往往偏重分析能力强的学生。轻量编程则使用少量代码片段来支持概念理解，教学上更容易推广，但大多数数学家并不认为这属于“真正的编程”，使其在认识论上处于边缘地位。在各种原型中，编程基本没有出现在评估中：即使在研究中高度依赖计算的数学家，也避免将其纳入评估，这表明他们对编程在教学中的认识论地位抱有矛盾看法。

对教育的贡献与启示

我们为 Broley 等（2018）提出的数学家在研究与教学中使用编程的地位差距提供了一种新的视角：这一差距反映的不仅是文化态度或制度约束，也反映了根植于数学家偏好的推理方式中的、更深层的认识论连续性。数学家将研究中的规范带入教学，但由于教学着重呈现数学的成果，并依赖分析性评估，编程的认识论可见性便会降低。

同样，我们的结果也表明，可以重新诠释 Burton（2004）对研究—教学差距的解释。与 Burton 一样，我们发现，教学的客体被视为研究的成果，这使得研究中至关重要的探索性实践受到边缘化。但 Burton 将实践上的研究—教学差距归因于数学家践行不同的认识论（采用她较为宽泛的社会学意义上的认识论）；而我们的发现表明，若采用 Weber 等人（2014）较狭义的认识论概念来考察，数学家在不同实践中展现出的认识论连续性与对象—结果转变相结合，便可解释这一差距。鉴于 Weber 等（2020）指出的方法论挑战，我们认为，突出此类连续性的分析框架或能为未来研究数学家的实践提供一条富有成效的路径。

此外，运用认识论矩阵有助于我们通过数学家的对象与工具取向来刻画他们，并观察“编程”等概念如何在不同实践边界上以不同方式被具体化，从而处理 Weber 等（2020）提出的一些方法论问题。这表明，认识论取向或可成为未来研究数学家专业实践与教学实践的有效分析视角。

鉴于数学家如何阐述编程在其专业工作中的正当用途，教育方面的一项实际启示是，应明确认可轻量级认识论编程的正当性。让其目的清晰可见，阐明它如何支持分析性理解，并使适度的评估与这些目标保持一致，或有助于缩小数学家研究实践与教学之间的差距。

局限性与未来研究

与我们的社会文化立场和解释主义立场一致，我们认为研究结果是作者团队与参与者共同建构的、依赖于具体情境的产物。

访谈者作为数学研究者的内部人士立场，促成了深入的技术对话，但也带来了过度认同参与者假设的风险。我们通过一段较长时间的分析来降低这一风险，在此期间，我们与数学和物理教育研究者以及职业数学家分享并讨论阶段性结果，从而促进解释性三角互证。

我们承认，我们的数据收集于2023年秋季，并在随后两年期间进行分析，因此不可避免地受到这样一个时代背景的影响：大语言模型正在迅速发展，并开始影响编程实践。这与其说是一项局限，不如说为未来研究提供了机会：几位数学家已经在认识论层面对人工智能工具进行定位，为我们的框架提供了自然的延伸。

我们的研究发现指向了若干值得进一步探究的方向。首先，若在其他场所采用相同的分析方法，将有助于揭示我们的研究发现如何超越本研究的具体情境，以及如何为其增添更多层次。其次，我们的结果引发了一些问题：学科边界如何塑造人们赋予编程的意义；跨界者——例如经典应用数学家和计算纯数学家——又如何促进创新与变化。第三，本文着重分析了认识论规范与信念。未来研究若能更详细地探讨受访数学家在研究和教学中如何以及为何使用编程，将会很有意义。第四，尽管制度因素似乎会影响数学家是否采用编程的决定，但数学家似乎拥有很大程度的操作能动性。更深入地了解制度压力与个体能动性之间的相互作用，将有助于推动涉及使用计算的改革。第五，认识论矩阵帮助我们解读数学家对将编程作为学习工具的态度。未来研究可以探讨，该矩阵是否也有助于洞察数学家在认识论层面如何定位生成式人工智能（参见 Fraser等人，2024a、2024b）。

利益冲突 **：**代表全体作者，通讯作者声明不存在利益冲突。

参考文献

Broley, L., Caron, F., & Saint-Aubin, Y.（2018年）。数学研究与大学数学教育中的编程层次。*International Journal of Research in Undergraduate Mathematics Education*，*4*，38-55。http://dx.doi.org/10.1007/s40753-017-0066-1

Burton, L. L.（2004年）。《作为探究者的数学家：学习数学学习》（第*34*卷）。Springer Science & Business Media。

Buteau, C., Jarvis, D., & Lavicza, Z.（2014年）。加拿大数学家对计算机代数系统（计算机代数系统）的整合：全国调查结果。《Canadian Journal of Science, Mathematics, & Technology Education》，14(1)，1–23。http://dx.doi.org/10.1080/14926156.2014.874614

Darragh, L.（2022年）。《跨越鸿沟的中介：参与教育的数学家的观点》。《数学行为杂志》，*67*，100989。https://doi.org/10.1016/j.jmathb.2022.100989

DeJarnette, A. F.（2019年）。学生在数学中使用编程环境时遇到的符号和图表方面的困难。《数学教育中的数字体验》，*5*(1)，36–58。 https://doi.org/10.1007/s40751-018- 0044-5

DiSessa, A. A.（2000年）。改变思维：计算机、学习与读写能力。麻省理工学院出版社。

diSessa, A. A.（2018年）。计算素养与关于数学教育中计算机的“大局观”。 数学思维与学习，*20(1)*，3–31。http://dx.doi.org/10.1080/10986065.2018.1403544

