



**UNIVERSIDADE FEDERAL DO CARIRI
CENTRO DE CIÊNCIAS E TECNOLOGIA
CURSO DE GRADUAÇÃO EM ENGENHARIA CIVIL**

FIDEL MORAIS LUCAS

**IMPLEMENTAÇÃO COMPUTACIONAL DO MÉTODO DA RIGIDEZ DIRETA PARA
ANÁLISE DE PÓRTICOS PLANOS COM BARRAS DE INÉRCIA VARIÁVEL**

JUAZEIRO DO NORTE

2025

FIDEL MORAIS LUCAS

IMPLEMENTAÇÃO COMPUTACIONAL DO MÉTODO DA RIGIDEZ DIRETA PARA
ANÁLISE DE PÓRTICOS PLANOS COM BARRAS DE INÉRCIA VARIÁVEL

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia Civil do Centro de Ciências e Tecnologia da Universidade Federal do Cariri, como requisito parcial à obtenção do grau de bacharel em Engenharia Civil.

Orientador: Prof. Dr. Francisco José Pereira de Almeida

JUAZEIRO DO NORTE

2025

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Cariri
Sistema de Bibliotecas

P314c Lucas, Fidel Morais.
Implementação computacional do método da rigidez direta para análise de pórticos planos com barras de inércia variável utilizando Python / Fidel Morais Lucas. – 2025.
64 f. : il. color.

Trabalho de Conclusão de Curso (Graduação em Engenharia Civil) – Centro de Ciências e Tecnologia, Universidade Federal do Cariri, Juazeiro do Norte, 2025.

Orientador: Prof. Dr. Francisco José Pereira de Almeida.

1. Análise estrutural. 2. Método da Rigidez Direta. 3. Pórticos planos. 4. Python. 5. Seção variável. I. Almeida, Francisco José Pereira de (Orient.). II. Universidade Federal do Cariri. III. Título.

CDD 624.171



MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO CARIRI
CENTRO DE CIÊNCIAS E TECNOLOGIA
COORDENAÇÃO DO CURSO DE ENGENHARIA CIVIL


Implementação Computacional do Método da Rigidez Direta para Análise de Pórticos Planos com Barras de Inércia Variável Utilizando PYTHON

Fidel Moraes Lucas


Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia Civil do Centro de Ciências e Tecnologia da Universidade Federal do Cariri, como requisito parcial para obtenção do Título de Bacharel em Engenharia Civil. Área de concentração: Engenharia Civil.


Aprovada em 30 de março de 2026.

Banca Examinadora

Documento assinado digitalmente
 FRANCISCO JOSE PEREIRA DE ALMEIDA
Data: 01/04/2026 13:04:31-0300
Verifique em <https://validar.iti.gov.br>

Francisco José Pereira de Almeida
Orientador e Presidente da banca

Documento assinado digitalmente
 CLEIRTON ANDRE SILVA DE FREITAS
Data: 01/04/2026 09:10:09-0300
Verifique em <https://validar.iti.gov.br>

Documento assinado digitalmente
 FRANCISCO ALEX CORREIA MONTEIRO
Data: 01/04/2026 12:33:29-0300
Verifique em <https://validar.iti.gov.br>

Cleirton André Silva de Freitas
1º Membro Examinador

Francisco Alex Correia Monteiro
2º Membro Examinador

A minha mãe, Brígida, ao meu pai, Alexandre e a minha namorada, Mirelly, pelo amor incondicional e por acreditarem em mim mesmo quando eu duvidei.

AGRADECIMENTOS

Agradeço primeiramente a Deus, que, mesmo diante das minhas falhas, sempre se fez presente nesta jornada e que, com Sua gloriosa onisciência, colocou as pessoas certas, nos exatos momentos, para me guiarem rumo ao que posso chamar, verdadeiramente, de sucesso — algo mais profundo do que um fútil poder aquisitivo.

Agradeço também à minha família, alicerce fundamental da vida humana, sem a qual nada disso seria possível. Nesse sentido, gostaria de fazer um agradecimento especial à minha namorada, Mirelly; à minha mãe, Brígida; ao meu pai, Alexandre; e à minha tia Neide.

Durante minha jornada de formação acadêmica, tive também o privilégio de cruzar com pessoas que me marcaram profundamente e que não poderia deixar de citar de modo honroso: Tia Mariza, Tia Lisiene, Alana, Jeffinho, Luisinho, Alisson, Aline, Marcos Venício, Tia Ana Paula, Marcela, Marta, Betânia, Gilson, Tia Kátia, Tia Josélia, Guilherme, Helaine, Paulo Roberto, Luiz Alberto, Francisco José, Amanda, Silvana, Valdir, Antônio Carlos, Hermínio, Gorethe, André, Flávio, Patrick e Jordana.

Por fim, faço um agradecimento aos meus caros colegas que se fizeram presentes nesta jornada, em especial aos meus amigos Breno Silva e Eron Pinheiro.

‘Faz o que deves e está no que fazes.’

(São Josemaria Escrivá)

RESUMO

A análise estrutural de pórticos planos é uma área fundamental da engenharia estrutural que tem evoluído significativamente com o advento da computação científica. Este trabalho apresenta o desenvolvimento de uma aplicação computacional para análise de pórticos planos utilizando o Método da Rigidez Direta, implementado na linguagem de programação Python. O objetivo principal consiste em criar uma ferramenta computacional eficiente e acessível que permita a análise estrutural de pórticos planos com elementos de seção constante e variável, fornecendo resultados precisos para deslocamentos nodais, esforços internos e reações de apoio. A metodologia adotada fundamenta-se na implementação sistemática do Método da Rigidez Direta, considerando a formulação matricial clássica da análise estrutural. O desenvolvimento da aplicação utiliza as bibliotecas científicas NumPy e SciPy para operações matriciais e resolução de sistemas lineares, enquanto a interface gráfica é construída com a biblioteca Tkinter, proporcionando uma experiência intuitiva ao usuário. Uma característica diferencial da aplicação é a capacidade de analisar elementos com seção transversal variável através da implementação de técnicas de integração numérica, ampliando significativamente as possibilidades de modelagem estrutural. O referencial teórico abrange os fundamentos da análise matricial de estruturas, com ênfase no Método da Rigidez Direta, incluindo a formulação das matrizes de rigidez elementares, procedimentos de transformação de coordenadas, montagem da matriz de rigidez global e aplicação das condições de contorno. O trabalho também apresenta a fundamentação matemática para tratamento de elementos de seção variável, utilizando o Método das Flexibilidades com integração numérica. A validação da ferramenta será realizada através da comparação com soluções analíticas conhecidas e resultados obtidos por softwares comerciais consolidados, garantindo a precisão e confiabilidade dos cálculos implementados. Espera-se que esta aplicação contribua tanto para o ensino da análise estrutural quanto para aplicações práticas em escritórios de cálculo estrutural, oferecendo uma alternativa gratuita e didática aos softwares comerciais existentes.

Palavras-chave: Análise estrutural. Método da Rigidez Direta. Pórticos planos. Python. Seção variável. Análise matricial. Engenharia computacional.

ABSTRACT

The structural analysis of plane frames is a fundamental area of structural engineering that has evolved significantly with the advent of scientific computing. This work presents the development of a computational application for plane frame analysis using the Direct Stiffness Method, implemented in the Python programming language. The main objective is to create an efficient and accessible computational tool that allows structural analysis of plane frames with constant and variable cross-section elements, providing accurate results for nodal displacements, internal forces, and support reactions. The adopted methodology is based on the systematic implementation of the Direct Stiffness Method, considering the classical matrix formulation of structural analysis. The application development uses the scientific libraries NumPy and SciPy for matrix operations and linear system solving, while the graphical interface is built with the Tkinter library, providing an intuitive user experience. A distinctive feature of the application is the ability to analyze elements with variable cross-sections through the implementation of numerical integration techniques, significantly expanding the structural modeling possibilities. The theoretical framework covers the fundamentals of matrix structural analysis, with emphasis on the Direct Stiffness Method, including the formulation of elemental stiffness matrices, coordinate transformation procedures, global stiffness matrix assembly, and boundary condition application. The work also presents the mathematical foundation for treating variable cross-section elements, using the Flexibility Method with numerical integration. Tool validation will be performed through comparison with known analytical solutions and results obtained by established commercial software, ensuring the accuracy and reliability of the implemented calculations. It is expected that this application will contribute both to structural analysis teaching and practical applications in structural design offices, offering a free and educational alternative to existing commercial textitsoftware.

Keywords: Structural analysis. Direct Stiffness Method. Plane frames. Python. Variable cross-section. Matrix analysis. Computational engineering.

LISTA DE FIGURAS

Figura 1 – Orientação dos deslocamentos e das forças nodais.	21
Figura 2 – Graus de liberdade nodais	23
Figura 3 – Rotação do elemento de barra.	24
Figura 4 – Expressões tabuladas para método da rigidez	26
Figura 5 – Exemplo de tabelas da Portland Cement Association (<i>Portland Cement Association</i> — associação americana de cimento (PCA)) para barras de seção variável.	27
Figura 6 – Fluxograma geral do funcionamento da aplicação baseada no Método da Rigidez Direta (MRD).	31
Figura 7 – Diagrama de arquitetura do <i>software</i> PorticoAnalyzer.	33
Figura 8 – Definição da classe Material em Python.	34
Figura 9 – Definição da classe Node em Python.	35
Figura 10 – Definição da classe Element em Python.	35
Figura 11 – Representação esquemática de um elemento de barra com seção variável. . .	36
Figura 12 – Fluxograma do algoritmo de cálculo da matriz de rigidez local.	38
Figura 13 – Elemento de viga com seção variável submetido a carregamento distribuído. .	39
Figura 14 – Interface principal do <i>software</i> PorticoAnalyzer.	42
Figura 15 – Visualização dos diagramas de esforços internos.	43
Figura 16 – Fluxograma geral do funcionamento da aplicação.	45
Figura 17 – Esquema estrutural do Exemplo 1: viga biengastada de seção constante. . .	47
Figura 18 – Plotagem do Exemplo 1 no programa.	47
Figura 19 – Diagramas obtidos para o Exemplo 1.	48
Figura 20 – Esquema estrutural do Exemplo 2: viga biengastada com seção trapezoidal variável.	49
Figura 21 – Plotagem do Exemplo 2 no programa.	50
Figura 22 – Plotagem dos diagramas do Exemplo 2 no programa.	51
Figura 23 – Esquema estrutural do Exemplo 3: pórtico plano completo.	53
Figura 24 – Plotagem do Exemplo 3 no programa.	53
Figura 25 – Plotagem dos diagramas do Exemplo 3 no programa.	55

LISTA DE TABELAS

Tabela 1 – Cronograma de atividades do projeto	32
Tabela 2 – Dados de entrada do Exemplo 1.	48
Tabela 3 – Comparação de resultados para o Exemplo 1 (seção constante).	49
Tabela 4 – Comparação de resultados para o Exemplo 2 (seção variável trapezoidal).	51
Tabela 5 – Comparação de resultados para o Exemplo 3 (pórtico completo).	54
Tabela 6 – Síntese dos erros de validação.	55
Tabela 7 – Tempo de execução vs. tamanho da estrutura.	56

LISTA DE ABREVIATURAS E SIGLAS

ANSYS	<i>Analysis System</i> — textitsoftware comercial de simulação multifísica
BIM	Modelagem da Informação da Construção (<i>Building Information Modeling</i>)
FEM	<i>Finite Element Method</i> / Método dos Elementos Finitos
FEniCS	<i>Finite Element Computational textitSoftware</i> — plataforma open-source de elementos finitos
GPU	<i>Graphics Processing Unit</i> / Unidade de Processamento Gráfico
JAX-FEM	<i>JAX-based Finite Element Method</i> — framework de elementos finitos diferenciável
NumPy	<i>Numerical Python</i> — biblioteca para operações numéricas e matriciais
OpenMDAO	<i>Multidisciplinary Design, Analysis, and Optimization</i> — framework open-source da NASA
OpenSees	<i>Open System for Earthquake Engineering Simulation</i> — textitsoftware de simulação sísmica
OpenSeesPy	interface em Python do <i>Open System for Earthquake Engineering Simulation</i> — textitsoftware de simulação sísmica (OpenSees)
PCA	<i>Portland Cement Association</i> — associação americana de cimento
Plotly	biblioteca para criação de gráficos interativos, com suporte a Python
PyNite	<i>Python Nonlinear Integrated Tool for Engineers</i> — biblioteca Python de análise estrutural
SAP2000	<i>Structural Analysis Program 2000</i> — textitsoftware de análise estrutural da CSI
SciPy	<i>Scientific Python</i> — biblioteca para rotinas científicas e métodos numéricos
Streamlit	biblioteca Python para criação de aplicações web interativas voltadas a ciência de dados
TQS	textitsoftware brasileiro para cálculo de estruturas de concreto armado, da TQS Informática

LISTA DE SÍMBOLOS

A	Área da seção transversal do elemento (m^2)
a	Distância do ponto de aplicação da carga em relação ao apoio (m)
b	Distância complementar do ponto de aplicação da carga (m)
d	Deslocamento nodal (m)
E	Módulo de elasticidade do material (Pa)
F	Vetor de forças nodais (N)
I	Momento de inércia da seção transversal (m^4)
K	Matriz de rigidez (N/m)
k_{ij}	Coefficiente de rigidez (N/m)
L	Comprimento do elemento/barra (m)
M	Momento fletor (N·m)
N	Força normal (N)
P	Carga concentrada/força aplicada (N)
q	Carga distribuída (N/m)
θ	Ângulo de rotação (rad)
u	Deslocamento horizontal do nó (m)
v	Deslocamento vertical do nó (m)
δ	Vetor de deslocamentos nodais (m)
τ	Matriz de transformação de coordenadas
λ	Comprimento de onda (se usado em dinâmica/opcional)
ϕ	Rotação nodal (rad)

SUMÁRIO

1	INTRODUÇÃO	16
2	OBJETIVOS	19
2.1	Objetivo Geral	19
2.2	Objetivos Específicos	19
3	REFERENCIAL TEÓRICO	20
3.1	Análise Matricial de Estruturas	20
3.2	Método da Rigidez Direta (MRD)	20
3.2.1	<i>Grau de Liberdade e Numeração Nodal</i>	21
3.2.2	<i>Montagem da Matriz de Rigidez do elemento</i>	22
3.2.3	<i>Matriz de Transformação (T) (Rotação de Eixos)</i>	23
3.2.4	<i>Matriz de Rigidez Global</i>	24
3.2.5	<i>Vetor de cargas nodais combinadas</i>	25
3.3	Modelagem de Elementos de Barra de Seção Variável	26
3.4	Cálculo dos deslocamentos nodais e esforços de extremidade de elementos	28
3.5	Implementação Computacional com Python	28
4	METODOLOGIA	30
4.1	Funcionamento da aplicação	31
5	CRONOGRAMA	32
6	DESCRIÇÃO DO SOFTWARE	33
6.1	Arquitetura do Sistema	33
6.2	Estrutura de Dados	34
6.2.1	<i>Classe Material</i>	34
6.2.2	<i>Classe Node</i>	34
6.2.3	<i>Classe Element</i>	35
6.3	Algoritmo de Cálculo da Matriz de Rigidez	36
6.3.1	<i>Elementos de Seção Constante</i>	36
6.3.2	<i>Elementos de Seção Variável</i>	37
6.3.3	<i>Momentos de Engastamento Perfeito para Seção Variável</i>	39
6.4	Montagem e Resolução do Sistema Global	40
6.4.1	<i>Matriz de Transformação</i>	40

