GEOMCP：人工智能辅助解析岩土工程的可信框架

预印本

Yared W.Bekele岩石与土力学组，
SINTEF Community，挪威特隆赫姆
yared.bekele@sintef.no

arXiv:2603.01022v1 [cs.CE] 1 Mar 2026

摘要

分析方法是岩土工程实践的基础，但其实现仍分散于容易出错的电子表格和不透明的专有软件中。大语言模型（LLM）为简化工程工作流程带来了变革性潜力，但其统计特性与安全关键计算所要求的严格确定性存在根本冲突。它们容易虚构公式、误解单位，或在不同会话间改变方法，由此造成了严重的信任差距。本文介绍 GeoMCP，旨在弥合这一差距；其核心理念是：工程方法应以结构化数据表示，而不是嵌入代码。GeoMCP 将分析方法封装为“方法卡片”，即用于定义公式、单位、适用范围限制和文献引文的声明式 JSON 文件。受限符号引擎通过验证量纲一致性来执行这些卡片，而结构化的“智能体技能”则引导大语言模型运用工程判断并统筹分析。通过模型上下文协议（MCP）公开这些经过验证的能力，GeoMCP 将人工智能从不可靠的计算器转变为智能协调者。通过官方 JRC《欧洲规范7》算例的验证，该框架展现了与传统方法相当的计算结果，同时确保数学过程完全透明。最终，GeoMCP 为行业从彼此孤立的传统软件转向互操作、面向人工智能的生态系统提供了蓝图，使工程师能够利用现代人工智能，而不必放弃专业责任。

关键词 岩土工程、分析方法、AI辅助工程、模型上下文协议、知识表示、符号计算

1 引言

分析方法，例如太沙基承载力理论和朗肯土压力方程，是岩土工程实践的基础。然而，用于应用这些方法的工具仍然彼此割裂，而且往往不透明。电子表格虽然无处不在，却容易出错；研究表明，其中相当一部分存在难以从基于单元格的结构中发现的错误 [1]。在现有软件和脚本中，分析公式往往嵌入在用户界面或实现细节中，这使得对方法本身进行验证和引用比应有的更加困难。

大语言模型（LLM）的出现加剧了这一挑战。大语言模型能够理解技术问题并生成代码，因此 AI 助手有望简化工程工作流程。然而，大语言模型不适用于安全关键计算：它们可能会虚构公式、误解单位，并混淆不同方法。例如，当被要求计算迈耶霍夫承载力时，大语言模型可能会生成使用 Terzaghi$N_\gamma = 2(N_q + 1)\tan\ \phi'$ 表达式、而非迈耶霍夫自身经深度和形状修正的公式的代码，得出看似合理却并非预期的结果，而且不会指出错误。因此，核心挑战并不在于大语言模型能否帮助工程师——显然可以——而在于如何利用其能力，同时保持工程实践所要求的透明性、可靠性和可审计性。我们认为，解决之道是让大语言模型能够使用经过筛选和验证的计算工具，而不是让它们凭记忆生成计算结果。

本文介绍 GeoMCP，这是一个面向 AI 辅助工程的分析型岩土计算框架。GeoMCP 基于一个关键洞见：岩土工程方法应表示为结构化数据，而非嵌入代码中。每种分析方法都以明确且便于阅读的“方法卡片”呈现，即一个 JSON 文件，其中声明变量、单位、方程、假设、适用范围限制和文献来源。基于 SymPy（符号数学）和 Pint（物理单位）构建的受约束评估引擎可执行方法卡片，并确保量纲一致性经过验证且计算过程完全透明。通过模型上下文协议（MCP） [2],公开这些功能后，GeoMCP 使 AI 助手能够执行可靠计算，同时确保结果可追溯至权威来源。

这种以数据为中心的方法改变了人工智能在岩土工程中的作用。AI助手借助 GeoMCP，不再生成可能有误的计算结果，而是成为经过验证的方法的接口。工程师可以用自然语言提出关于其问题的问题，并附上支持性文档和数据；随后，助手会调用 GeoMCP 工具，不仅给出答案，还会提供完整的计算轨迹，展示输入、中间步骤以及每个公式的文献来源。

本文作出以下贡献：

1. 提出一种声明式 JSON 模式，将岩土工程分析方法表示为结构化、可引用的数据对象，其中包含公式及其工程背景——假设、适用范围限制和文献引文。2. 构建安全、确定性的评估引擎，提供沙盒符号求值（SymPy）、自动量纲分析（Pint），以及用于解决岩土分析中常见循环依赖的迭代求解器。3. 与模型上下文协议（MCP）和智能体技能标准集成，使整理后的方法能够通过标准化接口供 AI 助手及其他客户端应用程序使用。4. 针对浅基础承载力，依据 JRC（联合研究中心）欧洲规范7官方算例进行验证。5. 建立不断扩充的方法卡片目录，涵盖常见岩土分析，并讨论其对 AI辅助工程实践的更广泛影响。

本文其余部分组织如下。第2节综述相关工作。第3节介绍设计理念与系统架构。第4节详述方法卡片、评估引擎、智能体技能和MCP接口的实现。第5节依据JRC《欧洲规范7》官方算例对GeoMCP进行验证。第6节讨论其优势、局限性以及对AI辅助工程的启示。第7节总结并展望未来方向。

2 相关工作

GeoMCP借鉴并贡献于三个相互关联的研究领域：岩土工程计算工具、土木工程中的人工智能应用，以及工程设计的知识表示框架。本节将我们的工作置于这些背景之中，并阐明GeoMCP如何弥补现有方法的不足。

2.1 岩土软件与计算工具

用于分析性岩土计算的计算工具已从简单的图表查取和手工计算，发展到在图形用户界面背后实现承载力理论、沉降分析、土压力计算和设计规范校核的复杂商业软件套件。这些工具在实践中已不可或缺，它们提供丰富的材料库、自动报告生成以及经过充分验证的实现，工程师有理由在常规设计工作中予以信赖。然而，仍存在两项局限。第一，所采用的具体分析公式通常难以在软件内部追溯至其文献来源：使用商业工具计算承载力的工程师，可能无法在工具内确认究竟正在计算哪一已发表表达式，或应用了形状系数的哪一变体。第二，这些工具是为通过专用界面由人直接操作而设计的，而非面向程序化或AI辅助访问。随着大语言模型和AI 助手在工程工作流程中日益重要，缺乏面向底层分析方法的机器可读、基于标准的接口，正成为一个日益突出的缺口。

除商业软件外，日常分析性岩土工程工作中有很大一部分是借助各公司或项目团队内部开发的内部电子表格、VBA 宏和定制脚本完成的。这些内部工具高度定制，对其所构建的任务往往效率很高，但它们加剧了上述局限性。公式嵌入在单元格引用或过程式代码中，与已发表的资料来源几乎没有或完全没有关联，而这些资料来源正是它们

