XXII Congresso Brasileiro de Mecânica dos Solos e Engenharia Geotécnica
XI Simpósio Brasileiro de Engenheiros Geotécnicos Jovens
24 a 28 de agosto de 2026 – Brasília/DF
DOI: 10.47094/COBRAMSEG2026.2/11

Da Conversa à Simulação: Modelagem Numérica Geotécnica Mediada por Modelos de Linguagem de Larga Escala (LLMs)

Matheus Tavares Dias
Engenheiro Geotécnico, Universidade de Brasília – UnB, Brasília, Brasil, matheusttd@gmail.com

André Luís Brasil Cavalcante, D.Sc.
Professor, Universidade de Brasília – UnB, Brasília, Brasil, albrasilc@gmail.com

Luan Carlos de Sena Monteiro Ozelim, D.Sc.
Engenheiro Civil, Universidade de Brasília – UnB, Brasília, Brasil, luanoz@gmail.com

RESUMO: O avanço da Inteligência Artificial (IA) tem transformado a forma como engenheiros interagem com sistemas computacionais complexos. Na engenharia geotécnica, o método dos elementos finitos (MEF) é amplamente empregado para simular o comportamento tensão-deformação e avaliar a estabilidade de estruturas de solo sob condições hidromecânicas diversas, embora sua aplicação ainda exija conhecimento especializado e dedicação na etapa de configuração. Nesse contexto, o presente trabalho apresenta o GeoAssist AI, uma plataforma inteligente desenvolvida para integrar modelos de linguagem de larga escala (LLMs) a ferramentas de modelagem numérica de código aberto, utilizando Python como base de integração. O principal objetivo da plataforma é automatizar e simplificar o processo de criação de modelos geotécnicos, permitindo que o engenheiro configure e execute simulações por meio de comandos em linguagem natural. A partir de descrições textuais, o sistema interpreta as instruções do usuário, gera a estrutura do modelo, define propriedades de materiais, condições de contorno e malhas, e executa automaticamente a simulação numérica. O GeoAssist AI é composto por três módulos complementares: (i) o módulo de interpretação semântica, que traduz a linguagem natural em comandos técnicos; (ii) o módulo de geração automática de modelos, que cria os arquivos de entrada compatíveis com o solver numérico; e (iii) o módulo de análise e visualização, que processa os resultados e apresenta gráficos de tensões, deformações e fatores de segurança. Como demonstração funcional, foi implementado um teste de execução automática de uma simulação geotécnica simples, em que o usuário fornece apenas instruções descritivas para gerar e visualizar um modelo de estabilidade. O sistema executa o processo de forma autônoma, convertendo o texto em código e exibindo os resultados numéricos e gráficos de forma integrada. Os resultados preliminares indicam que o GeoAssist AI representa um avanço metodológico na integração entre IA e modelagem numérica, aproximando o engenheiro do ambiente de simulação e ampliando a acessibilidade às análises. Além disso, o sistema demonstra potencial para uso em ensino, pesquisa e prática profissional, reforçando seu papel como ferramenta inovadora de apoio à engenharia geotécnica.

PALAVRAS-CHAVE: Inteligência Artificial, Modelagem Numérica, Elementos Finitos, Estabilidade, Python.

ABSTRACT: The advancement of Artificial Intelligence (AI) has transformed the way engineers interact with complex computational systems. In geotechnical engineering, the Finite Element Method (FEM) is widely employed to simulate stress–strain behavior and assess the stability of soil structures under different hydromechanical conditions, although its application still requires specialized knowledge and considerable effort during the model setup stage. In this context, the present study introduces GeoAssist AI, an intelligent platform developed to integrate large language models (LLMs) with open-source numerical modeling tools, using Python as the integration framework. The main objective of the platform is to automate and simplify the process of creating geotechnical models, allowing engineers to configure and execute simulations through natural language commands. From textual descriptions, the system interprets user instructions, generates the model structure, defines material properties, boundary conditions, and meshes, and automatically runs the numerical simulation. GeoAssist AI is composed of three complementary modules: (i) the semantic interpretation module, which translates natural language into technical commands; (ii) the automatic model