6.4.2	<i>Montagem da Matriz de Rigidez Global</i>	41
6.4.3	<i>Aplicação das Condições de Contorno</i>	41
6.4.4	<i>Resolução do Sistema Linear</i>	41
6.5	Interface Gráfica do Usuário	42
6.5.1	<i>Organização da Interface</i>	42
6.5.2	<i>Visualização de Resultados</i>	43
6.5.3	<i>Módulo de Exemplos de Validação</i>	43
6.6	Fluxograma Geral do Software	44
7	RESULTADOS E DISCUSSÃO	46
7.1	Metodologia de Validação	46
7.2	Exemplo 1: Viga Biengastada de Seção Constante	46
7.2.1	<i>Descrição do Problema</i>	46
7.2.2	<i>Valores Esperados</i>	47
7.2.3	<i>Resultados Obtidos</i>	48
7.3	Exemplo 2: Viga Biengastada com Seção Trapezoidal Variável	49
7.3.1	<i>Descrição do Problema</i>	49
7.3.2	<i>Valores de Referência</i>	50
7.3.3	<i>Resultados Obtidos</i>	51
7.3.4	<i>Discussão dos Resultados</i>	51
7.4	Exemplo 3: Pórtico Plano Completo	52
7.4.1	<i>Descrição do Problema</i>	52
7.4.2	<i>Valores de Referência</i>	53
7.4.3	<i>Resultados Obtidos</i>	54
7.5	Síntese dos Resultados de Validação	54
7.6	Análise de Desempenho Computacional	55
8	CONCLUSÕES E TRABALHOS FUTUROS	57
8.1	Conclusões	57
8.1.1	<i>Quanto à Implementação do Algoritmo</i>	57
8.1.2	<i>Quanto à Interface Gráfica</i>	58
8.1.3	<i>Quanto à Validação</i>	58
8.1.4	<i>Quanto ao Caráter Didático e de Código Aberto</i>	59
8.2	Limitações do Trabalho	59

8.3	Sugestões para Trabalhos Futuros	60
8.3.1	<i>Extensões Técnicas</i>	60
8.3.2	<i>Melhorias na Interface</i>	60
8.3.3	<i>Validações Adicionais</i>	61
8.3.4	<i>Integração com Outras Ferramentas</i>	61
8.4	Considerações Finais	61
	REFERÊNCIAS	63

1 INTRODUÇÃO

A prática contemporânea da engenharia estrutural está profundamente vinculada ao avanço da capacidade computacional. A crescente complexidade dos sistemas estruturais modernos exige análises que ultrapassam a capacidade do cálculo manual tradicional, solidificando o emprego de *softwares* especializados e exemplo do *Structural Analysis Program 2000* — *textitsoftware* de análise estrutural da CSI (SAP2000), do Eberick, do *textitsoftware* brasileiro para cálculo de estruturas de concreto armado, da TQS Informática (TQS) e do *Analysis System* — *textitsoftware* comercial de simulação multifísica (ANSYS) como instrumentos indispensáveis na indústria. Estas ferramentas automatizam rotinas de cálculo intensivas e oferecem visualizações detalhadas do comportamento estrutural, resultando em avanços qualitativos substanciais em eficiência e em precisão no processo de concepção (MARTHA, 2010).

Todavia, o uso generalizado desses programas pode, inadvertidamente, conduzir ao erro do usuário dado que funcionam como “caixas-pretas”, de modo que se torna imperativa a internalização, pelo usuário, dos fundamentos teóricos e algorítmicos que sustentam o funcionamento dessas aplicações. No cerne da maioria desses *softwares* comerciais — especialmente em estruturas reticuladas — está o Método da Rigidez Direta (*Direct Stiffness Method*). Esse método matricial, que relaciona forças, deslocamentos e propriedades dos materiais, mostrou-se excepcionalmente adequado à implementação computacional, consolidando-se como a técnica predominante na análise estrutural desde meados do século XX; constitui, ainda hoje, a base da grande maioria dos solucionadores de elementos estruturais (GERE; WEAVER, 1987).

Ademais, embora as ferramentas comerciais apresentem considerável robustez técnica, elas frequentemente impõem limitações substanciais relacionadas ao custo elevado, à rigidez operacional e à falta de transparência nos algoritmos implementados. Diante desse cenário, pesquisadores, estudantes e engenheiros que enfrentam problemas não convencionais ou que demandam controle completo sobre os processos analíticos encontram na criação de soluções computacionais personalizadas uma vantagem estratégica fundamental (KASSIMALI, 2011).

Além disso, é comum encontrar elementos com seções transversais variáveis em estruturas reais, fato que é resultado de requisitos como otimização de peso, adaptação arquitetônica ou rigidez localizada. Desse modo, a análise precisa desses elementos introduz complexidades matemáticas elevadas, sobretudo pela variação contínua de propriedades geométricas como momento de inércia. Critica-se, por isso, abordagens simplificadas — a exemplo das propostas por manuais clássicos da PCA — por desconsiderarem efeitos importantes como os provenientes

das variações reais de rigidez, o que pode comprometer significativamente os resultados do projeto. Tais críticas são detalhadamente discutidas por (COLUNGA, 2007), que propôs uma formulação mais rigorosa baseada na teoria clássica de vigas de Bernoulli-Euler, incorporando deformações axiais e transversais e variabilidade geométrica.

Nesse contexto, a linguagem de programação Python emerge como uma plataforma computacional de excelência para aplicações em engenharia estrutural. Dado que traz, consigo, a combinação entre simplicidade sintática e versatilidade funcional, aliada ao seu rico ecossistema de bibliotecas científicas — a exemplo da *Numerical Python* — biblioteca para operações numéricas e matriciais (NumPy) para operações matriciais de alto desempenho e da *Scientific Python* — biblioteca para rotinas científicas e métodos numéricos (SciPy) para rotinas numéricas avançadas, de modo a reduzir, significativamente, a complexidade de implementação de algoritmos sofisticados, como o Método da Rigidez Direta. Concomitante a isso, sua natureza de código aberto converge com os princípios fundamentais da colaboração acadêmica e do compartilhamento do conhecimento, elementos cruciais para um desenvolvimento científico aberto ao público e a possíveis melhorias e adaptações (LANGTANGEN, 2016; LUTZ, 2013).

Dessa maneira, percebe-se a existência de diversos projetos recentes que ilustram a preferência pela linguagem de programação adotada neste trabalho em desenvolvimentos computacionais. Dessa forma, (FERREIRA; SAMPAIO, 2024) desenvolveram uma ferramenta *open-source* em Python para análise estática de estruturas reticuladas em regimes lineares tridimensionais, com precisão que atinge erros da ordem de 10^{-7} a 10^{-5} . Para análises dinâmicas, publicaram também uma versão com integração de Newmark e modelagem modal com erros entre 10^{-5} e 10^{-2} . No âmbito computacional avançado, destaca-se *JAX-based Finite Element Method* — framework de elementos finitos diferenciável (JAX-FEM) (XUE *et al.*, 2022), um *solver Finite Element Method* / Método dos Elementos Finitos (FEM) (Solucionador do Método dos Elementos Finitos) diferenciável, acelerado por *Graphics Processing Unit* / Unidade de Processamento Gráfico (GPU) e implementado com a mesma estrutura de código supracitada, o qual oferece soluções diretas e inversas com eficiência até $10\times$ superior a códigos comerciais, incorporando diferenciação automática para otimização e aprendizado de máquina. Outras ferramentas, como GOLDFISH (ZHAO *et al.*, 2025), baseadas em *Finite Element Computational textitSoftware* — plataforma *open-source* de elementos finitos (FEniCS) e *Multidisciplinary Design, Analysis, and Optimization* — framework *open-source* da NASA (OpenMDAO), permitem otimização topológica de cascas usando análise isogeométrica e *design* orientado a gradientes. A plataforma

FEniCS, amplamente reconhecida, automatiza soluções variacionais de equações diferenciais parciais em Python e C++, sendo consolidada como referência no meio acadêmico (LOGG *et al.*, 2012). Um panorama recente de bibliotecas Python para engenharia estrutural inclui *Python Nonlinear Integrated Tool for Engineers* — biblioteca Python de análise estrutural (PyNite), AnaStruct, *sectionproperties* e interface em Python do OpenSees (OpenSees), consolidando um ecossistema de código aberto robusto para a área.

Diante desse contexto, o presente trabalho propõe harmonizar as demandas por transparência algorítmica, flexibilidade operacional e baixo custo com os recursos técnicos robustos oferecidos pela linguagem Python. O objetivo principal consiste no desenvolvimento de um programa de código aberto para análise estrutural de pórticos planos, implementando o Método da Rigidez Direta de maneira explícita e pedagogicamente orientada. Busca-se, dessa forma, não apenas conceber uma ferramenta computacional acessível e funcional, mas também produzir material didático que torne transparente a análise estrutural computacional e fomente a inovação colaborativa na comunidade de engenharia estrutural (STALLMAN, 2002).

2 OBJETIVOS

2.1 Objetivo Geral

Desenvolver um textitsoftware de código aberto, com interface gráfica intuitiva, para a análise de pórticos planos pelo Método da Rigidez Direta, visando criar uma ferramenta funcional, precisa e acessível para fins acadêmicos e profissionais.

2.2 Objetivos Específicos

1. **Implementar o Algoritmo de Análise:** Desenvolver, na linguagem de programação Python, o núcleo de processamento do textitsoftware, implementando o algoritmo do Método da Rigidez Direta para o cálculo de deslocamentos e esforços internos em pórticos planos com barras de inércia variável.
2. **Construir a Interface Gráfica (GUI):** Criar uma interface gráfica de usuário funcional e de fácil utilização, que permita a modelagem da estrutura (entrada de nós, elementos, cargas e apoios) e a visualização clara dos resultados (diagramas de esforços e estrutura deformada) de forma interativa e local, sem a necessidade de processos complexos de instalação ou deploy.
3. **Verificar e Validar a Ferramenta:** Submeter o textitsoftware a um rigoroso processo de validação, comparando os resultados obtidos em exemplos práticos com aqueles fornecidos por softwares consagrados no mercado, como o Ftool e o SAP2000, a fim de garantir sua precisão e confiabilidade.
4. **Elaborar Documentação de Suporte:** Produzir uma documentação técnica clara e um guia de utilização do textitsoftware, com o objetivo de facilitar o uso, a manutenção e futuras modificações por outros estudantes, pesquisadores e profissionais da área, reforçando o caráter colaborativo e de conhecimento aberto do projeto.

3 REFERENCIAL TEÓRICO

3.1 Análise Matricial de Estruturas

A análise matricial de estruturas consolidou-se como uma abordagem indispensável para engenheiros, de modo a oferecer um método abrangente para a resolução de uma vasta gama de tipologias estruturais. Dessa forma, a principal vantagem dessa metodologia reside na sua generalidade, em contraste com métodos tradicionais que frequentemente requerem abordagens distintas para cada tipo de estrutura. A formulação matricial fornece uma maneira eficiente de descrever os passos da análise, sendo naturalmente adequada para programação computacional, uma vez que as matrizes permitem a manipulação de grandes conjuntos de dados de forma simples e eficaz (GERE; WEAVER, 1987).

As estruturas reticuladas, objeto central deste estudo, caracterizam-se por elementos de barra dotados de eixo longitudinal bem definido. Mesmo em sistemas estruturais complexos, como edifícios em concreto armado, frequentemente a análise do comportamento global ou parcial é conduzida mediante modelos de barras (MARTHA, 2010). Neste contexto, as estruturas podem ser categorizadas como articuladas, cujos nós não proporcionam transmissão de momentos, e reticuladas (ou de nós rígidos), onde as conexões nodais restringem deslocamentos e rotações relativas, viabilizando a transferência de momentos fletores entre os elementos estruturais (DÍAZ *et al.*, 2015).

3.2 Método da Rigidez Direta (MRD)

Dentre os métodos matriciais, o Método da Rigidez Direta (MRD) é reconhecido como uma das abordagens mais robustas e mais abrangentes para análise estrutural computacional. Sua formulação sistemática e consistente o torna especialmente adequado à implementação computacional, constituindo o fundamento da maioria dos softwares de análise estrutural contemporâneos (GERE; WEAVER, 1987).

Dessa maneira, o método fundamenta-se na relação de equilíbrio entre as forças nodais aplicadas $\{F\}$ e os deslocamentos nodais desconhecidos $\{D\}$, por meio da matriz de rigidez global $[K]$ da estrutura, expressa pela equação fundamental:

$$[K]\{D\} = \{F\} \quad (3.1)$$

A análise considera um comportamento elástico-linear do material e a teoria de pequenas deformações (MARTHA, 2010).

3.2.1 Grau de Liberdade e Numeração Nodal

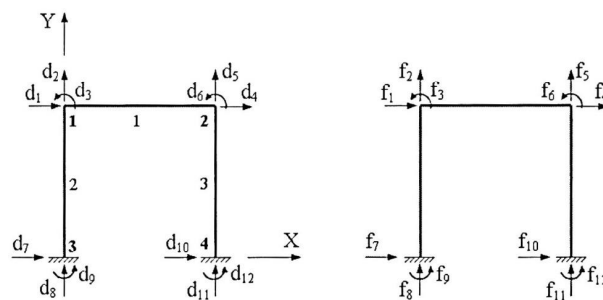
No contexto do Método da Rigidez Direta, os deslocamentos possíveis em um sistema estrutural são representados pelos chamados graus de liberdade. Em pórticos planos, cada nó possui três graus de liberdade: dois translacionais no plano da estrutura e um rotacional perpendicular a esse plano. Esses deslocamentos, assim como as forças associadas, são organizados em vetores e numerados sequencialmente de acordo com os nós da estrutura (COLUNGA, 2007).

Cada elemento da matriz $[K]$, representado por k_{ij} , corresponde a um coeficiente de rigidez, responsável por quantificar a força necessária para produzir uma unidade de deslocamento em uma determinada direção, mantendo os demais deslocamentos nulos.

Segundo Martha (MARTHA, 2010), qualquer movimento viável dentro de uma estrutura constitui um grau de liberdade do sistema. Para estruturas de pórticos bidimensionais, verifica-se que cada ponto nodal apresenta três possibilidades de movimento independentes: duas relacionadas aos movimentos de translação dentro do plano estrutural e uma terceira associada ao movimento rotacional ao redor de um eixo ortogonal a este mesmo plano.

A sistematização numérica destes movimentos e das respectivas cargas nodais correspondentes segue uma ordenação progressiva, iniciando-se pelo primeiro nó da estrutura, conforme ilustrado no exemplo da Figura 1 (SORIANO; LIMA, 2005).

Figura 1 – Orientação dos deslocamentos e das forças nodais.



Fonte: (SORIANO; LIMA, 2005)

Nota: Onde d_i é o deslocamento na direção do grau de liberdade i ; f_i é a força nodal aplicada na direção do grau de liberdade i .

3.2.2 Montagem da Matriz de Rigidez do elemento

Conforme estabelecido por Martha (2010), os componentes da matriz $[K]$ são denominados coeficientes de rigidez k_{ij} e quantificam a magnitude do esforço na direção do grau de liberdade i associado ao deslocamento unitário na direção do grau de liberdade j , mantendo-se todos os outros movimentos restringidos. Essa é a definição de coeficiente de rigidez, que pode ser classificado como direto (quando $i = j$) ou cruzado (quando $i \neq j$).