所实施的；版本控制很少做到系统化；单位处理取决于原作者的严谨程度；由此产生的计算难以审计、复现，或在创建它们的团队之外共享 [1]。与商业软件包一样，这些工具也并非为集成到自动化或人工智能辅助工作流中而设计。

在开源领域，有多个 Python 库致力于岩土工程分析的不同方面。groundhog [3] 是一个涵盖范围广泛的库，涉及场地勘察、桩基设计、浅基础承载力、固结、土压力和土动力学，并内置了根据原始文献得出的参数范围验证功能。geolysis [4]专注于承载力、土分类和标准贯入试验分析，而 pySlope [5] 则专为使用 Bishop条分法进行边坡稳定性评估而设计。这些库使其实现过程公开且可修改，但它们具有一个共同的架构特点：公式直接编码在程序代码中，计算逻辑与实现细节混杂在一起。要验证某项实现是否符合已发表的方法，需要阅读 Python 源代码；要修改或扩展某种方法，则需要编程专业知识；而且无法直接支持引用计算中使用的具体公式。此外，这些库面向程序化使用，而非与 AI 助手或其他客户端应用程序集成。

GeoMCP 通过将方法定义与评估逻辑分离来弥补这些不足。每种分析方法都被封装为独立的 JSON“方法卡片”，其中声明了方程、变量、单位，以及至关重要的公式来源文献，使可追溯性成为一项基本属性，而非事后补充。由于这些卡片是纯数据文件，工程师只需运用领域知识并使用文本编辑器，无需阅读源代码，即可检查、验证或扩展某种方法。随后，MCP接口通过标准化协议提供这些经过整理的方法，使 AI 助手和其他客户端应用程序能够以编程方式发现并调用经过验证的计算功能；现有的商业、内部开发或开源工具均不具备这一能力。

2.2 基于知识的工程与形式化表征法

以机器可解释的格式捕获工程知识这一概念，在基于知识的工程（KBE）领域有着悠久的历史。早期专家系统（例如20世纪80年代为结构分析与设计开发的系统）曾尝试将人类专业知识编码为基于规则的系统。这些系统虽然在技术上令人瞩目，却难以维护、扩展，也难以融入不断发展的工程实践。知识获取瓶颈——即从领域专家处提取并形式化专业知识，再用专门的知识表示语言对其进行编码所面临的挑战——限制了这类系统的应用。此外，这些系统往往会变得脆弱，无法处理训练范围之外的情况；随着规范和标准不断演变，更新系统也需要投入大量精力。

较新的基于知识的工程研究主要关注用于表示工程知识的本体和语义框架。工业基础类（IFC）、建筑拓扑本体（BOT）和网络本体语言（OWL）等标准提供了正式词汇，用于描述物理实体及其属性和相互关系。这些框架主要针对结构和拓扑描述，而非计算程序。它们擅长表示“存在什么”（地基具有几何形状，土壤具有属性，各组件彼此相关），却不擅长表示“如何”计算派生量（例如如何使用特定分析方法，根据土壤参数计算承载力）。这一差距促使人们开发一种互补的表示方法，专门用于工程计算。

GeoMCP 的方法卡片概念在结构上与 Minsky 基于框架的知识表示相似 [6], ，其中结构化对象通过类型化槽位封装相关数据。每张方法卡片都是一条结构化记录，包含变量（含类型、单位和角色）、方程（以符号形式表示）、适用条件和文献来源等槽位。然而，与使用专用语言嵌入可执行程序的经典人工智能框架不同，方法卡片采用纯 JSON，并将求值交由受限的外部引擎执行。由此形成的模式充当岩土工程分析的领域专用语言：它定义了专为工程计算量身打造的词汇和结构，使领域专家无需编程专业知识即可贡献方法，同时也解决了制约早期专家系统发展的知识获取瓶颈。

由于方法卡片是独立的数据文件，因此天然支持版本控制、公式层面的同行评审，以及通过标准软件开发工作流程实现的社区贡献。工程师可以派生一张卡片，更新方程以符合修订后的设计规范，并提交更改供评审，而无需修改源代码。每张卡片中嵌入的文献引文，确保每个公式的来源在修订过程中始终明确且可由机器读取。

2.3 岩土工程中的人工智能

近年来，机器学习和人工智能技术在岩土工程问题中的应用迅速增长。研究人员已成功应用人工神经网络 (ANNs) 预测土体行为 [7], 应用支持向量机进行沉降预测 [8], 并使用各种机器学习算法评估边坡稳定性

评估 [9], 液化潜势评估和承载力估算。这些数据驱动方法能够捕捉土体行为中的复杂非线性关系，并且在训练数据具有代表性时，往往能够给出良好的预测。然而，机器学习模型在本质上不同于基于物理的解析方法，其不透明性以及在训练条件之外适用性有限。用于预测承载力的人工神经网络无法说明得出某个特定数值的原因，也无法说明设计变更会如何影响承载力。这种黑箱特性与工程判断相冲突，并限制了其在安全关键应用中的可信度；在此类应用中，理解计算背后的推理与数值结果同样重要。

近年来，大语言模型展现出有望改变工程实践的能力：理解技术规范、生成代码并进行自然语言对话。已有多项研究探索了大语言模型辅助工程设计，包括为初步设计计算生成代码。然而，大语言模型生成的工程计算是否可靠仍令人担忧。近期针对大语言模型在岩土工程问题上的评估发现，各类问题中均存在概念错误、计算错误和依据错误，且在没有指导的情况下，准确率低于 30% [10]。另有关于工具增强型语言模型的研究 [11] 表明，或许存在另一种方案：大语言模型无需生成计算，而是可以学会调用能够可靠执行计算的外部工具。这一思路促成了 GeoMCP 的架构设计。

GeoMCP 为大语言模型提供经过验证的工具，而不是要求它们生成计算。大语言模型负责确定所需的计算、选择合适的方法卡片、提供输入并解读结果；这些任务适合发挥其自然语言理解能力，而实际计算则交由确定性、透明的方法卡片完成。这种分工兼顾了两种方法的优势：大语言模型擅长理解工程背景并将查询转换为结构化工具调用，而方法卡片则确保可靠性和可审计性。

3 方法论

本节介绍 GeoMCP 的设计理念和系统架构。我们首先阐述指导设计决策的原则，然后介绍系统的整体架构。

3.1 设计理念

GeoMCP 的核心设计原则是将岩土工程方法表示为结构化数据，而不是嵌入式代码。这一原则回应了 AI 辅助时代工程计算工具的若干基本需求。

透明性与可审计性。 工程计算，尤其是影响安全性的计算，必须能够由具备相应能力的专业人员验证。当公式嵌入程序代码中时，无论代码位于编译后的商业软件中，还是由大语言模型生成，验证都需要编程专业知识；如果源代码属于专有内容或是动态生成的，实际验证甚至可能无法进行。通过将方法表示为明确的 JSON 文档，GeoMCP 确保任何工程师都能检查公式、将其与已发表的资料来源进行比较，并验证其正确性。这种透明性不仅是为了方便，更是一种专业责任：工程师必须能够解释并为自己的计算结果辩护。

