

UNIVERSIDADE FEDERAL DO AMAZONAS
FACULDADE DE TECNOLOGIA
CURSO DE ENGENHARIA CIVIL

ERLI GOMES DA SILVA

DESENVOLVIMENTO DE UM PROGRAMA COMPUTACIONAL PARA
ANÁLISE MODAL DE ESTRUTURAS RETICULADAS PLANAS

MANAUS

2025

ERLI GOMES DA SILVA

DESENVOLVIMENTO DE UM PROGRAMA COMPUTACIONAL PARA
ANÁLISE MODAL DE ESTRUTURAS RETICULADAS PLANAS

Monografia apresentada à disciplina FTC-224
Trabalho de Conclusão de Curso como parte dos
requisitos necessários para a obtenção do título de
Engenheiro Civil.

Orientador: Prof. Dr. Henrique Barbosa Soares

MANAUS

2025

Ficha Catalográfica

Elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

S586d Silva, Erli Gomes da
Desenvolvimento de um programa computacional para análise modal de estruturas reticuladas planas / Erli Gomes da Silva. - 2025.
97 f. : il., color. ; 31 cm.

Orientador(a): Henrique Barbosa Soares.
Trabalho de Conclusão de Curso (graduação) - Universidade Federal do Amazonas, Faculdade de Tecnologia, Curso de Engenharia Civil, Manaus, 2025.

1. Análise modal. 2. Método da rigidez direta. 3. Pórtico plano. 4. Python. I. Soares, Henrique Barbosa. II. Universidade Federal do Amazonas. Faculdade de Tecnologia. Curso de Engenharia Civil. III. Título

ERLI GOMES DA SILVA

DESENVOLVIMENTO DE UM PROGRAMA COMPUTACIONAL PARA ANÁLISE
MODAL DE ESTRUTURAS RETICULADAS PLANAS

Monografia apresentada à disciplina FTC-224
Trabalho de Conclusão de Curso como parte dos
requisitos necessários para a obtenção do título de
Engenheiro Civil.

Este trabalho foi defendido e aprovado pela banca em 18/12/2025.

BANCA EXAMINADORA

Prof. Dr. Henrique Barbosa Soares - UFAM
Orientador

Prof. Dr. Regis Pamponet da Fonseca – UFAM
Avaliador

Prof. Dr. Wagner Queiroz Silva – UFAM
Avaliador

Dedico aos meus pais

AGRADECIMENTOS

Aos meus pais, por todo amor e esforço com que fui presenteado. Eles são o principal motivo da realização desse sonho.

Aos meus amigos, colegas e todos aqueles que contribuíram para meu crescimento ao longo dessa jornada. Em especial aqueles poucos com quem pude compartilhar sonhos, é uma honra poder chamá-los de amigos.

Ao corpo docente do Curso de Engenharia Civil da Universidade Federal, todos os professores com quem tive o privilégio de aprender a visualizar o mundo de outra forma. Um agradecimento especial ao meu orientador, Prof. Dr. Henrique Barbosa Soares, pelo direcionamento, inesgotável paciência e por não ter me abandonado na jornada.

RESUMO

SILVA, G. E. **Desenvolvimento de um programa computacional para análise modal de estruturas reticuladas planas.** 2025. Trabalho de Conclusão de Curso - Faculdade de Tecnologia, Universidade Federal do Amazonas, Manaus, 2025.

Este trabalho apresenta o desenvolvimento de um programa computacional em linguagem *Python* para a análise modal de estruturas reticuladas planas, com foco na determinação das frequências naturais e das formas modais associadas. A formulação adotada baseia-se no Método da Rigidez Direta, com implementação do elemento de pórtico plano e montagem das matrizes globais de rigidez e de massa, considerando matriz de massa consistente. A partir da aplicação das condições de contorno, o problema modal foi resolvido na forma de autovalores generalizada, utilizando as submatrizes correspondentes aos graus de liberdade livres. O modelo estrutural foi definido por meio de planilhas eletrônicas, permitindo padronização da entrada de dados e organização dos resultados. A validação do programa foi realizada por comparação com o *Autodesk Robot Structural Analysis Professional*, adotado como referência comercial, em dois exemplos de pórticos planos com diferentes arranjos. Como resultado para os três primeiros modos de vibração, observou-se concordância próxima entre as frequências naturais obtidas, com erros relativos da ordem de 1%, indicando consistência na implementação das rotinas de montagem matricial, aplicação de restrições e solução numérica do problema modal. O programa desenvolvido constitui uma ferramenta reprodutível e de caráter didático, aplicável a estudos preliminares do comportamento dinâmico de estruturas planas dentro das hipóteses de linearidade e vibração livre não amortecida.