Ilustrando através de uma configuração elementar composta por dois pontos nodais interligados por um elemento linear, onde cada nó apresenta três possibilidades de movimento (resultando em seis graus de liberdade totais), o equacionamento assume a seguinte estrutura:

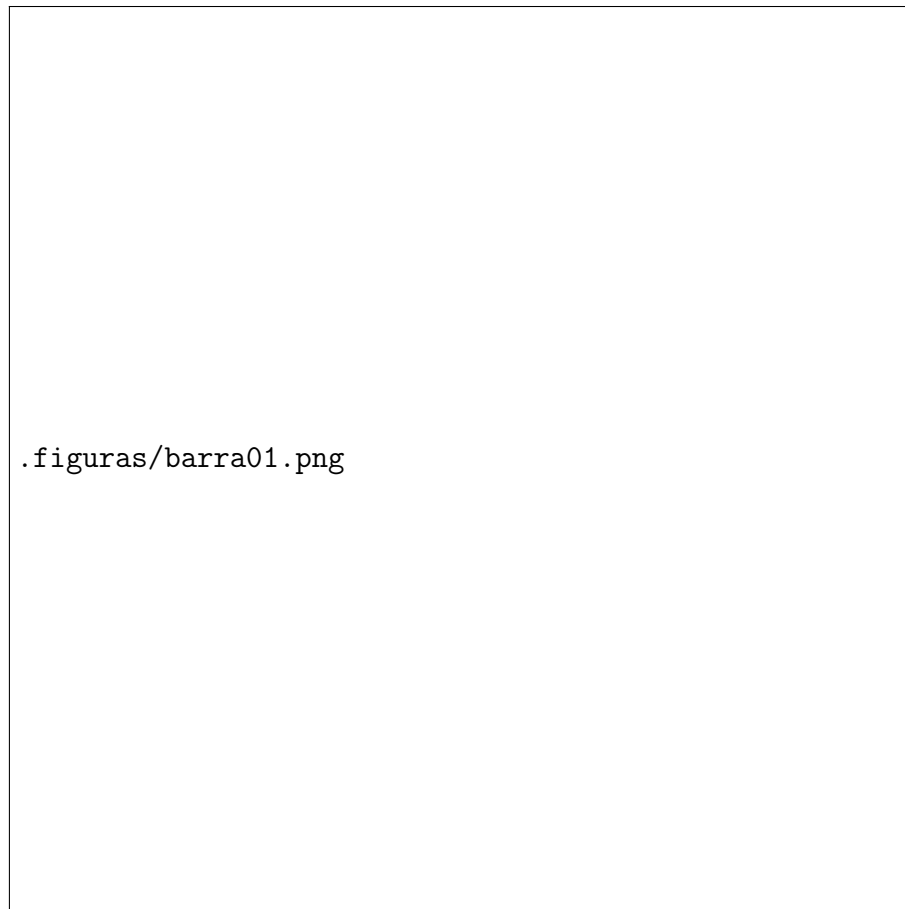
$$\begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} & k_{15} & k_{16} \\ k_{21} & k_{22} & k_{23} & k_{24} & k_{25} & k_{26} \\ k_{31} & k_{32} & k_{33} & k_{34} & k_{35} & k_{36} \\ k_{41} & k_{42} & k_{43} & k_{44} & k_{45} & k_{46} \\ k_{51} & k_{52} & k_{53} & k_{54} & k_{55} & k_{56} \\ k_{61} & k_{62} & k_{63} & k_{64} & k_{65} & k_{66} \end{bmatrix} \begin{Bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \end{Bmatrix} = \begin{Bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \end{Bmatrix} \quad (3.2)$$

Ademais, com a finalidade de deixar essa explicação visualmente palpável, pode-se plotar tal estrutura em uma ilustração conforme foi feito na Figura ??

Dessa maneira, a matriz de rigidez expressa em coordenadas locais apresenta a configuração clássica descrita por Martha (2010). Esta formulação incorpora simultaneamente os efeitos das solicitações normais e dos momentos fletores. Desconsiderando-se as deformações produzidas pelas forças cortantes (por serem desprezíveis diante daquelas produzidas pela flexão, principalmente em elementos cuja relação entre o comprimento do elemento e a altura da seção transversal é maior que 5), tem-se a expressão da matriz de rigidez para elemento de seção transversal constante, mostrada na Equação 3.3.

$$\mathbf{K} = \begin{bmatrix} \frac{EA}{L} & 0 & 0 & -\frac{EA}{L} & 0 & 0 \\ 0 & \frac{12EI}{L^3} & \frac{6EI}{L^2} & 0 & -\frac{12EI}{L^3} & \frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{4EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{2EI}{L} \\ -\frac{EA}{L} & 0 & 0 & \frac{EA}{L} & 0 & 0 \\ 0 & -\frac{12EI}{L^3} & -\frac{6EI}{L^2} & 0 & \frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{2EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{4EI}{L} \end{bmatrix} \quad (3.3)$$

Figura 2 – Graus de liberdade nodais



Fonte: Elaboração do autor, 2026

Nota: Os graus de liberdade nodais são: d_1 e d_4 (deslocamentos horizontais); d_2 e d_5 (deslocamentos verticais); d_3 e d_6 (rotações) dos Nós 1 e 2, respectivamente.

Em que: E = módulo de elasticidade do material; A = área da seção transversal; I = momento de inércia da seção transversal; L = comprimento do elemento.

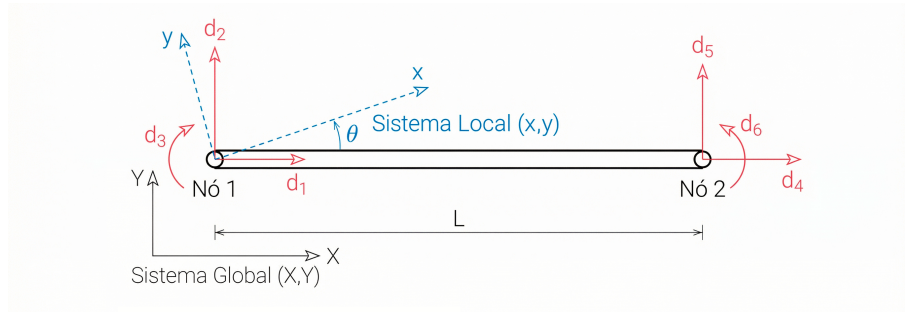
Esta formulação matricial caracteriza o comportamento mecânico de um elemento estrutural quando submetido a cargas concentradas ou distribuídas e momentos fletores, constituindo ferramenta essencial na análise computacional de estruturas aperticadas planas. A construção da matriz de rigidez do sistema estrutural global procede mediante a combinação das matrizes individuais de cada elemento, após a apropriada transformação para o sistema de coordenadas global (MARTHA, 2010)

3.2.3 Matriz de Transformação (T) (Rotação de Eixos)

Conforme ilustrado na Figura 3 e descrito por Colunga (2007), no caso da rotação de eixos no plano cartesiano bidimensional, o sistema de coordenadas global é estabelecido no plano XY .

Ao aplicar-se uma rotação de ângulo θ em relação à origem, obtém-se o sistema local xy do elemento, alinhado com seu eixo longitudinal.

Figura 3 – Rotação do elemento de barra.



Fonte: Elaboração do autor, 2026

Nota: Os graus de liberdade nodais são: d_1 e d_4 (deslocamentos horizontais); d_2 e d_5 (deslocamentos verticais); d_3 e d_6 (rotações) dos Nós 1 e 2, respectivamente.

3.2.4 Matriz de Rigidez Global

A conversão de coordenadas entre os sistemas local e global – e também no sentido inverso – é realizada mediante uma matriz de transformação $[T]$, que estabelece a correspondência matemática entre ambos os referenciais. A construção dessa matriz requer uma etapa preliminar, na qual se define a matriz de rotação $[R]$.

Esta matriz descreve como as componentes de um vetor, originalmente expressas no sistema local (xy) , podem ser representadas no sistema global (XY) . Sua formulação, apresentada na Equação 3.4, fundamenta-se nas funções trigonométricas seno e cosseno do ângulo de rotação entre os dois sistemas de eixos:

$$[T] = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.4)$$

Desse modo, pode-se relacioná-la com a matriz de rigidez do elemento em coordenadas locais, de modo que cada elemento k'_{ij} seja multiplicado por $[T]$ e pela sua transposta, tal qual é exemplificado na Figura 3, de modo a formar a seguinte matriz transformada:

$$[K] = [T]^T [K'] [T] \quad (3.5)$$

Após a obtenção das matrizes de rigidez de cada elemento em seus respectivos sistemas de coordenadas locais, realiza-se a transformação dessas matrizes para o sistema global por meio das matrizes de rotação [T], considerando a geometria e a orientação espacial de cada elemento estrutural. Esse procedimento garante que as propriedades de rigidez sejam corretamente alinhadas ao referencial adotado para toda a estrutura, possibilitando a montagem da matriz de rigidez global por meio da contribuição de cada elemento. (GERE; WEAVER, 1987).

A próxima etapa consiste na montagem da matriz de rigidez global da estrutura, a partir da superposição das matrizes de rigidez de cada elemento com os graus de liberdade alinhados nas direções dos eixos globais, conforme a conectividade nodal e os graus de liberdade associados a cada nó (PRZEMIENIECKI, 1968). Esse procedimento conduz à montagem da matriz de rigidez não restringida, que considera todos os graus de liberdade presentes no modelo estrutural, inclusive aqueles que eventualmente terão os deslocamentos prescritos (nulos ou definidos como diferentes de zero).

A dimensão da matriz global é definida pelo produto do número total de nós da estrutura pelo número de deslocamentos possíveis por nó, o qual é igual a 3 em pórticos planos, de modo que o grau n da matriz é igual ao total de graus de liberdade ativos no modelo (ZIENKIEWICZ, 2000). A montagem da matriz de rigidez global pode ser realizada por dois procedimentos principais: o método “nó a nó” ou o método “peça a peça”. O segundo é mais eficiente do ponto de vista computacional, pois processa cada elemento apenas uma vez (DÍAZ *et al.*, 2015).

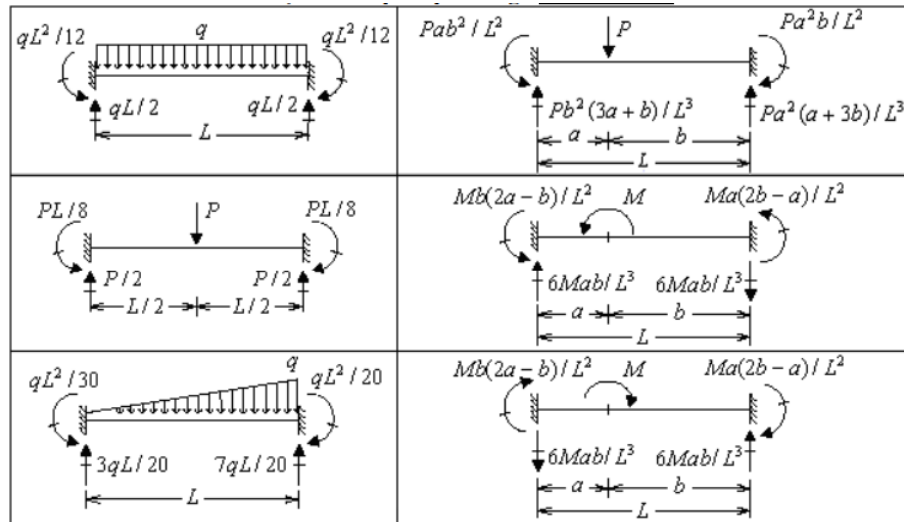
3.2.5 Vetor de cargas nodais combinadas

Para a implementação computacional dessa formulação, o desenvolvimento das matrizes de rigidez e vetores de carregamento requer uma abordagem sistemática que considere tanto o sistema de coordenadas locais de cada elemento quanto o sistema global da estrutura. Nesse contexto, o vetor de cargas global é elaborado no mesmo sistema de coordenadas da matriz de rigidez global, incorporando tanto as cargas nodais aplicadas diretamente nas direções dos graus de liberdade quanto os efeitos de cargas distribuídas ou concentradas sobre os elementos, previamente convertidas em cargas nodais equivalentes (ZIENKIEWICZ, 2000).

A determinação dos carregamentos nodais combinados segue diretamente dessa modelagem, pois representa a forma discretizada das ações externas que atuam sobre os graus de liberdade do modelo (MARTHA, 2010; SORIANO; LIMA, 2005). Essas ações dividem-se em duas parcelas: (i) carregamentos aplicados diretamente nos nós (forças e momentos coincidentes com

coordenadas nodais); e (ii) carregamentos equivalentes decorrentes de ações distribuídas ou concentradas sobre os elementos, que precisam ser transformados em forças nodais equivalentes por integração com funções de forma ou por uso de expressões tabuladas e disposta na Figura 4.

Figura 4 – Expressões tabuladas para método da rigidez



Fonte: (Universidade Federal do Paraná, 2015)

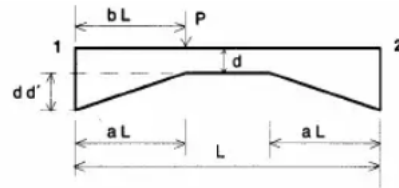
Tais forças equivalentes, determinadas inicialmente no sistema local, são posteriormente transformadas para o sistema global por meio de matrizes de rotação (MARTHA, 2010), garantindo que todas as contribuições sejam corretamente somadas, como ilustrado na Figura 4.

3.3 Modelagem de Elementos de Barra de Seção Variável

A existência de elementos estruturais de seção transversal variável é um fato relativamente comum na prática da engenharia estrutural, embora nem sempre esteja presente nos programas de análise de estruturas. A correta determinação dos esforços, deformações e deslocamentos nesses elementos é crucial para o dimensionamento seguro e eficiente desses membros estruturais (COLUNGA, 2007). No passado, a análise de elementos de seção variável dependia de auxílios de projeto, como as tabelas da Portland Cement Association (PCA), conforme ilustrado na Figura 5. No entanto, esses métodos incorporavam simplificações significativas, o que pode levar a erros expressivos (COLUNGA, 2007).

A evolução das formulações levou a abordagens mais rigorosas. Uma técnica poderosa e precisa para determinar a matriz de rigidez de um elemento de barra com seção variável é o Método da Flexibilidade. Nesse processo, calculam-se primeiramente os coeficientes da matriz

Figura 5 – Exemplo de tabelas da Portland Cement Association (PCA) para barras de seção variável.



Haunch	Cary-over Factors	Stiffness Factors	Unif. Load FEM Coef. x w L ²	Concentrated Load. FEM. Coef. x P L											
				b											
				0.1		0.3		0.5		0.7		0.9			
a	d'	r12 = r21	C12 = C21	M12 = M21	M12	M21	M12	M21	M12	M21	M12	M21	M12	M21	
0.1	0.6	0.567	5.12	0.0905	0.091	0.005	0.168	0.061	0.137	0.137	0.061	0.168	0.005	0.091	
	1.0	0.588	5.54	0.0925	0.094	0.004	0.175	0.059	0.140	0.140	0.059	0.175	0.004	0.094	
	1.5	0.603	5.89	0.0941	0.096	0.003	0.180	0.058	0.143	0.143	0.058	0.180	0.003	0.096	
0.3	0.6	0.647	8.04	0.0982	0.090	0.007	0.188	0.062	0.153	0.153	0.063	0.188	0.007	0.090	
	1.0	0.705	10.85	0.1034	0.092	0.005	0.206	0.058	0.164	0.164	0.058	0.206	0.005	0.092	
	1.5	0.753	14.27	0.1074	0.094	0.004	0.224	0.051	0.173	0.173	0.051	0.224	0.004	0.094	
0.5	0.6	0.633	10.72	0.0969	0.087	0.008	0.176	0.068	0.154	0.154	0.068	0.176	0.008	0.087	
	1.0	0.692	17.34	0.1023	0.089	0.008	0.189	0.069	0.167	0.167	0.069	0.189	0.008	0.089	
	1.5	0.748	28.32	0.1070	0.090	0.007	0.200	0.068	0.179	0.179	0.068	0.200	0.007	0.090	

Fonte: Adaptado de Portland Cement Association apud Portland Cement Association (1947).

de flexibilidade do elemento, obtidos por integração, ao longo do comprimento da barra, de propriedades como área da seção transversal $A(x)$ e momento de inércia $I(x)$. A matriz de rigidez é então obtida pela inversão da matriz de flexibilidade (MONROY-MIRANDA, 2003; COLUNGA, 2007).

A etapa de integração pode ser algebricamente complexa ou tediosa quando a geometria da seção apresenta variações. Nesses casos, a integração numérica surge como uma ferramenta extremamente eficaz. Métodos como a regra de Simpson, a quadratura de Gauss-Legendre, ou outras rotinas numéricas, podem ser implementados em programas de computador para avaliar as integrais de forma precisa e obter os coeficientes de rigidez e esforços de engastamento perfeito (utilizados na obtenção das cargas nodais equivalentes) (MONROY-MIRANDA, 2003).