generation module, which creates input files compatible with the numerical solver; and (iii) the analysis and visualization module, which processes the results and presents stress, deformation, and factor-of-safety plots. As a functional demonstration, an automatic execution test of a simple geotechnical simulation was implemented, in which the user provides only descriptive instructions to generate and visualize a stability model. The system autonomously executes the process, converting text into code and displaying the numerical and graphical results in an integrated manner. Preliminary results indicate that GeoAssist AI represents a methodological advance in the integration of AI and numerical modeling, bringing engineers closer to the simulation environment and increasing the accessibility of geotechnical analyses. In addition, the system shows potential for application in teaching, research, and professional practice, reinforcing its role as an innovative support tool for geotechnical engineering.

KEYWORDS: Artificial Intelligence, Numerical Modeling, Finite Elements, Stability, Python.

1 INTRODUÇÃO

A modelagem numérica é uma ferramenta central na análise de problemas geotécnicos complexos, especialmente quando estão presentes não linearidades constitutivas, heterogeneidade estratigráfica e acoplamento hidromecânico. Nesse contexto, o Método dos Elementos Finitos (MEF) consolidou-se como uma abordagem dominante, ao permitir a resolução de problemas governados por equações diferenciais parciais por meio de discretização espacial e formulações constitutivas elasto-plásticas (Zienkiewicz, Taylor, Zhu 2013; Potts, Zdravkovic 1999).

Apesar de sua consolidação teórica e de seu amplo uso, a aplicação do MEF em rotinas de projeto e pesquisa ainda enfrenta uma barreira operacional relevante. A construção de um modelo exige a definição explícita de geometria, malha, propriedades dos materiais, condições de contorno e etapas de carregamento, além da manipulação de arquivos de entrada e interfaces com sintaxes particulares de cada *solver*. Na prática, essa fase de pré-processamento demanda domínio geotécnico e computacional simultaneamente, o que pode restringir a acessibilidade, aumentar o tempo de preparação e ampliar a probabilidade de inconsistências ou omissões na configuração.

Em paralelo, modelos de linguagem natural de larga escala (LLMs), baseados na arquitetura Transformer (Vaswani et al. 2017), têm demonstrado capacidade de interpretar instruções técnicas e produzir representações estruturadas, inclusive código. O conceito de *foundation models* (Bommasani et al. 2021) ampliou o uso desses sistemas como intermediários entre linguagem humana e sistemas formais.

Diante disso, observa-se uma lacuna operacional: a ausência de um mecanismo estruturado que conecte, de modo rastreável, a descrição conceitual do problema geotécnico à sua implementação numérica. Este trabalho apresenta o GeoAssist AI, uma arquitetura que formaliza essa mediação por meio de um *schema* intermediário que separa a interpretação semântica da execução numérica determinística. A contribuição do artigo é metodológica: propor e demonstrar um *pipeline* reprodutível que conecta descrições em linguagem natural à modelagem geotécnica via MEF, preservando a coerência entre hipóteses físicas, organização de entradas e execução do *solver*.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Método dos Elementos Finitos na Engenharia Geotécnica

Sob uma perspectiva formal, problemas geotécnicos analisados pelo Método dos Elementos Finitos (MEF) podem ser formulados como problemas de valor de contorno governados pelas equações de equilíbrio, pelas relações cinemáticas entre deslocamentos e deformações e pelas leis constitutivas do material. Em notação contínua, busca-se o campo de deslocamentos u(x) que satisfaça as equações de equilíbrio $\nabla \cdot \sigma + b = 0$ no domínio, em conjunto com condições de contorno essenciais e naturais. A formulação em elementos finitos resulta da conversão do problema diferencial em forma fraca, com aproximação dos campos por funções de interpolação definidas em um conjunto discreto de elementos, conduzindo a um sistema algébrico que fornece

aproximações para deslocamentos, deformações e tensões (Zienkiewicz, Taylor, Zhu 2013; Potts, Zdravkovic 1999).