Palavras-chave: análise modal; método da rigidez direta; pórtico plano; *Python*.

ABSTRACT

SILVA, G. E. **Development of a computational program for modal analysis of frame structures.** 2025. Graduation work - College of Engineering, Federal University of Amazonas, Manaus, 2025.

This study develops a *Python* based computational program for the modal analysis of planar reticulated structures, with emphasis on determining natural frequencies and the corresponding mode shapes. The formulation is based on the Direct Stiffness Method, incorporating the plane frame element and assembling the global stiffness and mass matrices with a consistent mass formulation. After the enforcement of boundary conditions, the modal problem is cast and solved as a generalized eigenvalue problem using the reduced submatrices associated with the free degrees of freedom. The structural model is defined via spreadsheet input files, which standardize data entry and facilitate the organization of output results. Program validation is carried out by comparison with Autodesk *Robot* Structural Analysis Professional, adopted as a commercial reference, using two planar frame examples with distinct layouts. For the first three vibration modes, close agreement is obtained between the computed natural frequencies, with relative errors on the order of 1%, indicating consistency in the implementation of the matrix assembly routines, constraint handling, and numerical solution of the modal problem. The resulting program constitutes a reproducible and pedagogical tool, suitable for preliminary studies of the dynamic behavior of planar structures under the assumptions of linear response and undamped free vibration.

Keywords: modal analysis; direct stiffness method; planar frame; *Python*.

SUMÁRIO

1	INTRODUÇÃO	11
1.1	OBJETIVOS	12
1.1.1	Objetivos Específicos	12
1.2	JUSTIFICATIVA	12
1.3	METODOLOGIA.....	13
1.4	ORGANIZAÇÃO DO TRABALHO	14
2	REVISÃO BIBLIOGRÁFICA	16
2.1	ANÁLISE ESTRUTURAL	16
2.2	ANÁLISE MODAL.....	18
3	MÉTODO DA RIGIDEZ DIRETA	24
3.1	CONCEITOS FUNDAMENTAIS	24
3.2	DISCRETIZAÇÃO DA ESTRUTURA	25
3.3	VETOR DE FORÇAS	28
3.4	SISTEMA DE COORDENADAS.....	29
3.5	MATRIZ DE RIGIDEZ.....	33
3.5.1	Coefficientes de rigidez.....	34
3.5.2	Transformação de coordenadas	36
3.5.3	Montagem da matriz de rigidez global	37
3.6	CONDIÇÕES DE CONTORNO.....	39
4	ANÁLISE MODAL	42
4.1	MATRIZ DE MASSA	42
4.1.1	Formulação da matriz de massa de elemento	42
4.1.2	Matriz de massa consistente e montagem global	43
4.2	PROBLEMA GENERALIZADO DE AUTOVALOR	45
4.2.1	Equação de movimento e solução harmônica	45
4.2.2	Equação característica e interpretação dos autovares.....	46
4.2.3	Formulação em termos dos graus de liberdade livres.....	47
4.2.4	Resolução numérica no programa computacional	48
5	PROGRAMA COMPUTACIONAL	50
5.1	CONSIDERAÇÕES GERAIS.....	50

5.2	FUNCIONAMENTO DO PROGRAMA	52
5.2.1	Estrutura geral do código	53
5.2.2	Entrada e pré-processamento dos dados	55
5.2.3	Montagem das matrizes globais de rigidez e de massa	57
5.2.4	Aplicação das condições de contorno e solução dos problemas estático e modal	58
5.2.5	Saída de resultados e interface com o usuário	59
6	EXEMPLOS NUMÉRICOS	61
6.1	VALIDAÇÃO DO PROGRAMA	61
6.2	COMPARAÇÃO QUALITATIVA DAS FORMAS MODAIS.....	62
6.3	EXEMPLO 1	62
6.3.1	Geometria e conectividade	63
6.3.2	Propriedades do material, seção e massa	64
6.3.3	Condições de contorno e hipótese de modelo plano.....	65
6.3.4	Análise modal e critérios de comparação	65
6.4	EXEMPLO 2	70
6.4.1	Geometria e conectividade	70
6.4.2	Propriedades do material, seção e massa	71
6.4.3	Resultados e discussão	72
7	CONSIDERAÇÕES FINAIS.....	76
7.1	CONCLUSÕES	76
7.2	SUGESTÕES PARA FUTUROS TRABALHOS.....	77
	REFERÊNCIAS BIBLIOGRÁFICAS.....	78
	APÊNDICE A – CÓDIGO EM <i>PYTHON</i> DESENVOLVIDO	79