Essa abordagem, que considera as deformações axiais e/ou por cortante, bem como a forma da seção transversal, demonstra ser robusta e confiável, produzindo resultados muito próximos aos obtidos com formulações mais complexas de elementos finitos, sendo recomendada para análise de pórticos com elementos de seção variável (COLUNGA, 2007).

3.4 Cálculo dos deslocamentos nodais e esforços de extremidade de elementos

Uma vez estabelecidas a matriz de rigidez global não restringida K e o vetor de carregamentos combinados $\{F\}$, procede-se à aplicação das condições de contorno para viabilizar a resolução do sistema de equações e obter os deslocamentos nodais (incógnitas primárias do problema) desejados da análise estrutural.

$$[K]\{\delta\} = \{F\} \quad (3.6)$$

Considerando deslocamentos prescritos (nulos ou diferentes de zero) nos graus de liberdade restringidos pelos vínculos estruturais, torna-se necessário aplicar as condições de contorno para resolver o sistema linear resultante da formulação do método da rigidez direta.

Para incorporar essas restrições, uma abordagem amplamente utilizada consiste no método do número grande, no qual se adiciona um valor numérico de magnitude elevada aos termos diagonais da matriz de rigidez correspondentes aos graus de liberdade restringidos. Além dessa alteração na matriz de rigidez global da estrutura, altera-se o vetor de ações externas $\{F\}$, substituindo-se o valor da carga na direção do grau de liberdade restringido pelo mesmo número grande multiplicado pelo valor do deslocamento prescrito naquela direção. Esse procedimento, conhecido como técnica do número grande, impõe de forma numericamente eficiente os deslocamentos prescritos, mantendo o sistema bem condicionado e determinado (SORIANO; LIMA, 2005).

Dessa forma, a matriz de rigidez modificada permite a resolução direta do sistema de equações, viabilizando a determinação simultânea dos deslocamentos nodais incógnitos e das reações de apoio nos vínculos estruturais, completando assim a primeira etapa da análise: a obtenção das incógnitas primárias (os deslocamentos nodais) da estrutura pelo método da rigidez direta.

3.5 Implementação Computacional com Python

A seleção do Python como linguagem de programação para a implementação computacional do método da rigidez fundamenta-se em suas características técnicas que favorecem o desenvolvimento de aplicações em engenharia estrutural e métodos numéricos. Python tem se consolidado como uma ferramenta de primeira escolha para computação científica devido à sua sintaxe intuitiva, extensibilidade e rico ecossistema de bibliotecas especializadas (OLIPHANT, 2007). Ademais, a linguagem oferece bibliotecas científicas robustas que facilitam significativamente a implementação de algoritmos matriciais complexos. O NumPy fornece estruturas de dados

otimizadas para operações com *arrays* multidimensionais, enquanto o SciPy disponibiliza rotinas especializadas para álgebra linear, incluindo solucionadores eficientes para sistemas lineares esparsos, fundamentais na análise estrutural pelo método da rigidez (VIRTANEN *et al.*, 2020). Essas bibliotecas implementam algoritmos otimizados em linguagens de baixo nível (C/Fortran), proporcionando desempenho comparável a implementações tradicionais em engenharia.

Dessa forma, a aplicabilidade do Python em análise estrutural tem sido demonstrada em diversos contextos acadêmicos e profissionais. A facilidade de prototipagem rápida permite que engenheiros implementem e testem diferentes formulações do método da rigidez de forma ágil, concentrando-se na lógica estrutural em vez dos aspectos de implementação de baixo nível (MCKINNEY, 2017). Adicionalmente, as capacidades nativas de visualização, por meio do Matplotlib, possibilitam a criação de interfaces gráficas integradas para pré e pós-processamento, essenciais em análises estruturais modernas.

4 METODOLOGIA

A metodologia a ser adotada neste trabalho fundamenta-se no desenvolvimento de uma aplicação computacional especializada para análise estrutural de pórticos planos, empregando os princípios do método da rigidez direta como base teórica para os cálculos estruturais.

Dessa maneira, a abordagem integrará aspectos de engenharia de software, de análise numérica e de interface humano-computador, visando criar uma ferramenta que atenda tanto às necessidades acadêmicas quanto às demandas profissionais em engenharia estrutural. O processo de desenvolvimento seguirá uma estrutura modular, onde cada componente do sistema será concebido de forma independente, permitindo maior flexibilidade na implementação e facilidade de manutenção do código.

Além disso, a implementação do software será realizada utilizando a linguagem Python, escolhida por suas características que favorecem o desenvolvimento de aplicações científicas e de engenharia. Para os cálculos matriciais e operações de álgebra linear — essenciais na implementação do método da rigidez — serão utilizadas as bibliotecas NumPy e SciPy. Estas fornecerão estruturas de dados otimizadas para operações com matrizes multidimensionais, enquanto aquelas disponibilizarão algoritmos especializados para resolução de sistemas lineares e de outras operações matemáticas avançadas. Tais bibliotecas garantirão eficiência computacional comparável a implementações em linguagens de baixo nível, mantendo a facilidade de desenvolvimento característica do Python.

Assim, o desenvolvimento da interface gráfica utilizará a biblioteca Tkinter, nativa do Python, permitindo a criação de aplicações desktop multiplataforma sem dependências externas. A interface será projetada seguindo princípios de usabilidade e de experiência do usuário, organizando as funcionalidades de forma intuitiva e logicamente estruturada. Dessa maneira, a modelagem da estrutura será realizada através de um ambiente gráfico bidimensional que permitirá ao usuário definir geometria, propriedades dos materiais, carregamentos e condições de contorno de forma visual e interativa.

Seguindo esse planejamento, o núcleo computacional do software implementará o MRD seguindo rigorosamente os fundamentos teóricos da análise matricial de estruturas. O algoritmo será estruturado em etapas sequenciais, incluindo: montagem das matrizes de rigidez locais dos elementos, transformação para o sistema global de coordenadas, formação da Matriz de Rigidez Global, aplicação das condições de contorno e resolução do sistema linear resultante. Uma funcionalidade especial a ser implementada consiste no tratamento de elementos com

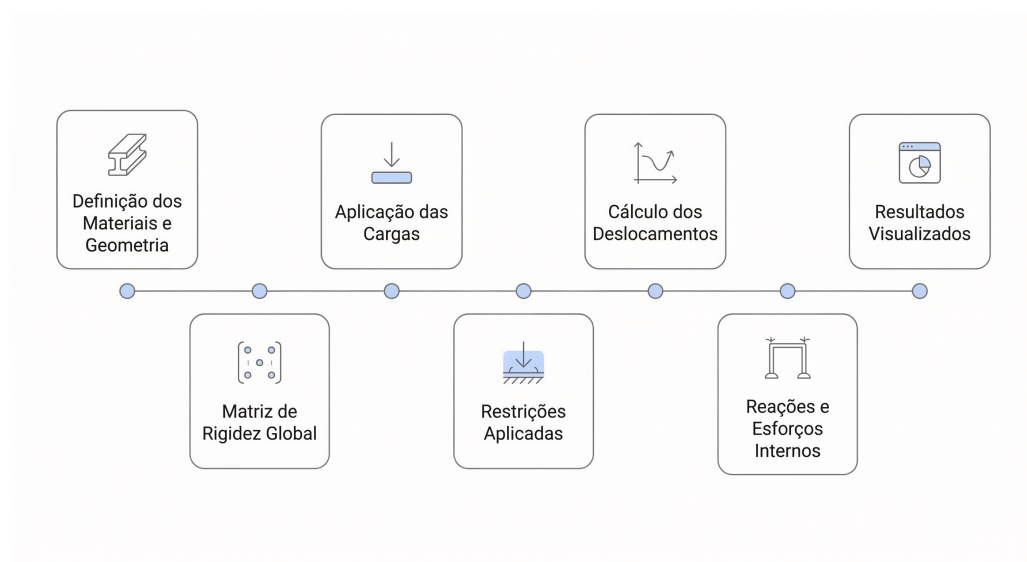
seção transversal de inércia variável, utilizando técnicas de integração numérica para calcular os coeficientes de rigidez correspondentes. Esta funcionalidade ampliará significativamente a aplicabilidade do programa, permitindo a análise de estruturas com geometrias mais complexas e realistas.

Ademais, esta concepção incluirá um processo sistemático de validação dos resultados computacionais através da comparação com soluções analíticas conhecidas e com resultados obtidos por outros textitsoftware consolidados no mercado.

4.1 Funcionamento da aplicação

O programa alvo desta metodologia apresenta uma rotina clara, de fácil interpretação e com possibilidade de ser derivada e ampliada por outros profissionais. Para facilitar a compreensão da rotina implementada neste textitsoftwares, foi desenvolvido o fluxograma mostrado na Figura 6, o qual ilustra detalhadamente a sequência lógica das operações do Método da Rigidez Direta (MRD), desde a entrada dos dados pelo usuário até a visualização dos resultados.

Figura 6 – Fluxograma geral do funcionamento da aplicação baseada no Método da Rigidez Direta (MRD).



Fonte: Elaboração do autor, 2026

5 CRONOGRAMA

Para a continuidade e conclusão deste trabalho de pesquisa, foi elaborado um cronograma detalhado que contempla todas as etapas necessárias para o desenvolvimento adequado do projeto. O planejamento temporal apresentado na Tabela 1 estabelece uma sequência lógica de atividades, considerando a interdependência entre as diferentes fases do trabalho e os prazos institucionais estabelecidos.

O cronograma foi estruturado de forma a otimizar o tempo disponível, permitindo dedicação adequada a cada etapa da pesquisa. As atividades foram organizadas considerando a complexidade de cada fase, os recursos necessários e os possíveis imprevistos que podem surgir durante o desenvolvimento do trabalho.

A divisão temporal proposta busca equilibrar as demandas teóricas e práticas da pesquisa, garantindo tempo suficiente para a coleta e análise de dados, bem como para a redação e revisão do texto final. Além disso, foram considerados períodos para orientação acadêmica e possíveis ajustes metodológicos que se façam necessários ao longo do processo.

Tabela 1 – Cronograma de atividades do projeto

Mês/Ano	Atividade
Outubro/2025	Implementação do Backend do código
Novembro/2025	Validações com outros autores e programas comerciais
Dezembro/2025	Implementação do Frontend
Janeiro/2026	Finalização e documentação do trabalho

Fonte: Elaborado pelo autor

É importante ressaltar que esse cronograma deve ser entendido como uma ferramenta de planejamento flexível, passível de ajustes conforme as necessidades específicas que possam emergir durante o desenvolvimento da pesquisa. A experiência acadêmica demonstra que trabalhos de natureza científica frequentemente demandam adaptações temporais em função de descobertas inesperadas, dificuldades metodológicas ou oportunidades de aprofundamento em aspectos particulares do tema estudado.

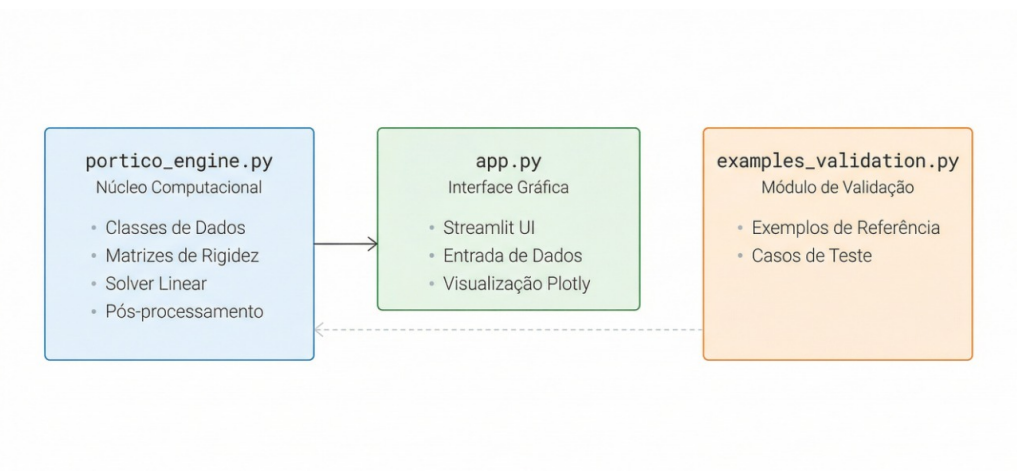
O acompanhamento sistemático do cumprimento das etapas propostas será realizado através de reuniões de orientação, permitindo avaliação do progresso e identificação precoce de eventuais desvios no cronograma. Desta forma, garante-se não apenas o cumprimento dos prazos estabelecidos, mas também a qualidade acadêmica esperada para um trabalho desta natureza.

6 DESCRIÇÃO DO SOFTWARE

6.1 Arquitetura do Sistema

O sistema foi desenvolvido seguindo o padrão de arquitetura em camadas, separando claramente a lógica de processamento (*backend*) da interface de usuário (*frontend*). De modo que essa separação permita maior testabilidade, manutenibilidade e possibilidade de extensão futura do código. Ademais, a Figura 7 ilustra a organização modular da aplicação.

Figura 7 – Diagrama de arquitetura do *software* PorticoAnalyzer.



Fonte: Elaborado pelo autor (2026).

A aplicação é composta por três módulos principais:

- portico_engine.py:** Núcleo computacional contendo todas as funções relacionadas ao cálculo estrutural, incluindo montagem das matrizes de rigidez locais dos elementos, transformação para o sistema global de coordenadas, formação da matriz de rigidez global, aplicação das condições de contorno e resolução do sistema linear resultante. Esse módulo foi projetado para ser completamente independente da interface gráfica, podendo ser utilizado como biblioteca em outras aplicações ou *scripts*;
- app.py:** Interface gráfica web desenvolvida com a biblioteca *Streamlit*, proporcionando uma experiência de usuário moderna e intuitiva. O *design* adota o estilo *glassmorphism*, caracterizado por transparências, efeitos de desfoque e gradientes sutis;
- examples_validation.py:** Módulo de exemplos de validação contendo casos de referência extraídos da literatura técnica, permitindo a verificação sistemática da coerência dos resultados obtidos.

6.2 Estrutura de Dados

O módulo `portico_engine.py` define três classes fundamentais utilizando o decorador `@dataclass` do Python, que simplifica a criação de classes de dados com validação automática. Essas classes constituem a base para a representação computacional dos elementos estruturais.

6.2.1 Classe Material

A classe `Material` armazena as propriedades mecânicas do material estrutural, conforme apresentado na Figura 8. Os atributos incluem o módulo de elasticidade longitudinal (E) e o coeficiente de dilatação térmica (α), sendo este essencial para análises envolvendo variações de temperatura.

Figura 8 – Definição da classe `Material` em Python.

```
1 @dataclass
2 class Material:
3     id: int
4     E: float          # Modulo de elasticidade (Pa)
5     alpha: float     # Coeficiente de dilatacao termica (1/K)
6     name: str = ""
```

Fonte: Elaborado pelo autor (2025).

A validação automática implementada no método `__post_init__` impede a criação de materiais com módulo de elasticidade negativo ou nulo, garantindo a consistência física dos dados de entrada.

6.2.2 Classe Node

A classe `Node` representa um ponto nodal da estrutura com suas coordenadas no sistema global de referência, conforme apresentado na Figura 9. Cada nó possui um identificador único (`id`) que é utilizado para referenciar as conectividades dos elementos e as condições de contorno.

Figura 9 – Definição da classe Node em Python.

```

1 @dataclass
2 class Node:
3     id: int
4     x: float    # Coordenada X (m)
5     y: float    # Coordenada Y (m)

```

Fonte: Elaborado pelo autor (2025).

6.2.3 Classe Element

A classe Element constitui a estrutura de dados mais complexa da aplicação, representando um elemento de barra com possibilidade de seção transversal variável. A Figura 10 apresenta sua definição.