可引用性与可追溯性。 岩土工程方法有其权威来源，包括教科书、期刊论文、设计规范和标准。计算的有效性往往取决于是否正确实现了特定作者或标准机构所发布的方法。当方法被编码进软件时，与文献来源的联系通常会丢失，或仅在一般性文档中有所提及。方法卡片将引用作为方法表示中不可或缺的一部分：每张卡片都会明确列出其文献来源，单个公式也可以引用这些来源中的特定方程。这使工程师能够将每项计算追溯至其理论基础，从而支持专业实践和教育用途。

可扩展性和可维护性。岩土工程实践涵盖数百种分析方法，每种方法又针对不同的土壤条件、岩土结构类型和荷载情景有多种变体。以代码为中心的方法要求具备编程专业知识才能添加新方法或修改现有方法，这会形成瓶颈，限制能够参与贡献的人群。以数据为中心的方法降低了这一门槛：创建方法卡片需要岩土工程方面的专业知识以及对 JSON 语法的熟悉，但不需要编程技能。这使更多人能够参与扩充方法库，也让维护工作更适合以社区协作的方式长期开展。当设计规范修订或需要新增变体时，更新 JSON 文件比修改源代码更容易上手。

人工智能就绪性。大语言模型擅长理解结构化数据，并能可靠地解析、查询和推理 JSON 文档。将方法表示为结构化数据后，我们就能让大语言模型了解有哪些计算可用、需要哪些输入以及适用哪些假设，而无需大语言模型生成计算

规范。这种职责划分——由大语言模型理解上下文并选择方法，由确定性引擎执行计算——充分发挥了两种方法各自的优势，同时保持了可靠性。

这些原则使系统架构实现了清晰的关注点分离：方法卡片包含岩土工程知识（公式、假设、引用文献），评估引擎提供安全计算，智能体技能使大语言模型能够将多张卡片组合成实用的分析流程，而MCP接口将这些能力以工具形式对外提供。这种分离确保改进某一组件（例如新增方法、优化引擎、增强智能体技能或扩展接口）时无需改动其他组件。

3.2 系统架构

GeoMCP采用分层架构，旨在使方法保持显式、评估保持安全、访问保持灵活。该系统由四个相互独立的层次组成，并与AI客户端应用程序交互，如图1所示。

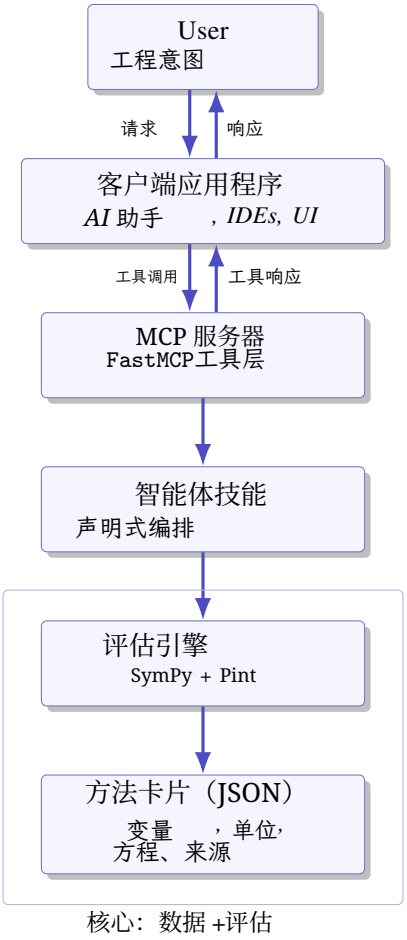


图1：GeoMCP 系统架构采用四层设计：以方法卡片作为知识库，通过受约束的评估引擎实现安全计算，利用智能体技能进行声明式编排和工程推理，并通过 MCP 服务器向客户端提供工具，支持双向请求-响应交互。

方法卡片 构成知识库层。每张卡片都是一个 JSON 文件，完整规定一种分析方法：其变量（包括物理单位）、方程（以 SymPy表达式表示）、变体（适用于不同条件）、假设（适用范围限制）以及来源（文献引文）。卡片采用版本控制，便于人类阅读，领域专家无需编程专业知识即可参与贡献。所有岩土工程知识、公式、分项系数和条件都位于这一层。特意选择 JSON 而非专用格式，可确保易于访问：任何工程师都可以使用标准文本编辑器或版本控制工具检查、验证或修改卡片。

评估引擎提供计算层。它基于 SymPy（符号数学）和 Pint（物理单位）构建，可解析方法卡片方程，并在严格的安全性和正确性保证下对其进行评估。该引擎

使用受限的 SymPy表达式允许列表，以防止代码注入并确保评估可复现。Pint 可自动进行量纲分析，捕捉单位错误，而这类错误是计算失误的常见来源。对于需要迭代求解的方法，该引擎支持带收敛性检查的不动点迭代，并且其功能还在不断扩展。重要的是，引擎本身不包含任何岩土工程公式，而是一个适用于方法卡片的通用安全评估框架。

智能体技能 提供编排和指导层。GeoMCP 并非将多步骤分析流程编码为程序代码，而是将其整理为结构化的 Markdown 文档，并遵循 Agent Skills 规范 [12]。每项技能都以目录形式组织，其中包含一个主指令文件（SKILL.md），内含 YAML 前置元数据（名称、描述、版本、类别），以及一个存放配套文档的 references/ 子目录。指令文件采用结构化的工程推理方法，针对具体任务量身定制，通常会先理解问题并进行场地评估，然后再 进行任何计算。例如，某项技能可能会指示助手首先判定问题类型并评估场地条件，然后依据工程判断选择适当的方法，通过有序的工具调用执行计算，最后结合公认的专业经验法则解释结果并进行合理性检查。参考文件可提供领域知识，供 AI助手与技能指令一并加载，例如说明各方法适用时机及原因的方法比较表、用于合理性检查的典型参数范围，以及相关设计规范表。由于技能采用纯 Markdown 编写，领域专家无需编程专业知识即可撰写、审阅和扩展这些内容。

这种方法解决了基于工具的人工智能交互中的一个根本局限。缺乏指导时，人工智能模型只能根据工具描述来推断多步骤分析流程和工程推理。在实践中，这会导致方法选择不理想、遗漏中间步骤，或缺少必要的上下文。另一种做法是将编排逻辑硬编码为程序逻辑，但这种方式难以扩展：要覆盖所有岩土分析领域，需要编写数千行过程式代码，而且每增加一个领域，都需要软件开发人员而非领域专家参与。技能使编排过程清晰、可审计，也便于任何具备岩土专业知识的人扩展。