1 INTRODUÇÃO

O interesse humano pelas vibrações remonta à invenção dos primeiros instrumentos musicais, marcando o início de uma jornada em busca da compreensão dos fenômenos vibratórios. Desde então, a análise crítica e o método científico têm sido aplicados ao estudo das vibrações, baseando-se no fato de que tanto objetos cotidianos quanto estruturas complexas exibem comportamentos vibratórios sob certas condições. A análise das vibrações transcendeu sua origem musical, expandindo-se para campos como a engenharia estrutural, onde uma compreensão aprofundada do comportamento dinâmico é essencial nas construções modernas.

A importância de compreender as vibrações no contexto estrutural é destacada por Rao (2011), que a descreve como um elemento-chave na análise de segurança e desempenho de estruturas mecânicas e civis. Através dessas investigações, o estudo da vibração evoluiu da simples curiosidade humana para uma ciência rigorosa e fundamental para o avanço da engenharia.

Nas últimas décadas, o campo da engenharia civil e estrutural experienciou avanços significativos impulsionados pela integração de tecnologias computacionais no processo de análise, projeto e avaliação de estruturas. Conforme comentado por Soriano (2014), a disponibilidade de diversos e eficientes programas de computador, aliada ao aumento da eficiência de processamento dos microcomputadores, tem tornado cada vez mais frequente a análise dinâmica.

De acordo com Hibbeler (2015), uma estrutura é definida como um conjunto de elementos que estão interligados e destinados a suportar e transferir cargas. Estes elementos podem ser barras, vigas, colunas, ou placas, e a combinação desses componentes resulta em uma configuração que proporciona resistência e estabilidade para suportar cargas aplicadas.

Estruturas reticuladas, conforme explicado por Martha (2010), são modeladas por elementos lineares de barras com um eixo principal bem definido. A NBR 6118 (ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS, 2023), no item 14.4.1, define como elementos lineares aqueles cujo comprimento longitudinal é, no mínimo, três vezes maior do que a maior dimensão da seção transversal.

Estruturas são sistemas físicos sujeitos a ações externas que podem variar com o tempo e são classificadas como dinâmicas quando desenvolvem forças de inércia relevantes. Essas ações causam vibrações nas estruturas, que podem não apenas danificá-las, mas também provocar fadiga nos seus materiais e afetar a utilidade das mesmas (Soriano, 2014).

A análise modal, um método essencial na dinâmica estrutural, proporciona uma compreensão detalhada do comportamento vibratório dos arranjos estruturais. Esse método permite aos engenheiros determinarem as frequências naturais e modos de vibração de uma estrutura, obtendo subsídios para detecção de potenciais problemas como ressonância ou respostas dinâmicas indesejadas, tornando possível a adoção de medidas preventivas e/ou corretivas adequadas.

Dada a relevância da análise dinâmica, torna-se necessário o emprego adequado dos conceitos da teoria das estruturas, aliados à modelagem computacional dos arranjos estruturais, de modo a garantir resultados confiáveis. Com esse raciocínio, o presente trabalho fundamenta-se no método da rigidez direta e nos conceitos da física dos sistemas vibratórios simplificados, oferecendo uma introdução a análise modal que, em geral, não é abordada nos cursos de graduação em Engenharia Civil.

1.1 OBJETIVOS

Desenvolver um programa computacional para análise modal em estruturas reticuladas planas.

1.1.1 Objetivos Específicos

- Estudar as formulações referentes ao problema dinâmico e à análise modal de estruturas;
- Elaborar rotinas na linguagem *Python* para montagem das matrizes de rigidez e de massa de uma estrutura reticulada plana;
- Aplicar bibliotecas numéricas em *Python* de resolução de problema generalizado de autovalor para obtenção das frequências naturais e modos de vibração;
- Realizar exemplos de validação e aplicação que permitam analisar e explorar as potencialidades do código desenvolvido.