Em geotecnia, a não linearidade é frequentemente dominante, uma vez que os solos apresentam comportamento elasto-plástico e dependente da trajetória de tensões. Por isso, a solução numérica exige procedimentos incremental-iterativos para atualização de tensões, verificação de plastificação e convergência do equilíbrio global. Em análises de estabilidade, destaca-se o procedimento de redução progressiva da resistência (*Strength Reduction Method*), no qual os parâmetros resistentes são degradados até a perda de convergência, usualmente interpretada como condição limite numérica (Potts, Zdravkovic 1999). Outro aspecto central é a definição das condições hidráulicas: em problemas com poropressão, o equilíbrio mecânico se relaciona diretamente à distribuição de pressões neutras e, consequentemente, às tensões efetivas, seja em meios saturados, seja em formulações que considerem sucção em solos não saturados (Biot 1941; Fredlund, Rahardjo 1993).

No contexto deste artigo, a relevância do MEF não se limita à sua capacidade de simulação, mas à sua função como formalização rigorosa de hipóteses físicas em uma representação matemática e computacional. Assim, qualquer abordagem que pretenda automatizar ou mediar a construção de modelos numéricos deve preservar a consistência entre hipóteses adotadas, representação das entradas e execução efetiva do *solver*.

2.2 Modelos de Linguagem Natural de Larga Escala (LLMs)

Modelos de linguagem natural de larga escala (*Large Language Models* — LLMs) são sistemas baseados predominantemente na arquitetura Transformer, capazes de capturar dependências contextuais em sequências extensas e de generalizar para múltiplas tarefas linguísticas e técnicas (Vaswani et al. 2017). No contexto dos *foundation models*, esses sistemas podem servir como base adaptável para tarefas como síntese de texto técnico, extração de informações estruturadas e geração de código (Bommasani et al. 2021). Essa última capacidade é particularmente relevante em engenharia, pois permite mapear descrições textuais em representações formais computacionais (Chen et al. 2021).

Em aplicações técnicas, entretanto, a geração textual livre é insuficiente quando se exige confiabilidade. Nessas condições, o papel dos LLMs é mais adequadamente entendido como o de mediadores semânticos entre linguagem humana e sistemas formais, desde que acoplados a mecanismos determinísticos de validação, tipagem e execução, capazes de reduzir ambiguidades e assegurar consistência das entradas.

3 METODOLOGIA: ESTRUTURAÇÃO DO *PIPELINE* DO GEOASSIST AI

3.1 Visão Geral do Pipeline

O GeoAssist AI é estruturado como um *pipeline* que organiza o processo de modelagem numérica geotécnica em três etapas: (i) estruturação conceitual das entradas, (ii) tradução programática para o *solver* e (iii) execução com pós-processamento automatizado. A proposta não altera a formulação matemática do MEF; ela reorganiza o fluxo operacional introduzindo uma camada estruturada entre linguagem natural e *solver*. Na **Erro! Fonte de referência não encontrada.** é apresentado um esquema geral do *pipeline*.

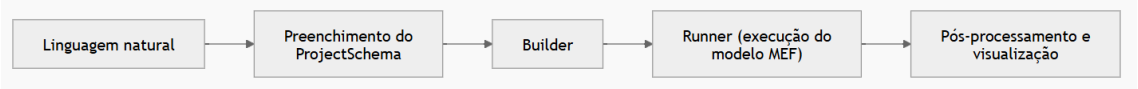


Figura 1. Estrutura conceitual do pipeline do GeoAssist AI, evidenciando a separação entre estruturação conceitual, tradução programática e execução numérica.