MCP 服务器提供接口层。它提供用于与方法目录交互的基础工具（列出方法、检索方法卡片、使用数值输入或带单位标记的输入进行计算）、使客户端能够浏览、搜索和检索分析技能的技能发现工具，以及用于常见多步骤分析的组合工具。服务器内置的指令强制采用技能优先工作流：收到岩土问题后，AI助手首先调用 recommend_skills 查找合适的分析技能，然后通过 get_skill 加载该技能（及其参考文件），并遵循该技能中的工程推理来评估相关方法卡片。这种指令级强制机制确保 AI 助手以结构化的工程判断处理问题，而不是临时调用工具。服务器会执行输入验证和可选的单位标准化，并返回确定性结果及中间值，以确保可追溯性。通过实现模型上下文协议， [2], GeoMCP 可供任何兼容 MCP 的客户端使用，包括 ChatGPT 或 Claude 等 AI 助手、IDE 插件和自定义应用程序。

这一架构体现了我们的设计原则。方法保持明确且可审计（卡片层），评估安全且具有确定性（引擎层），编排采用声明式方式并由领域专家编写（技能层），访问方式则实现标准化（MCP 服务器层）。工程师可以检查方法卡片，并使用已公布的公式重新计算，从而验证任何计算；AI助手则遵循技能中的工程推理，并通过 MCP 工具调用相同的卡片。两种路径执行的底层方法完全相同，确保 AI 辅助分析与人工分析得出相同结果。下一节将介绍各层的实现。

4 实现

本节介绍 GeoMCP 核心组件的实现：用于表示岩土工程知识的方法卡片模式、能够安全且正确地执行计算的评估引擎、提供声明式编排与工程指导的 AgentSkills，以及用于开放方法评估和技能发现工具的 MCP 服务器接口。

4.1 方法卡片模式

方法卡片是一份结构化 JSON 文档，可完整描述一种分析型岩土工程方法。该模式旨在捕捉的不仅是数学内容，还包括正确应用方法并验证其实现所需的完整工程背景。

核心要素。 每张方法卡片都包含几个重要部分。元数据 部分提供基本标识信息：唯一 ID（例如 BEARING_CAPACITY_TERZAGHI）、人类可读标题、类别分类和描述性概述。变量 部分声明方法中使用的所有量，并为每个量指定其键（方程中使用的标识符）、人类可读名称、角色（输入、输出、中间量，或用于固定常数的参数）、物理单位，

以及可选的详细描述。变体 部分允许一张卡片涵盖某个方法的多个版本（例如，适用于条形基础、方形基础或矩形基础的版本），每个版本都有自己的一组方程。方程部分将计算流程定义为一系列 SymPy 表达式，每个表达式针对特定变量，并可选择包含适用条件和描述性文字。假设 和适用性 部分正式说明理论假设以及使用该方法的实际限制。最后，来源部分列出文献参考资料和 URL，将该实现与其理论基础联系起来。

表 1 总结了模式中的关键字段。这种结构确保方法卡片能够自我说明：工程师仅通过 JSON 文件本身，就能了解某种方法的计算内容、所需输入、适用条件以及公式的来源。

表 1：方法卡片模式中的关键字段。该模式定义了一种结构化表示形式，涵盖数学公式和工程背景。

字段	Type	描述
`id`	字符串	唯一标识符（例如，BEARING_CAPACITY_MEYERHOF）
标题	字符串	人类可读标题
类别	字符串	分类（例如，“浅基础 - 承载力”）
描述	字符串	该方法及其理论基础的简要概述
变量	数组	包含角色、单位和描述的变量规范
变体	数组	适用于不同计算条件的方法变体
假设	数组	限制该方法使用的理论假设
适用性	数组	关于何时应用该方法的实用指南
来源	数组	连接到参考文献的文献引文和 URL
变量说明		
`key`	字符串	方程中使用的标识符（例如，phi_prime）
`name`	字符串	易读变量名（例如，有效_摩擦_角）
`role`	字符串	输入、输出、中间量或参数
`unit`	字符串	物理单位（例如，千帕、弧度、米）
描述	字符串	特定量的通俗语言说明
方程规范		
目标	字符串	本方程计算的变量键
sympy	字符串	用于计算的SymPy表达式
条件	字符串	可选的适用条件（例如，phi_prime `> 0`）
描述	字符串	用于描述特定方程或分项系数的可选文本

设计依据。 该模式体现了若干经过深思熟虑的选择。我们选择 JSON 是因为它应用广泛；任何工程师都能使用标准文本编辑器编辑 JSON 文件，而且所有主流编程语言和版本控制系统都支持这种格式。该模式由 Pydantic 在加载时强制执行。 [13], Pydantic 是一个 Python 数据验证库，可检查必填字段、数据类型和结构约束，并在任何卡片进入评估引擎之前提供清晰的错误消息。每个变量都明确声明单位，而非默认假定或另行记录，从而能够通过 Pint [14]自动进行量纲分析。在单张卡片中设置多个变体可减少重复：通用变量和元数据只需定义一次，各变体专属的方程则可使方法适用于不同条件。文献来源是卡片中的一等元素，而非注释，因此引用可由机器读取，并能自动生成计算报告中的参考文献列表。

图1展示了太沙基承载力理论方法卡片的简化示例。卡片声明了包括极限承载力（q_ult）在内的变量，并将其设为输出；土体性质（c_prime、phi_prime、gamma）和几何参数（B、q）则作为输入。针对条形基础整体剪切破坏的变体定义了承载力系数（N_q、 N_c、 N_γ）的方程以及极限承载力公式。符号表达式采用 SymPy 库中的标准数学记法（例如 exp、tan、pi），因此既可由机器求值，也便于人类阅读。

这种以数据为中心的表示具有多项实用优势。首先，卡片是可验证的：工程师可以将 SymPy表达式直接与已发表的公式进行比较，以确认其正确性。其次，卡片是易于维护的：规范修订后，更新方法只需编辑一个 JSON 文件，无需在源代码中费力查找。第三，卡片是可扩展的：添加新方法或变体无需编程，因此岩土工程专家也能参与贡献

```
{
  "id": "承载力_容量_Terzaghi",
  "title": "Terzaghi 的承载力理论",
  "类别": "浅基础 - 承载力",
  "描述": "用于评估极限承载力的首个综合理论……",
  "变量": [
    {
      "key": "q_极限",   "name": "极限_承载力_",                    "role": "输出",
      "unit": "千帕",
      "description": "单位面积的极限承载力……"
    },
    {
      "key": "phi_prime",   "name": "有效_摩擦_角",                    "role": "输入",
      "unit": "弧度",
      "描述": "有效内摩擦角..."
    },
    "...（其余变量为简洁起见省略）"
  ],
  "变体": [
    {
      "id": "通用_剪切_破坏_条形基础",
      "title": "条形基础的一般剪切破坏",
      "方程": [
        {
          "target": "N_q",   "sympy": "exp(pi*tan(phi_prime))*tan(pi/4 + phi_prime/2)**2",
          "description": "承载力系数 N_q"
        },
        {
          "target": "N_c",   "sympy": "Piecewise(((N_q - 1)*cot(phi_prime), phi_prime > 0),          (5.14,    True))".
          "描述": "承载力系数 N_c"
        },
        {
          "target": "N_gamma",   "sympy": "2*(N_q + 1)*tan(phi_prime)",
          "描述": "基于 Terzaghi 的承载力系数 N_gamma..."
        },
        {
          "target": "q_极限值",  "sympy": "c_prime*N_c +q*N_q + 0.5*gamma*B*N_gamma",
          "description": "连续（条形）基础的极限承载力"
        }
      ]
    },
    "...（为简洁起见，省略其余变体）"
  ],
  "assumptions": [
    "地基较浅..."                    ,
    "基底粗糙...",
    "...（其余假设省略）"
  ],
  "适用性": [
    "适用于初步估算以及中心荷载下的浅层……",
    "...（其余适用性说明省略）"
  ],
  "来源": [
    {
      "title": "K. 太沙基（1943年）。《理论土力学》。John Wiley & Sons。",
      "url":  "...（URL 已省略）"
    },
    "...（其余来源已省略）"
  ]
}
```