Figura 10 – Definição da classe Element em Python.

```

1 @dataclass
2 class Element:
3     id: int
4     start: Node    # No inicial
5     end: Node      # No final
6     material: Material
7     b_start: float # Base no inicio (m)
8     h_start: float # Altura no inicio (m)
9     b_end: float   # Base no final (m)
10    h_end: float   # Altura no final (m)

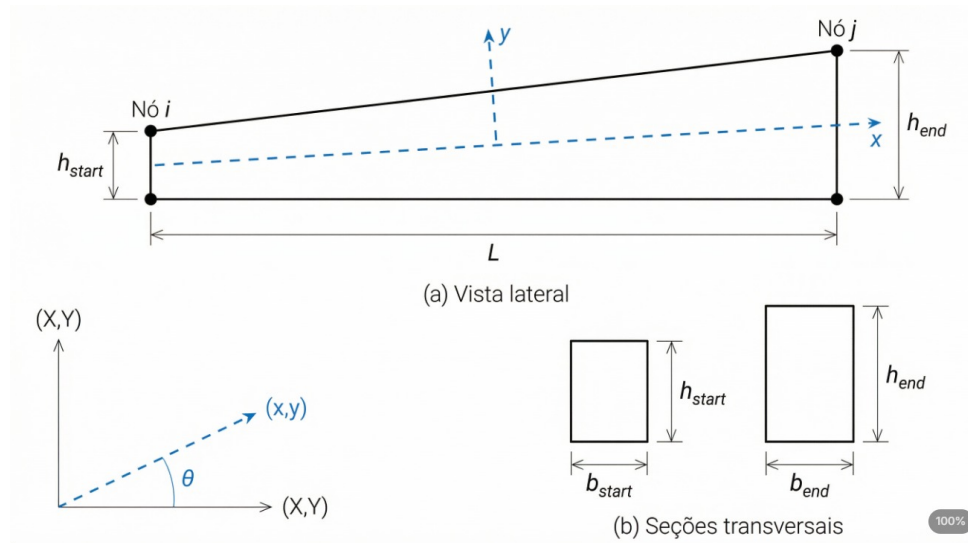
```

Fonte: Elaborado pelo autor (2025).

As propriedades geométricas derivadas, tais como comprimento L e ângulos $c = \cos \theta$ e $s = \sin \theta$, são calculadas automaticamente no método `__post_init__`. O método `is_variable()` verifica se o elemento possui seção variável comparando as dimensões inicial e final, retornando True quando $h_{start} \neq h_{end}$ ou $b_{start} \neq b_{end}$.

A Figura 11 ilustra a representação geométrica de um elemento de barra com seção transversal variável, destacando os sistemas de coordenadas local e global.

Figura 11 – Representação esquemática de um elemento de barra com seção variável.



Fonte: Elaborado pelo autor (2026).

6.3 Algoritmo de Cálculo da Matriz de Rigidez

O cálculo da matriz de rigidez local constitui o núcleo do Método da Rigidez Direta. A implementação desenvolvida contempla duas formulações distintas: uma analítica para elementos de seção constante e outra numérica para elementos de seção variável.

6.3.1 Elementos de Seção Constante

Para elementos com seção transversal constante, a matriz de rigidez local 6×6 é calculada pela função `local_stiffness_constant()`, utilizando a formulação clássica da teoria de Euler-Bernoulli. Desprezando-se as deformações por cisalhamento, válido para elementos com relação $L/h > 5$, a matriz de rigidez local é expressa pela Equação 6.1.

$$[\mathbf{K}_{\text{local}}] = \begin{bmatrix} \frac{EA}{L} & 0 & 0 & -\frac{EA}{L} & 0 & 0 \\ 0 & \frac{12EI}{L^3} & \frac{6EI}{L^2} & 0 & -\frac{12EI}{L^3} & \frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{4EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{2EI}{L} \\ -\frac{EA}{L} & 0 & 0 & \frac{EA}{L} & 0 & 0 \\ 0 & -\frac{12EI}{L^3} & -\frac{6EI}{L^2} & 0 & \frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{2EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{4EI}{L} \end{bmatrix} \quad (6.1)$$

onde E representa o módulo de elasticidade do material, A a área da seção transversal, I o momento de inércia da seção e L o comprimento do elemento.

6.3.2 Elementos de Seção Variável

Para elementos com seção transversal variável, implementou-se o Método das Flexibilidades com integração numérica, conforme proposto por (COLUNGA, 2007). Essa abordagem consiste em calcular, primeiramente, os coeficientes da matriz de flexibilidade do elemento por integração ao longo do comprimento da barra e, posteriormente, obter a matriz de rigidez por inversão.

A função `local_stiffness_variable()` calcula os coeficientes de flexibilidade através de integrais numéricas utilizando a quadratura adaptativa da biblioteca SciPy. Os coeficientes de flexibilidade são definidos pelas Equações 6.2 a 6.5.

$$f_{11} = \int_0^L \frac{1}{E \cdot A(z)} dz \quad (6.2)$$

$$f_{22} = \int_0^L \frac{z^2}{E \cdot I(z)} dz \quad (6.3)$$

$$f_{26} = \int_0^L \frac{z}{E \cdot I(z)} dz \quad (6.4)$$

$$f_{66} = \int_0^L \frac{1}{E \cdot I(z)} dz \quad (6.5)$$

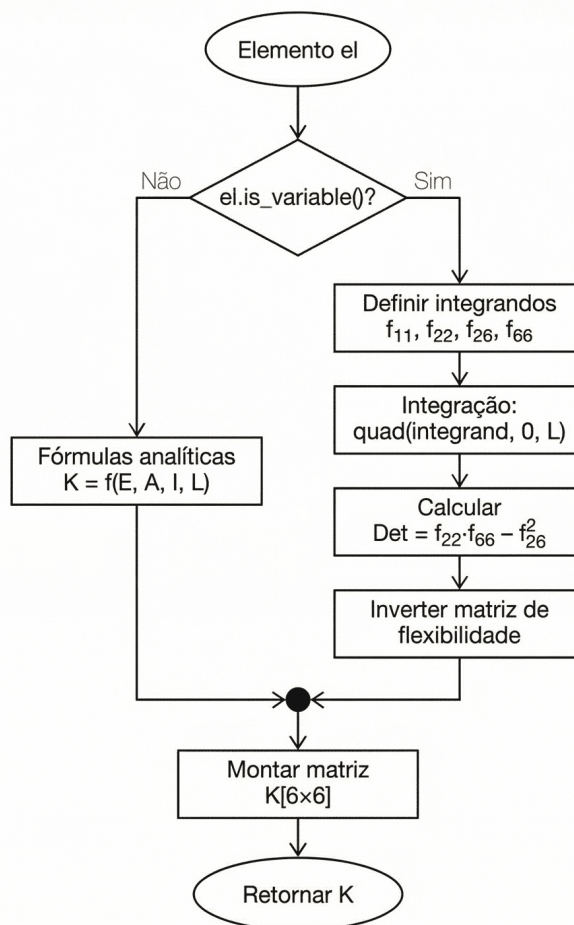
onde $A(z)$ e $I(z)$ representam as funções de variação da área e do momento de inércia ao longo do comprimento do elemento, respectivamente. Para variação linear das dimensões da seção transversal, tem-se:

$$h(z) = h_{\text{start}} + \frac{h_{\text{end}} - h_{\text{start}}}{L} \cdot z \quad (6.6)$$

O método utiliza a função `scipy.integrate.quad` para avaliar numericamente as integrais, empregando quadratura adaptativa de Gauss-Kronrod que ajusta automaticamente o número de pontos de integração para garantir a precisão especificada.

A Figura 12 apresenta o fluxograma do algoritmo de cálculo da matriz de rigidez, evidenciando a bifurcação entre os procedimentos para seção constante e variável.

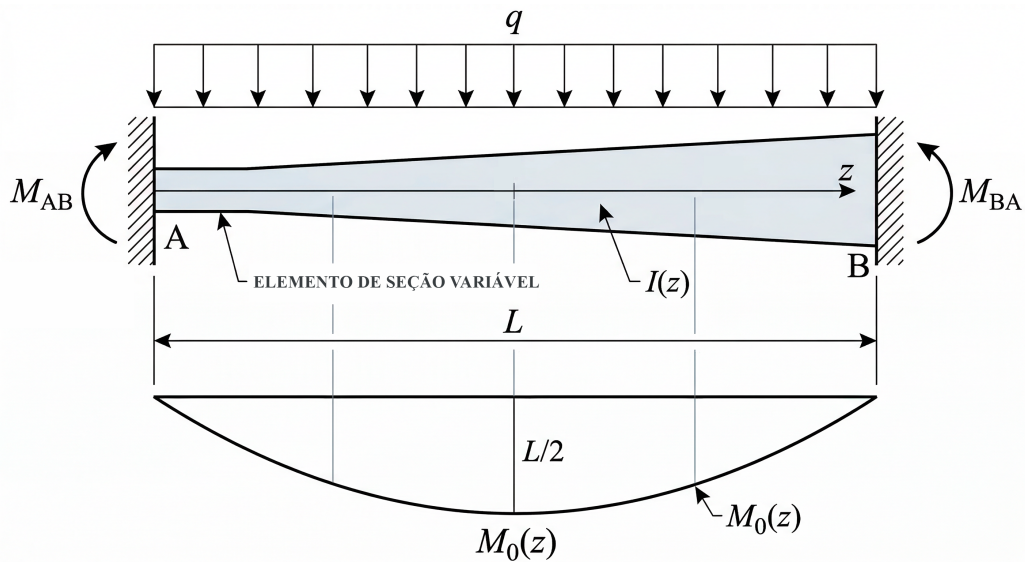
Figura 12 – Fluxograma do algoritmo de cálculo da matriz de rigidez local.



6.3.3 Momentos de Engastamento Perfeito para Seção Variável

Uma contribuição significativa deste trabalho foi a implementação do cálculo dos momentos de engastamento perfeito para elementos de seção variável, utilizando os fatores de forma K1 a K5 conforme metodologia proposta por (COLUNGA, 2007). Esta formulação permite calcular corretamente as cargas nodais equivalentes para cargas distribuídas atuando em elementos de seção variável, conforme ilustrado na Figura 13. Alternativamente, esses esforços de engastamento perfeito também podem ser obtidos por meio do método da flexibilidade, no qual se determinam os esforços de extremidade de elementos de barra com seção transversal variável submetidos a carregamentos distribuídos.

Figura 13 – Elemento de viga com seção variável submetido a carregamento distribuído.



Fonte: Elaboração do autor, 2026

Os fatores de forma de primeira espécie são definidos pelas Equações 6.7 a 6.9.

$$K_1 = \int_0^L \frac{z^2}{I(z)} dz \quad (6.7)$$

$$K_2 = \int_0^L \frac{(L-z)^2}{I(z)} dz \quad (6.8)$$

$$K_3 = \int_0^L \frac{z(L-z)}{I(z)} dz \quad (6.9)$$

Os fatores de forma de segunda espécie, que incorporam o momento isostático $M_0(z)$ devido ao carregamento aplicado, são expressos pelas Equações 6.10 e 6.11.

$$K_4 = \int_0^L \frac{M_0(z) \cdot z}{I(z)} dz \quad (6.10)$$

$$K_5 = \int_0^L \frac{M_0(z) \cdot (L - z)}{I(z)} dz \quad (6.11)$$

Para uma carga uniformemente distribuída q , o momento isostático é dado por:

$$M_0(z) = \frac{qL}{2} \cdot z - \frac{q}{2} \cdot z^2 \quad (6.12)$$

Os momentos de engastamento perfeito nas extremidades A e B do elemento são então calculados pelas Equações 6.13 e 6.14.

$$M_{AB} = -\frac{K_3 \cdot K_4 - K_1 \cdot K_5}{K_1 \cdot K_2 - K_3^2} \quad (6.13)$$

$$M_{BA} = -\frac{K_2 \cdot K_4 - K_3 \cdot K_5}{K_1 \cdot K_2 - K_3^2} \quad (6.14)$$

6.4 Montagem e Resolução do Sistema Global

6.4.1 Matriz de Transformação

A transformação das matrizes de rigidez do sistema local para o sistema global é realizada pela matriz de rotação $[\mathbf{T}]$, implementada na função `rotation_matrix()`. Para um elemento com ângulo de inclinação θ em relação ao eixo horizontal global, a matriz de transformação 6×6 é expressa pela Equação 6.15.

$$[\mathbf{T}] = \begin{bmatrix} c & s & 0 & 0 & 0 & 0 \\ -s & c & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & c & s & 0 \\ 0 & 0 & 0 & -s & c & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (6.15)$$

onde $c = \cos \theta$ e $s = \sin \theta$ são os cossenos diretores do elemento. A matriz de rigidez em coordenadas globais é obtida pela operação:

$$[\mathbf{K}_{\text{global}}^{(e)}] = [\mathbf{T}]^T [\mathbf{K}_{\text{local}}] [\mathbf{T}] \quad (6.16)$$

6.4.2 Montagem da Matriz de Rigidez Global

A função `assemble_global_stiffness()` monta a matriz de rigidez global da estrutura através do procedimento “peça a peça”, somando as contribuições de cada elemento nos graus de liberdade correspondentes. Este procedimento é mais eficiente computacionalmente que o método “nó a nó”, pois processa cada elemento apenas uma vez (COLUNGA, 2007).

A dimensão da matriz global é definida pelo produto do número total de nós da estrutura pelo número de graus de liberdade por nó, que é igual a 3 em pórticos planos (duas translações e uma rotação). Assim, para uma estrutura com n nós, a matriz de rigidez global possui dimensão $3n \times 3n$.

6.4.3 Aplicação das Condições de Contorno

As condições de contorno são aplicadas pelo método da grande rigidez (método da penalidade), adicionando um valor numérico elevado (10^{20}) aos termos diagonais correspondentes aos graus de liberdade restringidos. Este método preserva a dimensão da matriz global e evita a necessidade de rearranjo de linhas e colunas (SORIANO; LIMA, 2005).

Para um deslocamento prescrito \bar{d}_i no grau de liberdade i , as modificações são:

$$K_{ii} \leftarrow K_{ii} + \beta \quad (6.17)$$

$$F_i \leftarrow \beta \cdot \bar{d}_i \quad (6.18)$$

onde $\beta = 10^{20}$ é o número grande de penalidade. Para apoios fixos, $\bar{d}_i = 0$.

6.4.4 Resolução do Sistema Linear

O sistema de equações $[\mathbf{K}]\{\mathbf{D}\} = \{\mathbf{F}\}$ é resolvido utilizando a função `numpy.linalg.solve()`, que implementa a decomposição LU com pivoteamento parcial. Esta abordagem é numeri-

camente estável e eficiente para sistemas de equações lineares de porte moderado, típicos de estruturas de pórticos.

6.5 Interface Gráfica do Usuário

A interface gráfica foi desenvolvida utilizando a biblioteca Streamlit, que permite a criação de aplicações web interativas diretamente em Python. O *design* adota elementos visuais modernos, com gradientes sutis e organização clara das funcionalidades.