Essa camada é formalizada por meio de um *schema* de projeto, que atua como contrato entre a descrição conceitual do problema e sua materialização computacional. A etapa de linguagem natural não é tratada como garantia de correção técnica, mas como interface para preenchimento da estrutura de entradas. A consistência das informações é delegada ao *schema*, por meio de validações e restrições formais. Essa decisão é coerente com princípios de ciência computacional reprodutível, nos quais a confiabilidade depende de um caminho de execução auditável e de premissas registradas como dados, e não como inferências implícitas.

3.2 Estruturação das Entradas por Meio de *Schema*

O núcleo da mediação é o *ProjectSchema*, que representa o gabarito de entrada do sistema. Esse *schema* organiza parâmetros conceituais em quatro blocos: (I) especificação geométrica, (II) biblioteca de materiais, (III) condições de contorno e (IV) configurações do *solver*. O *schema* não armazena comandos específicos do *solver*; ele registra informações físicas e operacionais de alto nível, estruturadas de modo a permitir validação e tradução determinística.

A biblioteca de materiais contempla definições baseadas no modelo de Mohr–Coulomb, com parâmetros como densidade, módulos elásticos, coesão, ângulo de atrito, dilatância e resistência à tração. Esses parâmetros permanecem no nível conceitual até serem convertidos para a sintaxe do *solver* na etapa de compilação.

A geometria é descrita em um nível de abstração que favorece reconstrução programática. Quando expressa de forma paramétrica, essa representação reduz ambiguidades ao explicitar relações geométricas relevantes (altura, inclinações, camadas, crista) e restringir variáveis a parâmetros definidos. De modo análogo, a linha freática é representada por pontos de controle, permitindo interpolação e imposição coerente de poropressões iniciais, aspecto sensível em análises de estabilidade e problemas hidromecânicos (Biot 1941; Fredlund, Rahardjo 1993).

Condições de contorno e configurações do *solver* são descritas em termos conceituais, incluindo fixações geométricas, parâmetros de malha, unidade de tensão, gravidade e peso específico da água. O *schema* busca garantir consistência interna, como correspondência entre materiais e camadas e coerência de unidades.

3.3 Tradução para o *Solver: Builder*

Após validação do *ProjectSchema*, o *pipeline* avança para a etapa de compilação programática, implementada pelo *builder*. Essa etapa realiza a transição do nível conceitual para o operacional do *solver*, convertendo descrições abstratas em entidades discretizadas, identificadores internos e comandos executáveis. Essa necessidade decorre da natureza do MEF: embora o engenheiro formule o problema em termos físicos, o *solver* opera sobre discretizações e instruções formais.

Nesse processo, são geradas as linhas de contorno que definem o domínio e as interfaces internas entre camadas. A biblioteca de materiais é mapeada para identificadores numéricos internos, permitindo associar cada região geométrica ao material correspondente. A atribuição regional é realizada com base em pontos internos representativos, utilizados para identificar regiões fechadas delimitadas pela geometria.

Adicionalmente, a linha freática é convertida para uma representação compatível com a imposição de poropressão inicial, e as condições de contorno são traduzidas em entidades geométricas adequadas para aplicação de restrições.

A etapa é deliberadamente determinística: para um mesmo *schema*, o *builder* deve produzir o mesmo conjunto de entradas executáveis. Essa propriedade é essencial para reprodutibilidade e auditoria, especialmente quando se acopla uma etapa semântica a um procedimento numérico que exige controle e rastreabilidade.

3.4 Execução Numérica e Pós-Processamento: *Runner*

A etapa final do *pipeline* é implementada pelo *runner*, responsável por interagir com o software e executar, em sequência controlada, operações típicas do ciclo do MEF: construção do domínio, discretização e geração de malha, criação e atribuição de materiais, aplicação de gravidade e condições de contorno, imposição de poropressões iniciais e execução do procedimento SRF/FOS.

A imposição de poropressão inicial é tratada explicitamente, preservando coerência com a formulação de tensões efetivas e as hipóteses hidráulicas do problema. Em seguida, a análise SRF/FOS é conduzida como procedimento numérico para identificação de condição limite via redução progressiva de resistência, compatível com a prática de estabilidade em MEF e com modelos elasto-plásticos usuais. Após a solução, o *runner* aciona rotinas de pós-processamento para geração de visualizações de campos relevantes (tensões, deformações e indicadores de estabilidade), produzindo saídas consistentes.