图 1 太沙基承载力理论的简化方法卡片，展示了其核心要素：元数据、带单位的变量声明、符号方程、理论假设和文献来源。

即使他们不是软件开发人员。第四，卡片是可引用的：每项计算都可以通过方法卡片追溯到文献来源，从而满足专业文档要求。

4.2 评估引擎

评估引擎可执行方法卡片中定义的计算，并确保安全性和量纲正确性，同时支持岩土分析中常见的迭代求解过程。

安全的符号求值。 该引擎基于 SymPy [15], ，这是一款用于符号数学的 Python 库，它将表达式表示为代数对象，而非浮点近似。这意味着方法卡方程看起来就像数学公式（例如 exp(pi*tan(phi_prime)），并且可以通过检查进行验证。引擎在沙箱环境中解析这些表达式，将可用函数限制为预先批准的标准数学运算允许列表（sin、cos、exp、sqrt、log 等）和常量（pi、e）。它不使用 Python 的 eval()，也不允许访问内置函数和模块，从而确保方法卡片无法执行任意代码。对于一个需要评估外部贡献逻辑的系统而言，这是一项关键的安全特性。

量纲分析。 评估前，引擎使用 Pint 验证方程在量纲上是否一致：每个方程两侧的单位必须相容，且运算必须遵循量纲规则（例如，压力不能与长度相加）。当工程师提供带有明确单位的输入（例如，用“32 度”表示摩擦角）时，Pint 会自动将其转换为方法卡片所需的单位。如果检测到量纲不一致，引擎会在进行任何计算之前给出描述性错误提示，这类似于强类型编程语言中的编译时类型检查。

迭代求解支持。 许多岩土工程方法存在循环依赖，无法显式地将某个变量单独求解。例如，在为承载力设计地基尺寸时，容许承载压力取决于所需的基础宽度（B）；然而，计算作用承载压力并满足设计准则也需要 B 本身。评估引擎通过多阶段求解策略处理这些情况。首先，它会对方程组进行多轮遍历，尝试解析依赖关系。对于尚未解决的循环依赖，引擎会采用带收敛性检查的数值不动点迭代。总体而言，评估引擎会持续提升自身能力，以满足新方法卡片不断扩展的数学需求。

性能与局限性。符号求值方法以牺牲部分计算性能为代价，换取透明度和可验证性。对于涉及数十到数百个方程的典型岩土工程设计计算，求值时间通常为毫秒级，足以满足交互式使用需求，但速度低于编译型数值代码。对于工程师审查时间远超计算时间的设计计算而言，这是可以接受的权衡。对于需要进行数千次重复求值的应用（例如蒙特卡罗分析），可能需要采用更优化的方法，不过方法卡片仍然很有价值，因为它能确保实现与已发表的方法一致。

4.3 智能体技能

方法卡片虽然能描述单个分析方法，但实际工程任务通常需要按顺序组合多种方法/方程，根据问题特征选择合适的方法，并依据工程判断和专业规范解读结果。传统做法是将这些多步骤流程编写为程序代码，但这会形成可扩展性障碍：每个新的岩土领域（土压力、沉降、边坡稳定性、桩承载力）都需要软件开发人员编写数百行编排逻辑。GeoMCP 通过智能体技能解决这一问题：以结构化 Markdown 文档的形式记录相同的编排逻辑和工程推理，既便于人类阅读，也便于机器解读。

每项技能均遵循 Agent Skills 规范 [12] ，并以目录形式组织，其中包含一个主指令文件（SKILL.md），文件通过 YAML 前置元数据记录元数据，并使用 Markdown 正文描述分析流程；此外还可包含提供领域知识的参考文件，例如方法比较表、典型数值范围和示例。激活技能时，AI助手会一并加载这些参考文件和指令。

GeoMCP 的技能在设计上的关键决策是教授工程推理，而不只是工具调用序列。例如，浅基础承载力技能提供了结构化指导，其方式与经验丰富的岩土工程师的工作方法相仿。该技能并非规定一套僵化的步骤，而是引导 AI 首先判断问题类型（设计验算还是基础尺寸设计），并评估场地条件（排水、破坏模式、地下水）。随后，它通过编码工程判断的决策树，引导选择合适的方法；按照顺序协调工具调用来执行计算；在用户要求时执行特定设计规范工作流（例如 EC7 分项系数设计）；最后要求助手结合典型范围和专业经验法则解读结果并进行合理性检查。

重要的是，这一推理阶段发生在任何工具调用之前。该技能确保 AI 助手先停下来思考问题，对其进行分类、评估场地条件并选择合适的方法，然后再使用计算器。缺少这一指导时，人工智能模型往往会直接进行计算，常常不考虑所选方法是否适用于给定条件。

与硬编码编排相比，这种方法具有多项优势。首先，技能可由领域专家扩展：例如，岩土工程师只需使用 Markdown，就能编写一项用于侧向土压力分析的新技能，无需

编写 Python 代码。其次，技能能够捕捉工程推理，而规范无法做到这一点：例如，“对于承受倾斜荷载的矩形基础，Vesic 优于 Terzaghi，因为它考虑了压缩性并提供荷载倾斜系数”这类考量，用文字自然表达很容易，但要将其编码为条件逻辑则很不方便。第三，技能透明且可审计：完整的分析流程都呈现在人类可读的文档中，岩土工程专家即使没有编程背景，也能对其进行同行评审。

4.4 MCPServer 接口

MCP 服务器提供标准化接口，供客户端应用程序访问 GeoMCP 的各项能力。Anthropic 开发的模型上下文协议（MCP） [2], 规定了应用程序如何将自身能力作为可发现、可调用的工具公开，并提供结构化的输入和输出。GeoMCP 通过 FastMCP（一种负责处理协议细节的 Python 库）实现 MCP，因此无需为每个客户端单独编写集成代码，即可兼容任何支持 MCP 的客户端，包括 Claude Desktop、ChatGPT 和自定义应用程序。