Engeström, Y.（2015）。通过扩展而学习：一种发展研究的活动理论方法 . 剑桥大学出版社。

Fraser, M., Granville, A., Harris, M. H., McLarty, C., Riehl, E., & Venkatesh, A.（2024a）。机器会改变数学吗？美国数学会通报（新系列），*61(2)*，201–202。https://doi.org/10.1090/bull/1836

Fraser, M., Granville, A., Harris, M. H., McLarty, C., Riehl, E., & Venkatesh, A.（2024b）。机器会改变数学吗？美国数学会通报（新系列），*61(3)*，373–374。https://doi.org/10.1090/bull/1842

Helge Fredriksen & Said Hadjerrouit（2020年）大学阶段翻转数学课堂中矛盾的活动理论视角，《国际科学与技术数学教育杂志》，*51(4)*，520-541，https://doi.org/10.1080/0020739X.2019.1591533

Hamami, Y., & Morris, R. L.（2020年）。数学实践哲学：数学教育者入门指南。ZDM，52(6)，1113-1126。https://doi.org/10.1007/s11858-020-01159-5 Hersh, R.（1997）。数学究竟是什么？牛津大学出版社。Jaworski, B.（2010）。大学数学教学实践：实践共同体或活动系统中的中介性探究。CERME 6–工作组 9，1585–1594。Kotsopoulos, D., Floyd, L., Khan, S. 等（2017）。计算思维教学框架。Digit Exp Math Educ 3，154–171 https://doi.org/10.1007/s40751-017-0031-2 Lave, J., & Wenger, E.（1991）。情境学习：合法的边缘性参与。剑桥大学出版社。Lockwood, E., DeJarnette, A. F., & Thomas, M.（2019）。作为数学学科实践的计算。Journal of Mathematical Behavior，54。https://doi.org/10.1016/j.jmathb.2019.01.004 Lockwood, E., De Chenne, A. 通过在 Python 中使用循环和条件语句丰富学生的组合推理。Int. J. Res. Undergrad. Math. Ed. 6，303–346（2020年）。https://doi.org/10.1007/s40753-019-00108-2 Lockwood, E., & Mørken, K.（2021）。呼吁开展研究，探索计算与数学思维及本科数学教育研究中的活动之间的关系。International Journal of Research in Undergraduate Mathematics Education，7(3)，404-416。https://doi.org/10.1007/s40753-020-00129-2 Lovelace, A.（无日期）。计算机，增强。Lapham's Quarterly。检索于2025年4月5日，网址：https://www.laphamsquarterly.org/technology/computer-enhance Malthe-Sørenssen, A., Hjorth-Jensen, M., Langtangen, H. P., & Mørken, K.（2016年）。将计算融入物理学教学。Uniped，38。取自 http://hplgit.github.io/cse-physics/doc/pub/uniped15.html Moschkovich, J.（2007）。考察数学话语实践。For the Learning of Mathematics，27(1)，24–30。Mwanza, D., & Engeström, Y.（2005）。管理电子学习环境中的内容。British Journal of Educational Technology，36(3)，453–463。https://doi.org/10.1111/j.1467-8535.2005.00479.x Nardi, E.（2008）。数学家群体中的数学：大学阶段数学的教学与学习（第3卷）。Springer Science & Business Media。Núñez, I.（2009）。矛盾作为变化的源泉：活动理论及其在数学教育中对活动系统的运用之文献综述。Educate，9(3)，7-20。Olsen, J. F., Lockwood, E., & Odden, T. O. B.（2025a）。介于琐碎与不可能之间：将文化历史活动理论作为分析视角，研究编程在数学研究中的作用。载于欧洲数学教育研究学会第十四届大会论文集（CERME14）（第20号）。Olsen, J. F., Odden, T. O. B., & Lockwood, E.（2025b）。探究数学家在教学和研究中使用编程的差异。载于 S. Cook, B.P. Katz, & K. Melhuish（编），第27届本科数学教育研究年会论文集（第1102–1107页）。SIGMAA on RUME。弗吉尼亚州亚历山大。Papert, S.（1980）。Mindstorms：儿童、计算机与强大的思想。Basic Books：纽约，1980年。230页。Polya, G.（2014年）。怎样解题。普林斯顿大学出版社。Roth, W. M., & Radford, L.（2011）。数学教学与学习的文化历史视角（第2卷）。Springer Science & Business Media。Rasmussen, C., Wawro, M., & Zandieh, M.（2015）。考察个体与集体数学进步。Educational Studies in Mathematics，88，259–281。http://dx.doi.org/10.1007/s10649-014-9583-x Weber, K., Dawkins, P., & Mejía-Ramos, J. P.（2020）。数学教育研究中数学实践与数学教学法之间的关系。ZDM，52(6)，1063-1074。http://dx.doi.org/10.1007/s11858-020-01173-7

Weber, K., Inglis, M., & Mejia-Ramos, J. P. (2014). 数学家如何获得确信：对数学教学和认识论认知研究的启示。《教育心理学家》，*49*(1), 36-58页。https://doi.org/10.1080/00461520.2013.865527Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). 为数学和科学课堂界定计算思维。《科学教育与技术杂志》，*25(1)*，127-147页。http://dx.doi.org/10.1007/s10956-015-9581-5Wenger, E. (1998). 《实践社群：学习、意义与身份》，英国剑桥：剑桥大学出版社。Wing, J. M. (2006). 计算思维。《*ACM*通讯》，*49(3)*，33-35页。http://dx.doi.org/10.1145/1118178.1118215