Do ponto de vista metodológico, o produto do *pipeline* não é apenas o resultado final, mas um registro executável do caminho de geração e execução do modelo, sustentando a proposta de conectar linguagem natural à simulação sem perder rastreabilidade.

4 DEMONSTRAÇÃO FUNCIONAL: DA ESTRUTURA CONCEITUAL À EXECUÇÃO NUMÉRICA

A demonstração funcional foi concebida para verificar, em cenário controlado, a exequibilidade do fluxo ponta a ponta proposto: partir de uma descrição conceitual em linguagem natural, consolidá-la em uma representação formal (*ProjectSchema*) e, a partir dela, gerar e executar automaticamente um modelo numérico de estabilidade, com saídas gráficas e numéricas integradas. O foco recai sobre a robustez do encadeamento computacional e sobre a preservação de coerência física ao longo da tradução programática.

A etapa de execução numérica foi realizada no ADONIS, um programa gratuito de elementos finitos para aplicações geotécnicas disponibilizado pelo projeto GeoWizard (GeoWizard, 2026), que inclui rotinas de pré/pós-processamento e módulo de redução progressiva da resistência ao cisalhamento para análises de estabilidade (SRF/SSR).

Ressalta-se que esta demonstração não tem por objetivo avaliar sistematicamente o componente semântico (LLM) nem comparar desempenho com fluxos tradicionais. O objetivo é evidenciar que, uma vez formalizado o problema em um *schema* completo e verificável, o *pipeline* executa de forma determinística e reprodutível do pré-processamento ao pós-processamento, sem intervenção manual adicional.

4.1 Aquisição das Entradas via Linguagem Natural

Para tornar explícito o tipo de interação que alimenta a camada estruturada (*ProjectSchema*), apresenta-se a seguir uma transcrição ilustrativa de diálogo referente ao caso demonstrativo. O *listing* documenta o tipo de informação fornecida pelo usuário (geometria via segmentos extraídos de DXF, materiais, zoneamento, condições de contorno e linha freática) e como ela é organizada para posterior validação e execução.

Listing 1 – Transcrição ilustrativa (Usuário/GeoAssist AI) para preenchimento das entradas do caso demonstrativo.

Usuário: Quero rodar uma análise de estabilidade (SRF/FOS) no ADONIS usando o pipeline automático. Tenho um DXF com a seção do talude, de onde consegui extrair as coordenadas dos segmentos da geometria e também os pontos da linha freática. Também quero usar 3 materiais e modelo de Mohr-Coulomb.
GeoAssist AI: Perfeito.
Vamos por etapas: Geometria; Materiais; Distribuição por regiões; Condições de contorno; Freática; Configurações do solver.
Você quer fornecer a geometria como segmentos (x1,y1,x2,y2) exportados do DXF?
Usuário: Aqui estão os segmentos (do DXF):
(0.000, 0.000) ; (0.000, 60.000)
(114.208, 10.122) ; (111.787, 10.343) [..]
GeoAssist AI: Perfeito. Agora sobre os materiais: quantos materiais você quer e quais nomes você prefere?
Usuário: Quero 3 materiais: Solo Arenoso, Solo Argiloso e Rocha.
GeoAssist AI: Certo. Informe os parâmetros em unidades do SI: density (kg/m³), shear (Pa), bulk (Pa), coh (Pa), fric (°), dil (°), tens (Pa).
Usuário: Solo_Arenoso: density=2000.0, shear=5.0e7, bulk=1.0e8, coh=1.0e3, fric=35.0, dil=5.0, tens=0.0.
Solo_Argiloso: density=1900.0, shear=2.0e7, bulk=5.0e7, coh=2.0e4, fric=25.0, dil=0.0, tens=0.0.
Rocha: density=2400.0, shear=5.0e8, bulk=8.0e8, coh=2.0e5, fric=40.0, dil=0.0, tens=0.0.
GeoAssist AI: Agora vem a atribuição dos materiais. Você prefere por camadas (paramétrico) ou por pontos internos por região (mais seguro para geometria do DXF)?
Usuário: Vamos de pontos internos por região. Rocha: (10,2), (60,3), (110,2). Solo Argiloso: (10,20), (50,18), (90,15). Solo Arenoso: (10,55), (25,52), (40,50).
GeoAssist AI: Agora sobre condições de contorno: você quer fixar quais partes?