工具类别。服务器提供多个类别的工具。基础工具提供对方法目录的底层访问：列出方法、获取完整的方法卡片以供检查，以及使用标准化数值输入或带单位标记的输入（例如“30 deg”、“18kN/m³”）评估方法；这些输入会自动转换为方法卡片所要求的单位。值得注意的是，调用评估工具时，服务器会向客户端返回可审计的追踪记录，其中包含所有输入、中间变量计算和最终结果。技能发现工具使 AI 助手能够根据问题描述的相关性浏览、搜索和检索分析技能。服务器还提供会话管理功能，以便重复使用参数默认值，并提供诊断健康检查。

技能优先工作流。服务器内置的指令引导 AI助手采用以技能为中心的方式：当收到问题描述时，助手会被提示使用技能发现工具，查找并加载合适的分析技能及其相关参考文件。助手不会盲目调用计算工具，而是遵循技能提供的领域特定指导。这些指令层面的引导确保 AI助手以结构化的工程推理处理每个问题，而不是临时调用工具。

设计决策。GeoMCP 使评估引擎保持确定性且无副作用：给定相同的卡片、变体和规范化输入，就会产生相同的结果。为提升易用性，服务器支持可选的会话上下文（可在多次工具调用中重复使用的默认值），但这一便利层不会改变底层数学逻辑，该逻辑始终由方法卡片承载。这种分离使系统便于审计，同时支持实用的工程分析。

由于协议已经标准化，MCP 客户端的改进会自动惠及 GeoMCP 用户；方法目录或技能库中的新增内容也会立即提供给所有客户端，无需修改客户端。

5 应用与验证

本节从用户视角展示 GeoMCP，说明 AI助手如何使用该框架完成一项完整的欧洲规范7承载力设计。我们以 JRC（联合研究中心）的官方算例 [16] 为依据，追踪完整工作流程：从自然语言问题描述，到技能引导的分析，再到经过验证的结果。本演示既验证了该框架的计算准确性，也展示了技能引导式 AI 交互的实际效用。

5.1 方法目录概述

当前的 GeoMCP 实现包含多种方法卡片，涵盖基础土力学、浅基础与深基础、侧向土压力、沉降和边坡稳定性，且方法与变体的数量还在不断增加。本文的验证聚焦于承载力子集，其中包括一些成熟方法，例如 Terzaghi 的经典理论（BEARING_CAPACITY_TERZAGHI） [17, 18], Meyerhof 的形状和深度系数公式（BEARING_CAPACITY_MEYERHOF） [19, 18], 考虑压缩性的 Vesic 方法（BEARING_CAPACITY_VESIC） [20, 18], 以及欧洲规范7的排水和不排水计算（BEARING_CAPACITY_EUROCODE7） [21, 22]。

5.2 验证场景：JRC《欧洲规范7》算例A.3

我们对照欧盟委员会发布的、作为EN 1997-1权威实现指南的官方JRC《欧洲规范7》算例 [16], 对GeoMCP进行验证。

向 AI助手提出的问题是一个直接的**设计**请求：

为一栋 4 层混凝土建筑设计一个条形基础。基础的埋置深度为$D_f = 1.5\,m$。六根柱产生的永久荷载特征值总和为 $G_{k,\,\mathrm{col}} = 3500.96kN$，可变荷载特征值为 $Q_k = 967.10kN$。基础长度为 $L = 21.4\,m$。请确定满足欧洲规范7所有设计方法所需的宽度 B。

数据一致性说明：JRC 示例文本（第 105 页）指出，地下水位为 1.5 m。然而，JRC 岩土工程模型图 A.3.111（第 112 页）明确显示地下水位为 2.5 m （位于基底以下 1.0 m ）。若要复现 JRC 的具体结果，必须使用图中的数值。

5.3 AI 辅助分析演练

我们追踪 AI助手求解所需宽度 B 的工作流程。图 2 展示了交互过程 n.