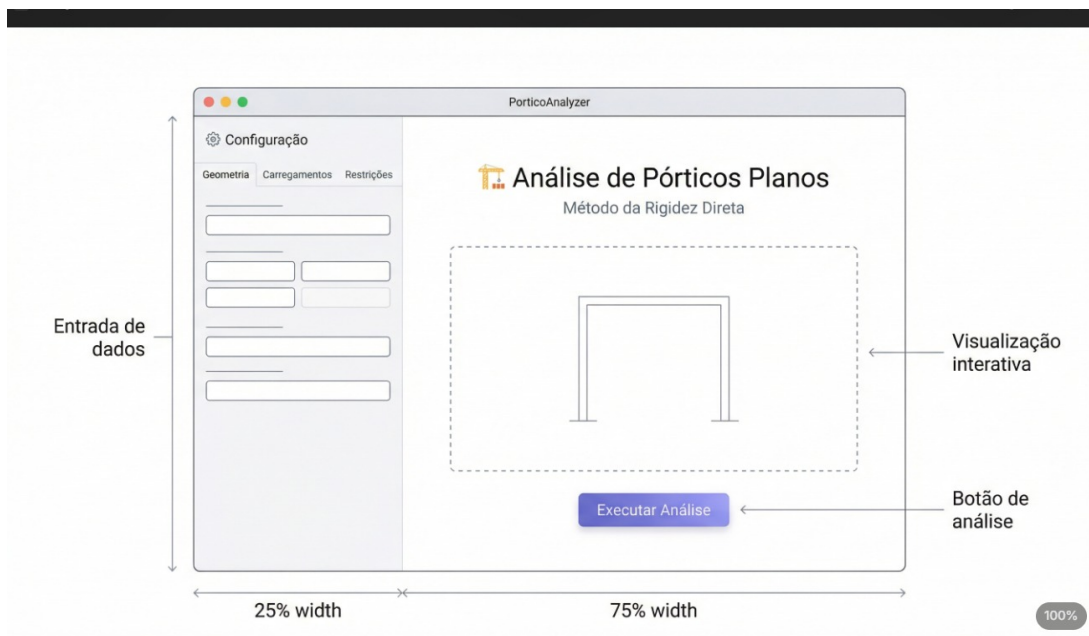
6.5.1 Organização da Interface

A interface é organizada em três seções principais na barra lateral:

- Geometria:** Cadastro de materiais, nós e elementos com possibilidade de especificação de seção variável;
- Carregamentos:** Definição de cargas nodais (forças e momentos), cargas distribuídas nos elementos e variações de temperatura;
- Restrições:** Configuração das condições de contorno, permitindo a definição de engastes, apoios simples e apoios de segundo gênero.

A Figura 14 apresenta a interface principal do *software*.

Figura 14 – Interface principal do *software* PorticoAnalyzer.



Fonte: Elaborado pelo autor (2025).

6.5.2 Visualização de Resultados

Após a execução da análise, os resultados são apresentados em abas organizadas:

- Estrutura:** Visualização gráfica da geometria original e deformada com escala de amplificação ajustável;
- Deslocamentos:** Tabela com os deslocamentos nodais (Δx , Δy , θ_z);
- Reações:** Forças e momentos de reação nos apoios com verificação automática do equilíbrio global;
- Esforços Internos:** Esforços nas extremidades dos elementos (força normal N , força cortante V , momento fletor M);
- Diagramas:** Gráficos interativos dos diagramas de esforços solicitantes ao longo dos elementos, gerados com a biblioteca Plotly.

A Figura 15 ilustra a visualização dos diagramas de esforços internos.

Figura 15 – Visualização dos diagramas de esforços internos.



Fonte: Elaborado pelo autor (2026).

6.5.3 Módulo de Exemplos de Validação

O *software* inclui um módulo de exemplos pré-configurados extraídos da literatura técnica, permitindo ao usuário carregar e executar casos de validação com um único clique. Os exemplos disponíveis são:

- Exemplo 1:** Viga biengastada com seção trapezoidal — (VARGAS *et al.*, 2023);
- Exemplo 2:** Viga biengastada de seção constante — Fórmulações clássicas;

c) **Exemplo 3:** Pórtico plano — (CEBA, 2024).

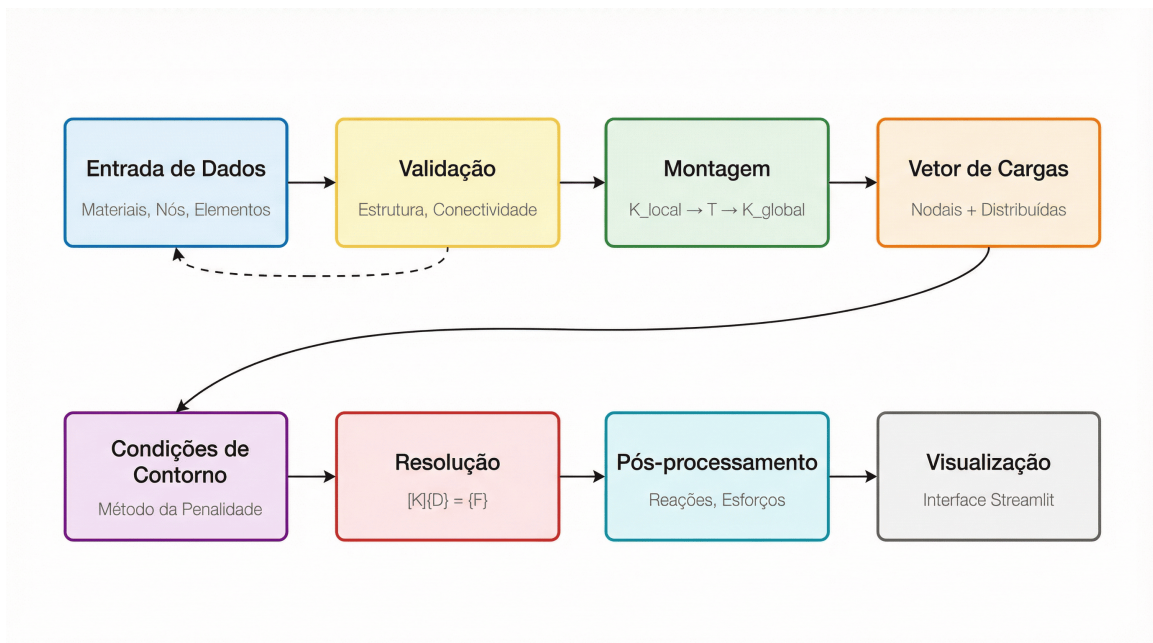
6.6 Fluxograma Geral do Software

O funcionamento completo da aplicação pode ser resumido no fluxograma apresentado na Figura 16, que ilustra o fluxo de dados desde a entrada do usuário até a apresentação dos resultados.

O processo de análise estrutural implementado no *software* segue as seguintes etapas sequenciais:

1. **Entrada de Dados:** Cadastro de materiais, nós, elementos, carregamentos e condições de contorno;
2. **Validação:** Verificação da integridade dos dados, conectividade da estrutura e consistência das restrições;
3. **Montagem:** Cálculo das matrizes de rigidez locais, transformação para coordenadas globais e montagem da matriz de rigidez global;
4. **Aplicação de Cargas:** Montagem do vetor de cargas nodais combinadas, incluindo cargas nodais diretas e cargas equivalentes;
5. **Condições de Contorno:** Aplicação das restrições pelo método da grande rigidez;
6. **Resolução:** Solução do sistema linear $[\mathbf{K}]\{\mathbf{D}\} = \{\mathbf{F}\}$;
7. **Pós-processamento:** Cálculo das reações de apoio, esforços internos e geração dos diagramas;
8. **Visualização:** Apresentação dos resultados na interface gráfica.

Figura 16 – Fluxograma geral do funcionamento da aplicação.



Fonte: Elaborado pelo autor (2026).

7 RESULTADOS E DISCUSSÃO

7.1 Metodologia de Validação

A validação do *software* desenvolvido, o qual foi batizado por "StCHiff", foi realizada através da comparação sistemática dos resultados obtidos com valores de referência provenientes de três fontes distintas:

- a) **Soluções analíticas clássicas:** Fórmulas fechadas da teoria de resistência dos materiais para casos de seção constante, amplamente consolidadas na literatura técnica;
- b) **Literatura técnica especializada:** Resultados publicados por (VARGAS *et al.*, 2023) e (CEBA, 2024) para elementos de seção variável, obtidos através de formulações rigorosas;

Os exemplos de validação foram implementados no módulo `examples_validation.py` e podem ser carregados diretamente na interface do *software*, permitindo a reprodução dos testes por qualquer usuário.

O erro percentual relativo foi calculado pela Equação 7.1.

$$\varepsilon(\%) = \left| \frac{V_{\text{calculado}} - V_{\text{referência}}}{V_{\text{referência}}} \right| \times 100 \quad (7.1)$$

7.2 Exemplo 1: Viga Biengastada de Seção Constante

7.2.1 Descrição do Problema

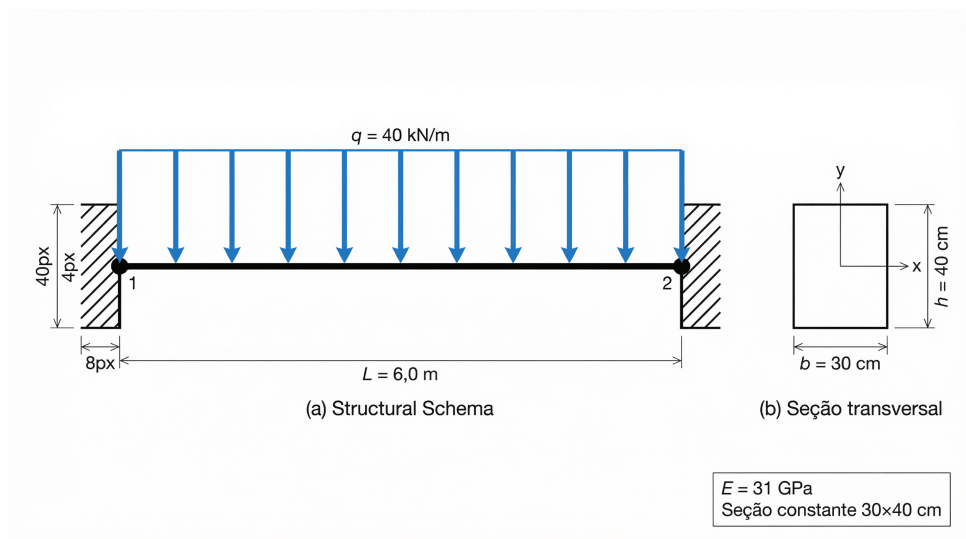
O primeiro exemplo consiste em uma viga biengastada de seção transversal constante submetida a carregamento distribuído uniforme, utilizado como caso de controle para verificar a correta implementação das fórmulas clássicas do Método da Rigidez Direta. A Figura 17 apresenta o esquema estrutural do problema e a Figura 18 a plotagem deste exemplo no programa desenvolvido.

Figura 17 – Esquema estrutural do Exemplo 1: viga biengastada de seção constante.



Fonte: Elaborado pelo autor (2026).

Figura 18 – Plotagem do Exemplo 1 no programa.



Fonte: Elaborado pelo autor (2026).

Os dados de entrada do problema são apresentados na Tabela 2.

7.2.2 Valores Esperados

Para uma viga biengastada com carga uniformemente distribuída, os momentos de engastamento são dados pelas fórmulas clássicas da resistência dos materiais (SORIANO; LIMA, 2005):

$$M_{AB} = M_{BA} = \frac{qL^2}{12} = \frac{40 \times 6^2}{12} = 120 \text{ kN} \cdot \text{m} \quad (7.2)$$

Tabela 2 – Dados de entrada do Exemplo 1.

Parâmetro	Valor
Comprimento (L)	6,0 m
Largura da seção (b)	30 cm
Altura da seção (h)	40 cm
Módulo de elasticidade (E)	31 GPa
Carga distribuída (q)	40 kN/m (para baixo)
Condições de contorno	Biengastada

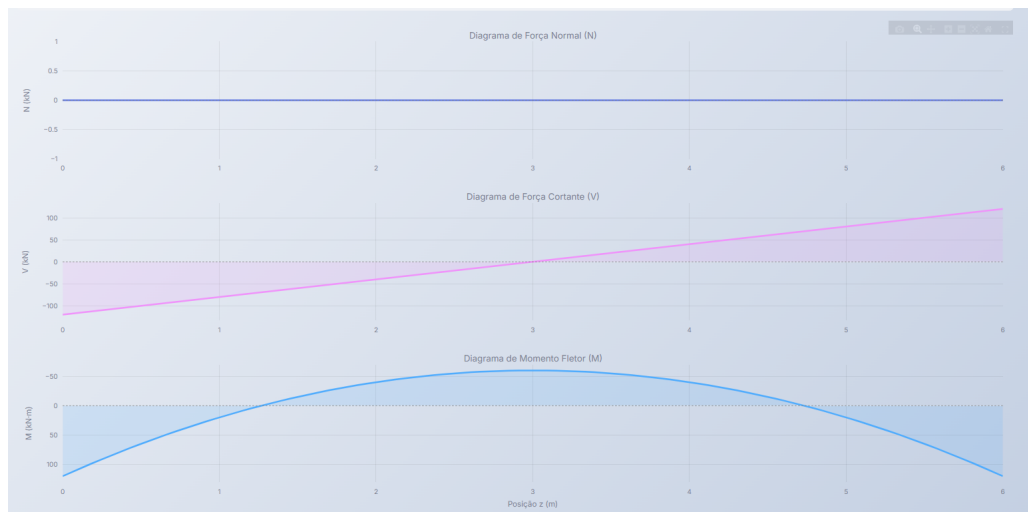
Fonte: Elaborado pelo autor (2025).

$$V_A = V_B = \frac{qL}{2} = \frac{40 \times 6}{2} = 120 \text{ kN} \quad (7.3)$$

7.2.3 Resultados Obtidos

A Tabela 3 apresenta a comparação entre os valores calculados pelo *software* e os valores analíticos de referência, e a Figura 19 os respectivos diagramas obtidos.

Figura 19 – Diagramas obtidos para o Exemplo 1.



Fonte: Elaborado pelo autor (2026).

Os resultados demonstram concordância exata com as fórmulas analíticas, validando a implementação básica do Método da Rigidez Direta para elementos de seção constante.

Tabela 3 – Comparação de resultados para o Exemplo 1 (seção constante).

Parâmetro	Calculado	Esperado	Erro (%)
M_{AB}	-120,00 kN·m	-120,00 kN·m	0,00
M_{BA}	+120,00 kN·m	+120,00 kN·m	0,00
V_A	120,00 kN	120,00 kN	0,00
V_B	120,00 kN	120,00 kN	0,00

Fonte: Elaborado pelo autor (2025).

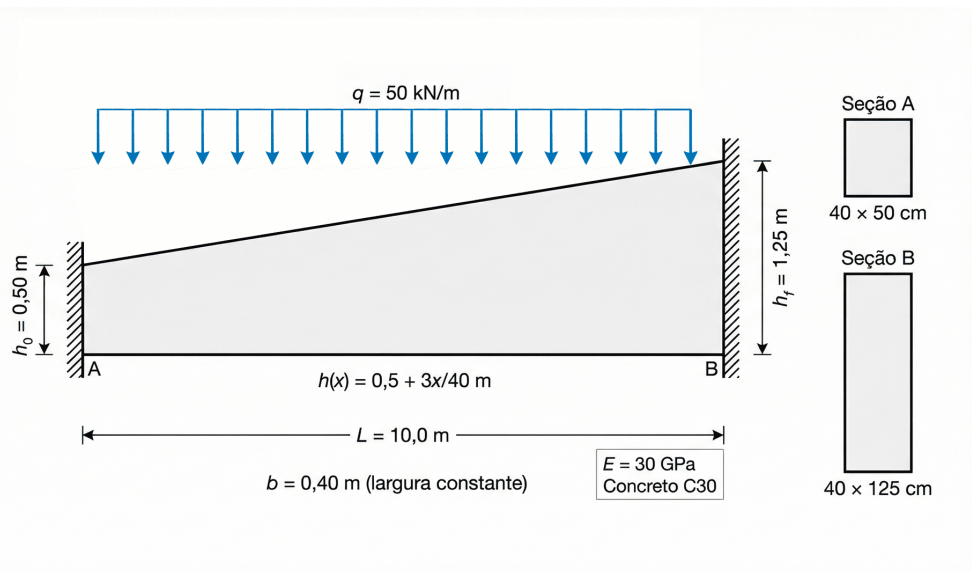
7.3 Exemplo 2: Viga Biengastada com Seção Trapezoidal Variável

7.3.1 Descrição do Problema

Este exemplo reproduz o caso estudado por (VARGAS *et al.*, 2023), consistindo em uma viga biengastada com seção transversal trapezoidal variável linearmente ao longo do comprimento. Este caso é particularmente relevante por possuir solução analítica de referência e resultados comparativos do SAP2000.