Usuário: base e lateral esquerda.
GeoAssist AI: Agora a linha freática: informe os pontos do DXF.
Usuário: wt_x = [0.000, 19.543, 25.938, 35.177, 42.994, 56.851, 72.485, 116.993]; wt_y = [30.000, ..]
GeoAssist AI: Vou gerar as entradas e executar SRF/FOS no ADONIS, com pós-processamento do SSI.

4.2 Representação estruturada: *ProjectSchema* simplificado

As informações coletadas são consolidadas em uma representação estruturada (*ProjectSchema*), utilizada como contrato entre intenção e execução. Para fins de transparência metodológica, apresenta-se abaixo um *schema* simplificado (ilustrativo) no estilo Pydantic, evidenciando os blocos de informação empregados nesta demonstração.

Listing 2 – ProjectSchema simplificado usado como contrato de entrada.

ProjectSchema = {geometry, materials, zoning, boundary_conditions, water_table, solver_config}.
Onde, *geometry* armazena segmentos extraídos de DXF; *materials*, a biblioteca Mohr–Coulomb; *zoning*, os pontos internos por região; *boundary_conditions*, as restrições geométricas; *water_table*, os pontos da polilinha freática; e *solver_config*, os parâmetros de execução no ADONIS.

4.3 Definição do Caso e Estrutura Física do Modelo

Para demonstrar a operacionalização do pipeline, conduziu-se uma análise de estabilidade de um talude estratificado em seção bidimensional, com aproximadamente 60 m de altura e inclinação global descendente da esquerda para a direita. A geometria foi definida explicitamente por segmentos de reta, permitindo representar quebras locais de inclinação e interfaces internas inclinadas entre unidades. A Figura 2 apresenta a geometria discretizada e a separação regional dos materiais após a geração automática da malha.

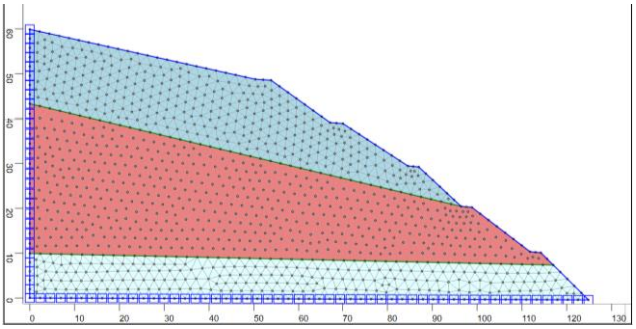


Figura 2. Geometria do talude, discretização triangular (T3) e distribuição dos materiais.

O domínio foi restringido na base e na lateral esquerda, eliminando modos de corpo rígido e assegurando estabilidade numérica. O modelo contempla três unidades geotécnicas (rocha, solo argiloso e solo arenoso), com atribuição regional realizada por pontos internos associados aos identificadores constitutivos, o que evita a necessidade de definição manual de polígonos e é compatível com geometrias oriundas de DXF.

4.4 Execução Numérica e Pós-Processamento: *Runner*

A linha freática foi definida como polilinha interpolada linearmente ao longo do talude, e o campo inicial de poropressão foi imposto sob hipótese hidrostática. O modelo foi resolvido no ADONIS com elementos triangulares lineares (T3), com geração automática de malha e aplicação de peso próprio via campo gravitacional uniforme. A estabilidade foi avaliada pelo método de redução de resistência (SRF/FOS), procedimento consolidado em análises por MEF para identificação de condição limite (Griffiths, Lane 1999; Dawson, Roth, Drescher 1999).