1.2 JUSTIFICATIVA

Considerando as dificuldades dos cálculos manuais em estruturas com múltiplos graus de liberdade, torna-se essencial o emprego de rotinas computacionais para a auxílio na obtenção

das soluções dos problemas matemáticos que a análise modal envolve. Tais ferramentas são fundamentais para obtenção das propriedades modais, como frequências naturais e modos de vibração de estruturas sujeitas a fenômenos como ressonância.

Este trabalho dedica-se ao desenvolvimento de um programa em *Python* para análise modal de estruturas reticuladas planas não amortecidas. A escolha da linguagem é motivada pela sua acessibilidade, flexibilidade, sintaxe relativamente simples e grande disponibilidade de bibliotecas para cálculo matricial e de problemas algébricos o que facilita a implementação da teoria envolvida para análise estruturais dinâmicas.

A implementação computacional por meio do código em linguagem de alto nível possibilita uma visualização mais sistemática e conceitual dos processos envolvidos, em contraposição ao uso de softwares comerciais que muitas vezes funcionam como uma “caixa preta” para o usuário. Além disso, a modelagem das estruturas pelo método da rigidez direta permite ilustrar de forma clara o poder da computação sobre a formulação matricial e os métodos de solução de autovalores e autovetores, aspectos fundamentais da análise modal.

Dessa forma, busca-se não apenas automatizar os cálculos estruturais, mas também aplicar a teoria de uma forma mais ativa, tendo em vista o robusto alicerce conceitual exigido para os estudos dinâmicos propostos no trabalho.

Ferramentas computacionais aplicadas à análise estrutural já foram exploradas em diversos trabalhos acadêmicos, que apontam resultados consistentes e próximos aos obtidos por softwares comerciais que possuem licenças onerosas. Por exemplo, Almeida e Fonsaca (2018) desenvolveram o código CERBERUS em *Python*, voltado para estruturas reticuladas espaciais, validando os resultados frente ao SAP2000 e confirmando a viabilidade didática da implementação. Nesse sentido, Soriano (2014) e Martha (2010) destacam a importância de rotinas próprias na consolidação do aprendizado de métodos matriciais e na compreensão dos processos de cálculo. Portanto, o programa proposto neste trabalho busca se inserir como uma ferramenta complementar, com foco específico em estruturas reticuladas planas e na análise modal, fornecendo resultados confiáveis ao mesmo tempo em que fortalece a compreensão conceitual da teoria.

1.3 METODOLOGIA

Na etapa inicial, realizou-se uma pesquisa bibliográfica sobre análise modal e métodos para análise estrutural. Consultou-se materiais como, artigos, monografias, dissertações, livros,

cursos e documentações da linguagem de programação, servindo como fontes para a construção da base conceitual do trabalho a ser desenvolvido.

A partir da pesquisa sobre os métodos de análise estrutural e suas aplicações vinculadas ao estudo de vibrações foi possível definir a abordagem a ser utilizada no desenvolvimento do programa, restringindo-se as considerações de estruturas planas, não amortecidas e sem esforços externos atuantes.

Dentre os métodos identificados na teoria, o Método da Rigidez Direta foi selecionado devido à sua formalização matricial adequada à implementação computacional. Conforme Martha (2010), essa versão do método dos deslocamentos tem uma montagem direta da matriz de rigidez global por simples soma de blocos de rigidez locais, dessa forma facilitando o tratamento de múltiplos graus de liberdade. Além disso, sua consistência com a estrutura do problema generalizado de autovalores simplifica a integração com a matriz de massa e os algoritmos numéricos de extração de modos de vibração.

Com base no referencial teórico e no entendimento das formulações matriciais necessárias à determinação dos parâmetros modais, foi desenvolvido um algoritmo na linguagem *Python*. Esse programa computacional automatiza as rotinas de cálculo, incluindo a construção das matrizes globais, aplicação de condições de contorno e solução do problema generalizado de autovalores para extração das frequências naturais e modos de vibração. O programa gera como saída um arquivo no formato *Microsoft Excel* (.xlsx) contendo as frequências naturais e modos de vibração correspondentes à estrutura analisada.