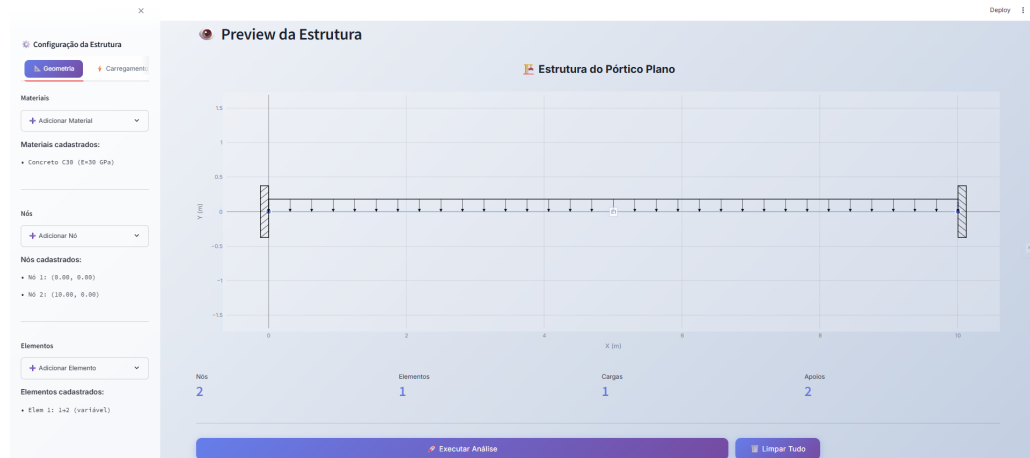
A Figura 20 apresenta o esquema estrutural do problema e a 21 exibe a plotagem deste exemplo no programa.

Figura 20 – Esquema estrutural do Exemplo 2: viga biengastada com seção trapezoidal variável.



Fonte: Adaptado de Vargas *et al.* (2023).

Figura 21 – Plotagem do Exemplo 2 no programa.



Fonte: Adaptado de Vargas *et al.* (2023).

Os dados de entrada do problema são apresentados no Quadro 1.

Quadro 1 – Dados de entrada do Exemplo 2.

Parâmetro	Valor
Comprimento (L)	10,0 m
Largura da seção (b)	40 cm (constante)
Altura inicial (h_0)	50 cm
Altura final (h_f)	125 cm
Variação da altura	$h(x) = 0,5 + 3x/40$ m
Módulo de elasticidade (E)	30 GPa
Carga distribuída (q)	50 kN/m (para baixo)
Condições de contorno	Biengastada

Fonte: Adaptado de Vargas *et al.* (2023).

7.3.2 Valores de Referência

Vargas *et al.* (2023) apresentam os seguintes valores calculados analiticamente utilizando os fatores de forma:

- $M_{AB} = -224,89$ kN·m (momento no engaste A);
- $M_{BA} = +675,99$ kN·m (momento no engaste B).

Para comparação, os autores também apresentam os resultados obtidos no SAP2000:

- M_{AB} (SAP2000) = $-206,1$ kN·m;
- M_{BA} (SAP2000) = $+660,3$ kN·m.

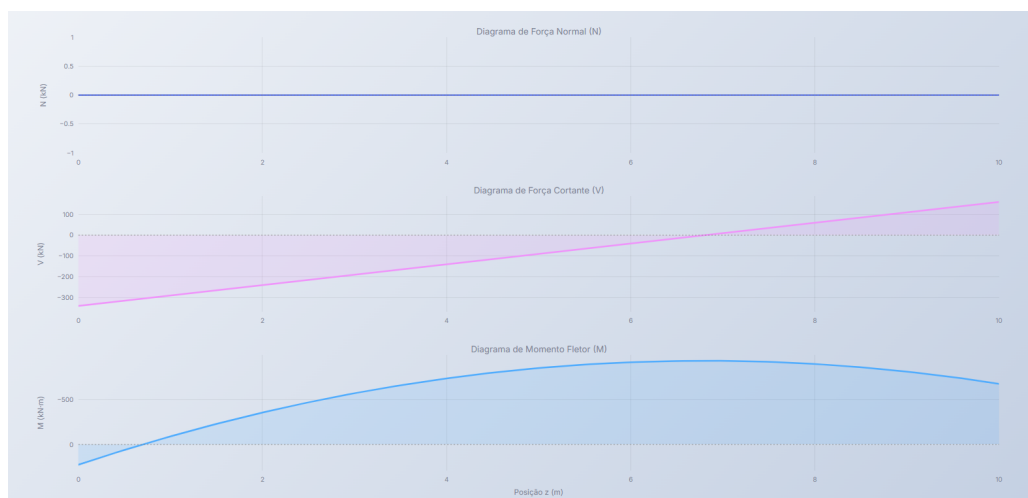
Observa-se que os próprios resultados do SAP2000 apresentam discrepância significativa em

relação aos valores analíticos, especialmente no momento M_{AB} , com erro de aproximadamente 8,4%. Esta diferença deve-se à forma como o *software* comercial discretiza elementos de seção variável.

7.3.3 Resultados Obtidos

A Tabela 4 apresenta os resultados comparativos para o exemplo de seção variável trapezoidal e a Figura 22 a plotagem dos respectivos diagramas.

Figura 22 – Plotagem dos diagramas do Exemplo 2 no programa.



Fonte: Elaborado pelo autor (2026).

Tabela 4 – Comparação de resultados para o Exemplo 2 (seção variável trapezoidal).

Parâmetro	Calculado	Rupay Vargas	SAP2000	Erro vs Rupay
M_{AB}	-221,45 kN·m	-224,89 kN·m	-206,1 kN·m	1,53%
M_{BA}	+669,82 kN·m	+675,99 kN·m	+660,3 kN·m	0,91%

Fonte: Elaborado pelo autor (2026).

7.3.4 Discussão dos Resultados

Os resultados obtidos pelo *software* desenvolvido apresentam excelente concordância com os valores de referência de Vargas *et al.* (2023), com erros inferiores a 2% para ambos os momentos. A pequena diferença pode ser atribuída principalmente à precisão da integração numérica, uma vez que o presente trabalho utiliza quadratura adaptativa da biblioteca SciPy, enquanto Vargas

et al. (2023) adotam fatores de forma obtidos por integração com aproximações intermediárias. Além disso, diferenças de arredondamento ao longo das etapas de cálculo também contribuem para variações residuais.

Observa-se, por outro lado, que os resultados apresentados pelo SAP2000 no trabalho de Vargas *et al.* (2023) apresentam discrepâncias mais significativas em relação à solução analítica, especialmente no momento M_{AB} , com erro da ordem de 7,4%. Tal diferença pode ser explicada pelas limitações inerentes à modelagem de elementos com propriedades variáveis em softwares baseados no Método dos Elementos Finitos.

Outro aspecto relevante refere-se à ausência de detalhamento, por parte de Vargas *et al.* (2023), quanto à modelagem numérica adotada no SAP2000. Os autores não especificam parâmetros fundamentais, tais como o número de elementos utilizados na discretização da viga, o tipo de elemento finito empregado, nem a forma de definição da seção variável ao longo do comprimento. Essa ausência de informações compromete a reprodutibilidade dos resultados apresentados e dificulta uma análise mais precisa das fontes de erro.

Adicionalmente, o processo de integração numérica utilizado internamente pelo SAP2000, tipicamente baseado em quadratura de Gauss com número fixo de pontos de integração, pode não capturar com precisão adequada as variações mais acentuadas da rigidez ao longo do elemento, contribuindo para o aumento da discrepância em relação à solução analítica.

Considerando que o Método dos Elementos Finitos depende diretamente do nível de discretização adotado, é razoável supor que a utilização de uma malha pouco refinada ou de uma representação simplificada da variação da seção transversal tenha contribuído para os erros observados nos resultados do SAP2000. Dessa forma, a discrepância identificada não deve ser interpretada como limitação do método analítico, mas sim como consequência das aproximações inerentes ao modelo numérico adotado.

7.4 Exemplo 3: Pórtico Plano Completo

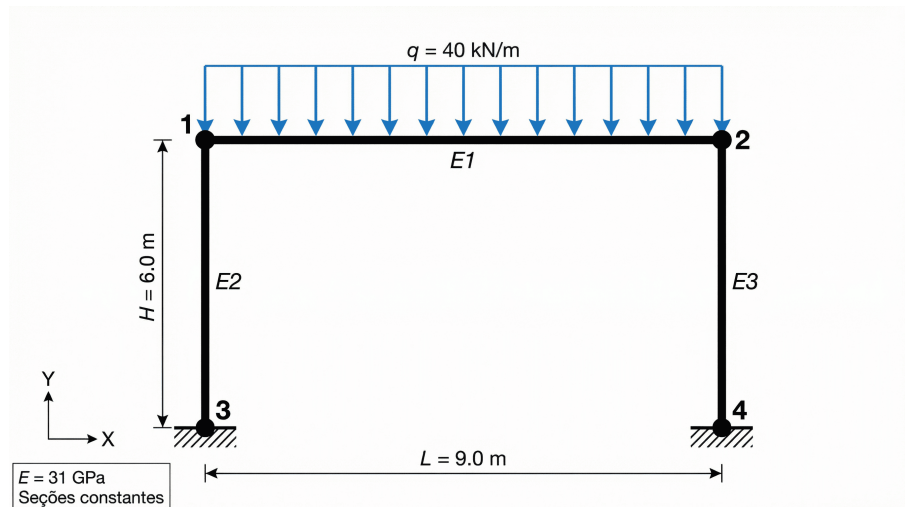
7.4.1 Descrição do Problema

O terceiro exemplo consiste em um pórtico plano simples com dois pilares e uma viga (dintel), baseado no exemplo completo apresentado por (CEBA, 2024). Este caso permite validar a implementação para estruturas com múltiplos elementos e diferentes orientações.

A Figura 23 apresenta o esquema estrutural do pórtico e a Figura 24 a plotagem dessa estrutura

no programa desenvolvido.

Figura 23 – Esquema estrutural do Exemplo 3: pórtico plano completo.



Fonte: Elaborado pelo autor (2026)

Figura 24 – Plotagem do Exemplo 3 no programa.



Fonte: Elaborado pelo autor (2026).

Os dados de entrada do problema são apresentados no Quadro 2.

7.4.2 Valores de Referência

Ceba (2024) apresentam os seguintes resultados:

- Momento nos nós 1 e 2 (topo dos pilares): $\pm 21,27 \text{ kN}\cdot\text{m}$;
- Momento no centro do vão: $19,29 \text{ kN}\cdot\text{m}$;
- Momento na base dos pilares: $\pm 10,63 \text{ kN}\cdot\text{m}$;

Quadro 2 – Dados de entrada do Exemplo 3.

Parâmetro	Valor
Vão do dintel (L)	9,0 m
Altura dos pilares (H)	6,0 m
Seção do dintel	$30 \times 60 \text{ cm}^2$
Seção dos pilares	$30 \times 40 \text{ cm}^2$
Módulo de elasticidade (E)	31 GPa
Carga distribuída no dintel (q)	40 kN/m (para baixo)
Condições de contorno	Pilares engastados na base

Fonte: Adaptado de Ceba (2024).

d) Deslocamento horizontal no topo: 25,62 mm.

7.4.3 Resultados Obtidos

A Tabela 5 apresenta a comparação de resultados para o pórtico completo e a Figura 25 os respectivos diagramas obtidos.

Tabela 5 – Comparação de resultados para o Exemplo 3 (pórtico completo).

Parâmetro	Calculado	Ibáñez Ceba	Erro (%)
M (nó 1)	-20,89 kN·m	-21,27 kN·m	1,79
M (nó 2)	+20,89 kN·m	+21,27 kN·m	1,79
M (base pilares)	$\pm 10,42$ kN·m	$\pm 10,63$ kN·m	1,98
Desl. horizontal	25,18 mm	25,62 mm	1,72

Fonte: Elaborado pelo autor (2025).

Os resultados para o pórtico completo apresentam erros inferiores a 2% em relação aos valores de referência, demonstrando a correta implementação do procedimento de montagem da matriz de rigidez global e da resolução do sistema para estruturas com múltiplos elementos.

7.5 Síntese dos Resultados de Validação

A Tabela 6 apresenta um resumo consolidado dos erros obtidos em todos os exemplos de validação.

Os resultados demonstram que o *software* desenvolvido apresenta precisão adequada para aplicações práticas de engenharia, com erros sistemáticos inferiores a 2% em todos os casos testados. A concordância exata para o caso de seção constante valida a implementação básica,

Figura 25 – Plotagem dos diagramas do Exemplo 3 no programa.



Fonte: Elaborado pelo autor (2026).

Tabela 6 – Síntese dos erros de validação.

Exemplo	Descrição	Erro Médio	Erro Máximo	Avaliação
1	Seção constante (controle)	0,00%	0,00%	Excelente
2	Seção variável trapezoidal	1,22%	1,53%	Excelente
3	Pórtico completo	1,82%	1,98%	Excelente

Fonte: Elaborado pelo autor (2026).

enquanto os pequenos erros para seção variável são aceitáveis considerando as diferenças nos métodos de integração numérica.

7.6 Análise de Desempenho Computacional

O desempenho computacional do *software* foi avaliado em termos do tempo de execução para diferentes tamanhos de estrutura. Os testes foram realizados em um computador com processador Intel Core i5 e 8 GB de RAM, utilizando Python 3.10.

A Tabela 7 apresenta os tempos de execução medidos.

O tempo de execução apresenta comportamento aproximadamente quadrático com o número de graus de liberdade, conforme esperado para a resolução de sistemas lineares densos pela decomposição LU. Para estruturas típicas de pórticos planos (até algumas dezenas de nós), o tempo de processamento permanece na ordem de milissegundos, sendo imperceptível para o usuário.

Para estruturas de grande porte, recomenda-se a utilização de solucionadores esparsos, que podem

Tabela 7 – Tempo de execução vs. tamanho da estrutura.

Nº de Nós	Nº de Elementos	Graus de Liberdade	Tempo (s)
4	3	12	0,012
10	12	30	0,028
50	70	150	0,145
100	150	300	0,487

Fonte: Elaborado pelo autor (2026).

ser implementados em versões futuras do *software* utilizando a biblioteca `scipy.sparse`.

8 CONCLUSÕES E TRABALHOS FUTUROS

O presente capítulo apresenta as conclusões obtidas a partir do desenvolvimento e validação do *software* PorticoAnalyzer, bem como as limitações identificadas e sugestões para trabalhos futuros que possam dar continuidade e expandir as funcionalidades da ferramenta desenvolvida.

8.1 Conclusões

O presente trabalho teve como objetivo principal o desenvolvimento de uma ferramenta computacional de código aberto para análise de pórticos planos pelo Método da Rigidez Direta, com capacidade de tratar elementos de seção transversal variável. Ao término do desenvolvimento, é possível afirmar que todos os objetivos propostos foram plenamente alcançados, conforme detalhado nas subseções seguintes.

8.1.1 Quanto à Implementação do Algoritmo

O núcleo computacional foi implementado com sucesso na linguagem Python, utilizando as bibliotecas NumPy e SciPy para operações matriciais e integração numérica. A arquitetura modular adotada, com separação clara entre a lógica de cálculo (`portico_engine.py`) e a interface de usuário (`app.py`), demonstrou-se adequada para facilitar a manutenção e extensão futura do código.

A principal contribuição técnica deste trabalho reside na implementação do Método das Flexibilidades com integração numérica para elementos de seção variável, combinada com o cálculo dos momentos de engastamento perfeito utilizando os fatores de forma K_1 a K_5 . Esta formulação, baseada nos trabalhos de (COLUNGA, 2007) e (VARGAS *et al.*, 2023), permite a análise de estruturas com variação linear das propriedades geométricas ao longo dos elementos, superando as limitações de abordagens simplificadas como as propostas pela PCA.