4.5 Resultados Numéricos e Padrão de Plastificação

Os contornos do índice de estabilidade (SSI) indicaram concentração de deformações ao longo do talude, com desenvolvimento de uma superfície potencial de ruptura iniciando em porções superiores e propagando-se em direção ao pé jusante. A distribuição das zonas plastificadas foi compatível com um mecanismo de ruptura rotacional composta, coerente com taludes estratificados sob carregamento gravitacional e influência de pressões neutras. A Figura 3 apresenta o contorno do índice de estabilidade obtido no pós-processamento.

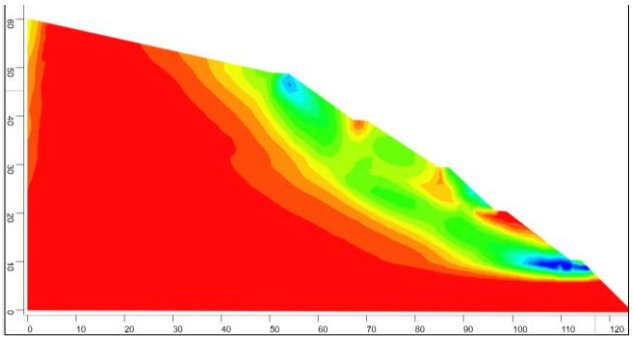


Figura 3. Contorno (SSI), com zonas de plastificação e desenvolvimento de superfície potencial de ruptura.

A região crítica localizou-se próxima ao contato entre a camada argilosa intermediária e a rocha subjacente, o que é mecanicamente plausível. De um lado, o contraste de rigidez tende a concentrar deformações e favorecer mobilização diferencial ao longo da interface. De outro, a menor resistência friccional da unidade intermediária constitui zona preferencial de plastificação, em especial quando tensões efetivas são reduzidas pela presença de poropressões associadas à linha freática (Terzaghi, Peck, Mesri 1996; Fredlund, Rahardjo 1993). Assim, o mecanismo observado se mantém consistente com a interpretação esperada para um problema com estratigrafia contrastante e regime hidráulico capaz de afetar tensões efetivas em regiões-chave do domínio.

4.6 Validação da Mediação Computacional

A evidência produzida por esta demonstração refere-se à consistência do encadeamento computacional entre a representação estruturada do problema e a execução numérica. A compatibilidade entre um mecanismo geotécnico esperado (ruptura composta influenciada por estratigrafia e poropressão) e a resposta observada (plastificação em regiões mecanicamente consistentes) sugere que a tradução programática não introduziu alterações conceituais relevantes ao converter o *schema* em comandos do *solver*. Essa verificação é condição necessária para escalabilidade: antes de ampliar complexidade constitutiva, hidráulica ou geométrica, é importante assegurar que o encadeamento formal preserve a intenção do problema.

Do ponto de vista de engenharia de software, o *schema* intermediário estabelece uma camada auditável entre intenção e execução, reduzindo suscetibilidade a erros típicos de configuração manual (por exemplo, inconsistências de unidade, omissões e associações incorretas de propriedades) e contribuindo para reprodutibilidade ao registrar o estado do problema como dados estruturados. Embora esta demonstração não quantifique ganhos de produtividade nem avalie sistematicamente o componente semântico (LLM), os resultados apresentados sustentam a premissa central deste estudo: é possível conduzir, de forma automatizada e reprodutível, a transição entre uma descrição estruturada e a execução numérica com pós-processamento integrado, criando base para avaliações futuras de robustez semântica e ampliação do repertório constitutivo e hidráulico.