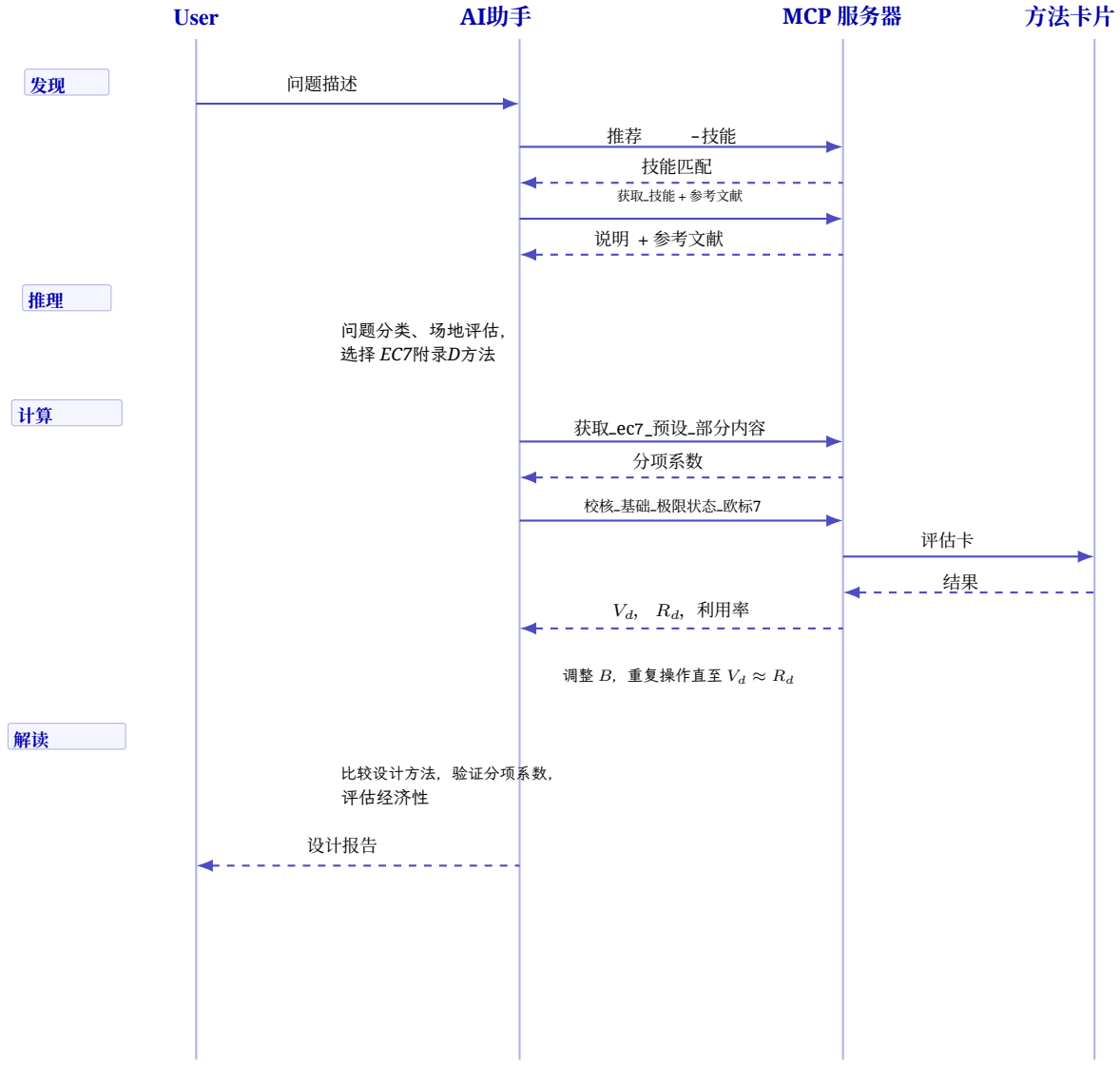


图2：技能引导的欧洲规范7设计交互序列。助手发现并加载合适的技能，对问题进行推理，使用承载能力极限状态验算工具迭代搜索所需宽度，并在报告结果前对其进行解读。

发现与推理。 该助手发现浅基础承载力技能并对问题进行分类：欧洲规范7的承载能力极限状态设计。它注意到，对于设计方法1，必须检查两个组合（C1

和 C2），而 DA2 和 DA3 只有一个组合。根据土体性质（$\phi' = 38°$），它预判会发生整体剪切破坏，并选择采用排水条件下的 EC7 附录 D 方法。

设计作用。 按照技能建议的 EC7工作流程，助手首先获取每种设计方法对应的分项系数。清单 2 展示了针对 DA1-C2 的工具调用及响应。随后，将分项系数集 A2+M2+R1 ($\gamma_G = 1.0$,$\gamma_Q = 1.3$, $\gamma_\phi = 1.25$, $\gamma_R = 1.0$) 应用于特征荷载，以计算设计作用 V_d。由于该技能推荐检查所有设计方法，助手对 DA1-C1、DA2 和 DA3 重复执行这一操作。

```
1  // 请求2 {"design_approach":"DA1-C2" }34  // 响应5  {"design_
app
7
8
gam
gamm
分项
```

清单 2 geo_get_ec7_preset_ 的 MCP 工具调用与响应。请求中仅包含设计方法；响应则返回完整的 EN 1997-1 分项系数组。

承载力与抗力。 对于每种设计方法，助手都会迭代搜索满足 B where$V_d \approx R_d$ 条件的宽度。每次尝试某一宽度时，都会调用 check_footing_uls_ec7，并传入几何尺寸、土体特征参数、设计方法和分项作用（包括该试算宽度下的基础自重）。该工具应用分项系数，将土体特征参数折减为设计值，根据欧洲规范7附录 D 承载力方法卡进行评估，并返回包含利用率的可审计追踪记录。
验证与解读。 助手会检查收敛宽度下的承载力分项系数，并将设计方法的排序（DA2 最经济，DA3 最保守）与 JRC 的结论进行比较。助手还指出尚未检查沉降；该正常使用极限状态验证需要额外的方法卡片。

5.4 验证结果

我们从三个层面对 GeoMCP 的结果进行验证，并与 JRC 参考结果进行比较：承载力系数、所需基础宽度，以及 JRC 参考宽度下的设计作用/抗力。

承载力系数 表 2 列出了 DA1-C2 在收敛宽度下的中间承载力和形状系数。所有系数均与算例一致，证实 BEARING_CAPACITY_EUROCODE7 方法卡片正确实现了理论。

表 2：DA1-C2 的承载力和形状系数（$\phi'_d = 32.0°$， $B = 1.497$米）。

参数	JRC	GeoMCP
N_q	23.17	23.19
N_c	35.48	35.51
N_γ	27.72	27.74
s_q	1.04	1.037
s_γ	0.98	0.979

所需基础宽度 主要验证指标为**所需基础宽度（B）**，因为这一单一决策变量综合体现了所有分项系数、承载力方程、形状系数和荷载组合。表3将由技能协调的设计搜索得出的宽度与JRC参考值进行了比较。

结果证实了JRC的结论：DA2最经济（$B = 1.211$m），而DA3最保守（$B = 1.738$m）。小数点后第三位的细微差异源于承载力系数（表2）的轻微舍入误差，远低于基础设计所需的精度。

设计作用与抗力作为最终验证，表 4 对比了 JRC参考宽度下的 V_d 和 R_d。由于 V_d 包含基础自重（$\gamma_{\text{concrete}} \times B \times D_f \times L$），其数值取决于所选宽度，因此在确定宽度后再予以列出。V_d 数值完全一致（精确到所报告的精度），R_d 的吻合度在 0.012% 以内，证实 GeoMCP 能够复现 JRC 的完整计算流程。

表 3：各设计方法所需的基础宽度。DA1-C1 不起控制作用，JRC 参考资料中未报告该项。

		宽度 B (m)	
设计方法	分项系数集	JRC	GeoMCP
DA1-C1	A1+M1+R1	–	1.038
DA1-C2	A2+M2+R1	1.50	1.497
DA2	A1+M1+R2	1.21	1.211
DA3	A1+M2+R3	1.74	1.738

表4：JRC参考宽度处的设计作用（V_d）和抗力（R_d）。

	V_d (kN)		R_d (kN)	
DA	GeoMCP	JRC	GeoMCP	JRC
DA1-C2	5661.00	5661.00	5675.89	5676.52
DA2	7160.11	7160.10	7151.43	7150.58
DA3	7590.75	7590.75	7611.23	7612.07

所有验证均在代码仓库中实现为自动化回归测试，防止方法目录演进过程中出现偏差。

6 讨论

6.1 解读验证结果

JRC《欧洲规范7》验证表明，GeoMCP 的评估引擎能够以极高精度复现权威参考值：承载力系数的偏差在 0.15 % 以内，设计作用与报告精度完全一致，设计抗力的偏差则在 0.012% 以内。这些结果证实了 GeoMCP 的基本架构理念是可靠的：将岩土工程方法从命令式代码中抽象出来，并以声明式 JSON 中的符号表达式表示，不会引入有意义的数值误差。该框架在计算结果上与成熟的实现方法不相上下，同时实现了充分的方法级透明度和可追溯性。

6.2 对工程实践的启示

将方法规范与评估逻辑分离，带来了一项超越 AI 应用场景的实际优势：分析方法本身因此可以独立接受审查、进行版本管理并作为独立成果引用。工程师审查 GeoMCP 计算时，可以将每一项输出追溯到特定方法卡片中的具体公式，再由此追溯到最初发表的来源。使用电子表格时，公式隐藏在单元格引用之后；使用专有软件时，底层实现则是已编译的黑箱，因此要实现这种程度的可追溯性极为困难。

这种透明性是在岩土工程工作流程中负责任地部署 AI 的前提。当业界讨论 AI 智能体在选择方法或配置分析时专业责任如何适用之际，可辩护工作流程的最低要求便是可审计性。通过记录分析链中的每一项决策，GeoMCP 确保透明性并不取代专业判断，而是使人们能够有效地对 AI 的行为行使专业判断。