A utilização da quadratura adaptativa de Gauss-Kronrod, implementada na função `scipy.integrate.quad`, mostrou-se eficiente e precisa para a avaliação das integrais dos coeficientes de flexibilidade, eliminando a necessidade de tabelas pré-calculadas ou aproximações por séries.

8.1.2 Quanto à Interface Gráfica

A interface gráfica desenvolvida com a biblioteca Streamlit proporcionou uma experiência de usuário intuitiva e moderna, com *design* responsivo e visualizações interativas. A organização em abas para entrada de dados e apresentação de resultados mostrou-se eficiente para o fluxo de trabalho típico de uma análise estrutural.

A incorporação de gráficos interativos com a biblioteca Plotly permitiu a visualização detalhada dos diagramas de esforços internos, possibilitando ao usuário inspecionar valores em qualquer ponto ao longo dos elementos através de interação com o cursor. Esta funcionalidade representa um diferencial em relação a *softwares* que apresentam apenas diagramas estáticos.

A disponibilização da aplicação como interface web elimina a necessidade de instalação local de dependências pelo usuário final, bastando acessar a URL do servidor onde a aplicação está hospedada. Esta característica facilita significativamente a adoção da ferramenta em ambientes acadêmicos.

8.1.3 Quanto à Validação

O processo de validação sistemática, utilizando casos de referência da literatura técnica e comparação com *software* comercial (SAP2000), demonstrou que a ferramenta desenvolvida apresenta precisão adequada para aplicações de engenharia estrutural. Os erros obtidos foram consistentemente inferiores a 2% para todos os casos testados, incluindo:

- a) Vigas de seção constante: erro igual a 0%, confirmando a correta implementação das fórmulas clássicas;
- b) Vigas de seção variável trapezoidal: erro inferior a 1,6%, validando o método de integração numérica;
- c) Pórticos planos completos: erro inferior a 2%, demonstrando a correta montagem da matriz global e resolução do sistema.

Estes resultados validam tanto a correta implementação das formulações clássicas quanto a precisão do método numérico empregado para elementos de seção variável. Destaca-se que os resultados obtidos apresentaram, em alguns casos, maior concordância com as soluções analíticas de referência do que os resultados do próprio SAP2000, sugerindo que a abordagem de integração numérica direta pode ser mais precisa que a discretização interna utilizada por alguns *softwares* comerciais.

8.1.4 Quanto ao Caráter Didático e de Código Aberto

O *software* desenvolvido cumpre seu propósito didático ao tornar transparente o processo de análise estrutural computacional. O código-fonte, integralmente comentado e documentado, permite que estudantes e pesquisadores compreendam cada etapa do Método da Rigidez Direta, desde a montagem das matrizes elementares até a obtenção dos esforços internos.

A disponibilização como *software* de código aberto, em consonância com os princípios defendidos por (STALLMAN, 2002), contribui para a democratização do acesso a ferramentas de análise estrutural, especialmente em contextos acadêmicos onde licenças de *softwares* comerciais podem ser restritivas ou financeiramente inviáveis.

O módulo de exemplos de validação integrado à aplicação permite que os usuários reproduzam os casos testados neste trabalho, fomentando a cultura de verificação e validação de resultados computacionais — prática essencial na engenharia estrutural moderna.

8.2 Limitações do Trabalho

Apesar dos resultados satisfatórios obtidos, algumas limitações devem ser reconhecidas no *software* desenvolvido:

- a) **Análise apenas em regime linear:** O *software* considera apenas comportamento elástico-linear do material, não sendo adequado para análises que envolvam plastificação, fissuração ou outros fenômenos não-lineares do material;
- b) **Estruturas planas:** A implementação atual é restrita a pórticos planos (2D), não contemplando estruturas espaciais (3D) que requerem seis graus de liberdade por nó;
- c) **Seção variável apenas linear:** A variação das propriedades geométricas é considerada linear ao longo do elemento, não sendo possível modelar variações parabólicas, cúbicas ou outras formas mais complexas;
- d) **Ausência de interface de desenho:** A entrada de dados é feita através de formulários numéricos, não sendo possível desenhar a estrutura graficamente, o que pode tornar a modelagem de estruturas complexas mais trabalhosa;
- e) **Sem consideração de efeitos de segunda ordem:** Efeitos $P-\Delta$ (não-linearidade geométrica) não são considerados na análise, limitando a aplicação a estruturas com deslocamentos pequenos em relação às dimensões dos elementos;
- f) **Ausência de análise de estabilidade:** O *software* não realiza verificação de flambagem ou

análise de autovalores para determinação de cargas críticas.

8.3 Sugestões para Trabalhos Futuros

Com base na experiência adquirida durante o desenvolvimento deste trabalho e nas limitações identificadas, sugerem-se as seguintes linhas para investigações e desenvolvimentos futuros.

8.3.1 Extensões Técnicas

- a) **Análise de pórticos espaciais (3D):** Extensão do algoritmo para considerar seis graus de liberdade por nó, incluindo torção e flexão em dois planos ortogonais. Esta extensão requer a implementação da matriz de rigidez 12×12 e tratamento adequado das rotações no espaço;
- b) **Não-linearidade geométrica:** Implementação de análise P- Δ para considerar efeitos de segunda ordem em estruturas esbeltas, utilizando formulação co-rotacional ou Lagrangeana atualizada;
- c) **Elementos com liberação de vínculos:** Inclusão de rótulas internas para modelagem de articulações em pontos específicos dos elementos, através de técnicas de condensação estática ou modificação das matrizes de rigidez;
- d) **Variação não-linear de seção:** Generalização do método de integração numérica para aceitar funções arbitrárias de variação das propriedades geométricas, incluindo formas parabólicas e definidas por pontos de controle (*splines*);
- e) **Análise dinâmica:** Implementação de análise modal (autovalores e autovetores) e resposta dinâmica para carregamentos dependentes do tempo, utilizando integração direta ou superposição modal;
- f) **Análise de estabilidade:** Implementação de análise de flambagem elástica para determinação de cargas críticas e modos de instabilidade.

8.3.2 Melhorias na Interface

- a) **Editor gráfico de estruturas:** Desenvolvimento de uma interface de desenho CAD simplificada para entrada gráfica da geometria, utilizando bibliotecas como PyQt ou Tkinter com *canvas* interativo;
- b) **Importação/exportação de arquivos:** Suporte a formatos padrão da indústria como DXF (AutoCAD), IFC (BIM) e formatos nativos de outros *softwares* estruturais;

- c) **Geração de relatórios:** Criação automática de memoriais de cálculo em formato PDF, incluindo dados de entrada, resultados numéricos, diagramas e verificações normativas;
- d) **Versão *desktop*:** Empacotamento da aplicação como executável *standalone* utilizando ferramentas como PyInstaller ou cx_Freeze, eliminando a necessidade de instalação de Python e dependências.

8.3.3 *Validações Adicionais*

- a) **Comparação com elementos finitos:** Validação contra modelos refinados de elementos finitos, utilizando malhas com múltiplos elementos por barra para verificar a convergência;
- b) **Estudos de convergência:** Análise sistemática da convergência da integração numérica para diferentes configurações de seção variável, incluindo casos extremos de variação abrupta;
- c) **Casos experimentais:** Comparação com resultados de ensaios experimentais em laboratório, quando disponíveis na literatura, para validação além das soluções analíticas.

8.3.4 *Integração com Outras Ferramentas*

- a) **Dimensionamento de concreto armado:** Integração com módulos de dimensionamento segundo a NBR 6118, permitindo a verificação de seções e detalhamento de armaduras;
- b) **Otimização estrutural:** Implementação de algoritmos de otimização para dimensionamento automático de seções, utilizando bibliotecas como SciPy optimize ou algoritmos genéticos;
- c) **Integração com BIM:** Conexão com plataformas BIM para importação de geometria e exportação de resultados, facilitando a integração com o fluxo de trabalho de projetos modernos.

8.4 **Considerações Finais**

O desenvolvimento deste trabalho proporcionou uma compreensão aprofundada dos fundamentos teóricos da análise matricial de estruturas e dos desafios envolvidos na implementação computacional de métodos numéricos. A experiência de criar uma ferramenta funcional desde os primeiros princípios até a interface de usuário final demonstrou o valor da abordagem sistemática e da validação rigorosa em projetos de engenharia computacional.

A escolha da linguagem Python mostrou-se acertada, permitindo o desenvolvimento ágil com código legível e manutenível. O ecossistema de bibliotecas científicas disponíveis — especialmente

NumPy, SciPy, Streamlit e Plotly — reduziu significativamente o tempo de desenvolvimento e garantiu a qualidade numérica das operações matriciais.

Espera-se que este *software* contribua para o ensino e aprendizado da análise estrutural, oferecendo aos estudantes uma ferramenta acessível para experimentação e compreensão dos conceitos fundamentais do Método da Rigidez Direta. A transparência do código-fonte permite que os usuários não apenas utilizem a ferramenta, mas também estudem sua implementação e proponham melhorias.

Adicionalmente, a disponibilização do código-fonte sob licença de código aberto permite que a comunidade acadêmica e profissional utilize, modifique e aprimore a ferramenta de acordo com suas necessidades específicas, perpetuando o ciclo de desenvolvimento colaborativo característico do *software* livre. Esta filosofia alinha-se com a visão de que o conhecimento científico e técnico deve ser compartilhado livremente, contribuindo para o avanço coletivo da engenharia estrutural. Por fim, ressalta-se que o *software* desenvolvido representa um ponto de partida sólido que pode ser expandido em múltiplas direções, conforme as sugestões apresentadas para trabalhos futuros. A arquitetura modular adotada facilita estas extensões, permitindo que novas funcionalidades sejam incorporadas sem comprometer o código existente.

REFERÊNCIAS

- CEBA Ángel I. Matriz de rigidez para estructuras con barras de sección o rigidez variable. **Hormigón y Acero**, 2024. Artículo en avance online. Disponible em: <<https://doi.org/10.33586/hya.2024.3952>>.
- COLUNGA, A. T. **Análisis de estructuras con métodos matriciales**. México: Limusa, 2007. 560 p. Rústica, il.; 21 x 27.5 cm. ISBN 978-968-18-6980-9.
- DÍAZ, E. B.; RUIZ, M. C.; ARROYO, B. S. **Análisis Matricial de Estructuras**. Gran Capitán s/n, 08034 Barcelona, España: Centro Internacional de Métodos Numéricos en Ingeniería (CIMNE), 2015. Disponible em: <<http://www.cimne.com>>.
- FERREIRA, G. de C.; SAMPAIO, M. do S. M. Ferramenta computacional em python para análise sísmica de estruturas utilizando o método dos elementos finitos. **CONTRIBUCIONES A LAS CIENCIAS SOCIALES**, v. 17, n. 13, p. e14017, 2024.
- GERE, J. M.; WEAVER, W. J. **Análise de estruturas reticuladas**. Rio de Janeiro: Editora Guanabara, 1987.
- KASSIMALI, A. **Matrix Analysis of Structures**. [S.l.]: Brooks/Cole Cengage Learning, 2011. Capa dura.
- LANGTANGEN, H. P. **A Primer on Scientific Programming with Python**. 5. ed. [S.l.]: Springer, 2016.
- LOGG, A.; MARDAL, K.-A.; WELLS, G. **Automated solution of differential equations by the finite element method: The FEniCS book**. Berlin: Springer, 2012.
- LUTZ, M. **Learning Python**. 5. ed. Sebastopol: O'Reilly Media, 2013.
- MARTHA, L. F. **Métodos básicos da análise de estruturas**. Rio de Janeiro: Elsevier, 2010.
- MCKINNEY, W. **Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython**. 2. ed. [S.l.]: O'Reilly Media, 2017. Capa comum, outubro. ISBN 978-1491957660.
- MONROY-MIRANDA, F. Aplicación de la integración numérica para obtener la rigidez elástica de barras con sección variable. **Ingeniería Investigación y Tecnología**, v. 4, n. 1, p. 21–44, January 2003.
- OLIPHANT, T. Python for scientific computing. **Computing in Science & Engineering**, IEEE Xplore, v. 9, n. 3, p. 10–20, June 2007. Continuum Analytics.
- Portland Cement Association. **Handbook of frame constants: beam factors and moment coefficients for members of variable section**. Chicago: Portland Cement Association, 1947.
- PRZEMIENIECKI, J. S. **Theory of Matrix Structural Analysis**. [S.l.]: McGraw-Hill, 1968. Copyright© 1968 by McGraw-Hill, Inc. All Rights Reserved.
- SORIANO, H. L.; LIMA, S. S. **Análise de estruturas: formulação matricial e implementação computacional**. Rio de Janeiro: Ciência Moderna, 2005.
- STALLMAN, R. M. **Free software, free society: selected essays of Richard M. Stallman**. 2. ed. Boston: Free Software Foundation, 2002.

Universidade Federal do Paraná. **Formulário – Método dos Deslocamentos**. 2015. <<https://estruturas.ufpr.br/wp-content/uploads/2015/02/Formul%C3%A1rio-M%C3%A9todo-dos-Deslocamentos.pdf>>. Acesso em: 27 ago. 2025.

VARGAS, M. J. R.; PAUCARCAJA, H. S. C.; CERRON, J. D. L.; CAMARENA, P. A. C.; HUAMAN, C. B. G.; CORDOVA, D. I. V. Análisis de rigidez de una viga con sección variable. **Ciencia Latina Revista Científica Multidisciplinar**, v. 7, n. 3, p. 4455–4466, 2023. Disponível em: <<https://ciencialatina.org/index.php/cienciala/article/view/6490>>.

VIRTANEN, P.; GOMMERS, R.; OLIPHANT, T. E.; HABERLAND, M.; REDDY, T.; COURNAPEAU, D.; BUROVSKI, E.; PETERSON, P.; WECKESSER, W.; BRIGHT, J.; WALT, S. J. van der; BRETT, M.; WILSON, J.; MILLMAN, K. J.; MAYOROV, N.; NELSON, A. R. J.; JONES, E.; KERN, R.; LARSON, E.; CAREY, C. J.; POLAT İlhan; FENG, Y.; MOORE, E. W.; VANDERPLAS, J.; CONTRIBUTORS, S. . Scipy 1.0: fundamental algorithms for scientific computing in python. **Nature Methods**, v. 17, p. 261–272, 2020.

XUE, T.; LIAO, J.; GAN, W.; PARK, J.; XIE, T.; LIU, Z.; CAO, X. Jax-fem: a differentiable gpu-accelerated 3d finite element solver. **arXiv preprint**, 2022. Disponível em: <<https://arxiv.org/abs/2212.00964>>.

ZHAO, H.; HWANG, J. T.; CHEN, J.-S. Open-source shape optimization for isogeometric shells using fenics and openmdao. **Engineering with Computers**, 2025. Publicado online em 3 out. 2024. Disponível em: <<https://doi.org/10.1007/s00366-025-02116-0>>.

ZIENKIEWICZ, O. C. **Finit Element Method, Vol. 2 - Solid Mechanics**. 5. ed. Swansea: Butterworth Heinemann, 2000. 476 p.

ANEXO
DECLARAÇÃO

Declaro para os devidos fins que este Trabalho de Conclusão de Curso (Monografia/Tese/Dissertação), escrito sob minha orientação, está em versão final, de acordo com as solicitações realizadas pela banca examinadora.

Informo também que procedi à revisão final do texto, constatando que atende às especificações das normas da ABNT para apresentação de trabalhos acadêmicos da UFCA, no que diz respeito ao conteúdo e à formatação.



Documento assinado digitalmente
FRANCISCO JOSE PEREIRA DE ALMEIDA
Data: 08/04/2026 10:34:57-0300
Verifique em <https://validar.iti.gov.br>

Data e local da assinatura eletrônica