5 CONCLUSÃO

Este trabalho apresentou o GeoAssist AI como uma arquitetura de mediação entre modelos de linguagem natural (LLMs) e modelagem numérica geotécnica por MEF, implementada em Python e acoplada a um ambiente de simulação. O objetivo foi formalizar e demonstrar um fluxo capaz de transformar descrições técnicas em uma representação estruturada e, a partir dela, executar de forma automatizada uma análise numérica, preservando coerência física e rastreabilidade do processo.

A contribuição principal é metodológica: a introdução de um *schema* intermediário como contrato formal entre a intenção do usuário e a execução computacional. Ao separar explicitamente a etapa de interpretação/extração, potencialmente mediada por LLMs, da etapa de execução numérica determinística, a abordagem favorece auditabilidade e reprodutibilidade do caminho de geração do modelo, além de reduzir pontos recorrentes de falha associados ao pré-processamento manual.

A demonstração funcional confirmou a exequibilidade ponta a ponta do *pipeline* no contexto de um talude estratificado 2D, com geometria definida por linhas, materiais Mohr–Coulomb, condições de contorno e linha freática. O modelo foi construído e resolvido automaticamente por SRF/FOS e produziu saídas compatíveis com interpretação geotécnica: reconstrução geométrica consistente com o *schema*, atribuição regional de materiais conforme definido, imposição hidrostática de poropressão coerente com a linha freática e um padrão de plastificação/ruptura mecanicamente plausível diante do contraste de rigidez e resistência entre camadas e do regime hidráulico adotado.

Por fim, delimita-se o escopo do estudo: não foram quantificados ganhos de produtividade, nem avaliada de forma sistemática a acurácia do componente semântico sob variações linguísticas, tampouco conduzidas comparações com fluxos tradicionais ou soluções de referência. Assim, os resultados devem ser interpretados como evidência de viabilidade técnica do encadeamento formal proposto e de consistência entre entradas estruturadas e execução numérica, e não como validação abrangente de desempenho.

REFERÊNCIAS BIBLIOGRÁFICAS

Biot, M. A. (1941) General theory of three-dimensional consolidation. Journal of Applied Physics, 12 (2), p. 155–164.

Bommasani, R.; Hudson, D. A.; Adeli, E.; Altman, R.; Arora, S.; von Arx, S.; Bernstein, M. S.; et al. (2021) On the opportunities and risks of foundation models. arXiv preprint arXiv:2108.07258, p. 1–137. https://doi.org/10.48550/arXiv.2108.07258

Chen, M.; Tworek, J.; Jun, H.; Yuan, Q.; de Oliveira Pinto, H. P.; Kaplan, J.; Edwards, H.; et al. (2021) Evaluating large language models trained on code. arXiv preprint arXiv:2107.03374, p. 1–37. https://doi.org/10.48550/arXiv.2107.03374

Dawson, E. M.; Roth, W. H.; Drescher, A. (1999) Slope stability analysis by strength reduction. Géotechnique, 49 (6), p. 835–840. https://doi.org/10.1680/geot.1999.49.6.835

Fredlund, D. G.; Rahardjo, H. (1993) Soil mechanics for unsaturated soils. John Wiley & Sons, New York.

GeoWizard (2026). ADONIS: Free finite element program for Geo-Engineers. Disponível em: <https://www.geowizard.org/download_adonis.html>. Acesso em: 10 jun. 2026.

Griffiths, D. V.; Lane, P. A. (1999) Slope stability analysis by finite elements. Géotechnique, 49 (3), p. 387–403. https://doi.org/10.1680/geot.1999.49.3.387

Potts, D. M.; Zdravkovic, L. (1999) Finite element analysis in geotechnical engineering: theory. Thomas Telford, London.

Terzaghi, K.; Peck, R. B.; Mesri, G. (1996) Soil mechanics in engineering practice. 3rd ed., John Wiley & Sons, New York.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; Polosukhin, I. (2017) Attention is all you need. Advances in Neural Information Processing Systems, 30, p. 5998–6008.

Zienkiewicz, O. C.; Taylor, R. L.; Zhu, J. Z. (2013) The finite element method: its basis and fundamentals. 7th ed., Elsevier.