最终，这种问责机制推动了更广泛的范式转变。我们认为，土木工程下一波生产力提升，不会来自单个大语言模型逐步变得更聪明，也不会来自孤立的软件工具不断增加专有功能。工程软件的未来，在于让我们的集体知识和可信赖的计算工具实现广泛互操作，并达到“AI 就绪”状态。GeoMCP 从嵌入式代码中提取分析方法，并通过 MCP 等标准化协议将其作为结构化数据开放出来，从而推动行业摆脱为人工操作者构建单体工具的模式。它指向一个互联的生态系统，在其中，AI助手能够大规模、可靠地编排确定性且可验证的工程计算。

6.3 局限性与风险

JRC 验证表明，该方法在特定问题类别中具有精确性。要发布可投入生产的 GeoMCP 版本，就必须针对更广泛的一系列可通过分析方法解决的岩土工程问题开展系统验证。要实现这一扩展，必须具备全面的验证数据集，其中还应包含中间计算值。此外，方法卡片方案本质上仅适用于闭式方程和迭代系统。

一个更值得关注的局限涉及人机交互范式。虽然 GeoMCP 可以强制大语言模型在使用分析方法时保持确定性，但无法消除推理或数据提取错误的可能性。大语言模型充当“操作员”，负责解读问题情境、选择适当的技能，并从用户提示中提取输入参数。在这一解读阶段出现的错误，例如误认土壤参数，或错误判断排水条件与不排水条件，都会传递到引擎的输入中。

尽管系统提供的可审计计算轨迹使所有输入值都能在最终报告中清晰呈现，但这种透明性也带来了自动化偏见这一衍生风险。当系统生成一份精致、格式专业的设计报告，其中包含学术引文和无可辩驳的数学计算轨迹时，输出结果所呈现的权威感可能会让人忽视必要的批判性审查。透明性为核验提供了手段，但强制独立校核等组织实践仍不可或缺。该框架将大语言模型的自主性限制在推理和编排范围内，严格压低了发生隐性数学错误的可能性，同时也强化了这一原则：人工智能工具应当增强而非取代专业工程判断。

7 结论

大语言模型的出现为工程工作流程带来了变革潜力，但其统计特性与安全关键结构与岩土设计所要求的严格确定性存在根本冲突。本文提出了 GeoMCP，这是一个旨在弥合这一差距的框架，其前提是工程计算应表示为结构化数据，而非嵌入式代码。

通过将分析方法捕获为声明式 JSON 方法卡（包含公式、单位、适用条件和文献引文），并经由受限符号引擎进行求值，GeoMCP 保证了与传统实现方法的数值一致性。对照欧洲规范7官方算例的验证确认，这种以数据为中心的抽象不会带来数值损失，能够准确复现权威的设计抗力和基础尺寸，同时提供透明、可审计的计算轨迹。
更重要的是，该框架将人工智能的角色从不可靠的计算器转变为智能编排者。通过模型上下文协议和结构化的智能体技能，大语言模型主导工程推理，同时将数学计算委托给统一、可验证的事实来源。

GeoMCP 的架构为该行业展示了一条可行的前进道路。它摒弃不透明的电子表格和专有黑箱，转而倡导开放、可互操作且面向人工智能的生态系统。通过使计算明确、可引用且易于获取，GeoMCP 使工程师能够利用现代人工智能的强大能力，同时不放弃其专业责任。

源 c ode for GeoMCP 将在本文发表后于 GitHub 上公开发布 .

人工智能使用披露

作者声明，在本稿件准备过程中，生成式人工智能工具仅用于校对、编辑和格式排版辅助。研究设计、概念构思、方法论、分析以及对结果的解释均为作者的原创工作。

参考文献

[1] Raymond R Panko。《电子表格错误：我们所知的一切》。终端用户计算期刊，10(2):15–21，2008年。
[2] Anthropic。模型上下文协议（MCP）。https://github.com/modelcontextprotocol/spec，2024年。[3] Bruno Stuyts。groundhog：通用岩土工程软件包。https://github.com/snakesonabrain/ groundhog，2025年。

[4]Patrick Okonkwo。geolysis：岩土工程分析库。https://github.com/ patrickboateng/geolysis，2024年。[5] Jesse Bonanno。pyslope：用于边坡稳定性分析的 Python 软件包。https://github.com/JesseBonanno/ PySlope，2024年。[6] Marvin Minsky。知识表示框架。*MIT*人工智能实验室备忘录 *306*，1974年。[7] Mohamed A Shahin、Mark B Jaksa 和 Holger R Maier。人工神经网络在岩土工程中的应用。澳大利亚岩土工程，36(1):49–62，2001年。[8] Pijush Samui。支持向量机在无黏性土浅基础沉降分析中的应用。计算机与岩土工程，35(3):419–427，2008年。[9] Chongchong Qi 和 Xiaolin Tang。采用集成元启发式与机器学习方法预测边坡稳定性：一项比较研究。计算机与工业工程，118:112–122，2018年。[10] Liuxin Chen、Amir Tophel、Umidu Hettiyadura 和 Jayantha Kodikara。大语言模型在岩土工程教育与问题求解中的实用性研究。岩土工程，4(2):470–498，2024年。[11] Timo Schick、Jane Dwivedi-Yu、Roberto Dessì、Roberta Raileanu、Maria Lomeli、Eric Hambro、Luke Zettlemoyer、Nicola Cancedda 和 Thomas Scialom。Toolformer：语言模型可以自学使用工具。神经信息处理系统进展，37，2023年。[12] Agent Skills 贡献者。Agent Skills 规范。https://agentskills.io/specification，2025年。[13] Samuel Colvin 等。Pydantic：使用 Python 类型提示进行数据验证。https://docs.pydantic. dev/，2024年。[14] Hernan E. Grecco 等。Pint：Python 单位。https://pint.readthedocs.io/，2024年。[15] Aaron Meurer 等。Sympy：Python 符号计算。*PeerJ* 计算机科学，3:e103，2017年。[16] 联合研究中心。在欧洲规范7：岩土设计研讨会上展示的算例。技术报告，欧盟委员会，联合研究中心，意大利伊斯普拉，2005年。[17] Karl Terzaghi。理论土力学。John Wiley & Sons，1943 年。[18] Braja M Das 和 Nagaratnam Sivakugan。基础工程原理。Cengage Learning，2018年。[19] G.G. Meyerhof。地基承载力的最新研究。加拿大岩土工程期刊，1(1):16–26，1963年。[20] Aleksandar S. Vesi´c。浅基础极限荷载分析。土力学与基础工程分部期刊，99(1):45–73，1973年。[21] CEN。欧洲规范7：岩土设计——第1部分：通则，2004年。EN 1997-1:2004。[22] Brian Simpson 和 Robin Driscoll。欧洲规范*7*：评注。建筑研究通讯，2012年。