Por fim, procedeu-se à validação do programa por meio de estudos de caso, comparando-se diretamente as frequências naturais e as formas modais obtidas no código com as fornecidas pelo *Autodesk Robot Structural Analysis Professional*, adotado como referência externa. A comparação quantitativa é realizada por meio do erro relativo percentual entre frequências naturais, e a comparação qualitativa das formas modais é discutida considerando-se as diferenças de pós-processamento e de representação gráfica.

1.4 ORGANIZAÇÃO DO TRABALHO

Este trabalho está organizado em sete capítulos, além das referências bibliográficas e eventuais anexos e apêndices. No Capítulo 1 são apresentados a contextualização do tema, os objetivos gerais e específicos, a justificativa, a metodologia adotada e a presente descrição da organização do texto. O Capítulo 2 reúne a revisão bibliográfica, abordando os conceitos fundamentais de análise estrutural de sistemas reticulados, os princípios da dinâmica das

estruturas e os fundamentos teóricos da análise modal que servem de base para as formulações adotadas ao longo do trabalho.

O Capítulo 3 trata do método da rigidez direta aplicado a estruturas reticuladas planas, detalhando a discretização da estrutura, a definição dos graus de liberdade, a formulação das matrizes de rigidez dos elementos, a montagem da matriz global e a consideração das condições de contorno. No Capítulo 4 é apresentada a formulação da análise modal, com ênfase na construção da matriz de massa, na definição do problema generalizado de autovalor e nos procedimentos para obtenção das frequências naturais e modos de vibração do sistema. O Capítulo 5 descreve o programa computacional desenvolvido, discutindo a organização do código, o fluxo de informações, as rotinas de montagem das matrizes, os algoritmos numéricos empregados e a interface com o usuário. No Capítulo 6 são apresentados exemplos de validação e aplicação, nos quais os resultados do programa são comparados com soluções de referência e aplicados em estruturas representativas. Por fim, o Capítulo 7 apresenta as conclusões obtidas ao longo do trabalho e sugestões para desenvolvimentos futuros. Essa estrutura segue o cronograma definido para o projeto, de modo que a sequência de capítulos acompanha a evolução natural do estudo, desde a fundamentação teórica até a implementação computacional e a validação dos resultados.

2 REVISÃO BIBLIOGRÁFICA

Este capítulo apresenta os principais aspectos teóricos que fundamentam os métodos de análise empregados neste trabalho. Inicialmente, discorre-se sobre a análise estrutural e os métodos de resolução de estruturas, com ênfase no Método da Rigidez Direta, devido a sua eficácia na implementação de rotinas computacionais para resolução de estruturas reticuladas hiperestáticas.

Em seguida, aborda-se a análise modal, destacando-se sua relevância na determinação das propriedades dinâmicas das estruturas e apresentando um breve panorama da evolução teórica que fundamenta esse tipo de análise.

2.1 ANÁLISE ESTRUTURAL

Uma estrutura pode ser definida como um sistema de partes conectadas utilizadas para suportar ações. Quando se projeta uma estrutura que servirá a uma função específica, o engenheiro tem de se preocupar com a sua segurança, estética e funcionalidade, ao mesmo tempo tendo que levar em consideração as restrições econômicas e ambientais (Hibbeler, 2015).

Para a execução de uma estrutura de maneira que atenda a todas as necessidades para qual será construída é necessário o desenvolvimento de um projeto estrutural que satisfaça as condições de segurança, de utilização, econômicas, estéticas, ambientais, construtivas e legais. Um projeto estrutural deve ter como produto a especificação completa de uma estrutura, através da documentação dos estudos e instruções da engenharia, contemplando todas as informações necessárias para sua materialização (Martha, 2010).

A análise estrutural é uma das etapas fundamentais do projeto de engenharia, responsável por determinar os esforços internos, deslocamentos e reações de apoio decorrentes das ações atuantes em uma estrutura. É na análise estrutural que o engenheiro aplica o conhecimento das teorias físicas na linguagem matemática, resultado da formalização da ciência engenharia estrutural.

No contexto da análise moderna, Martha (2010) descreve quatro níveis de abstração presentes na Figura 1 que interligam o mundo físico ao modelo computacional: estrutura real, modelo estrutural, modelo discreto e modelo computacional. A estrutura real corresponde ao objeto no mundo físico, tal como é construída, sujeito às ações e restrições do meio físico.

Figura 1 – Quatro níveis de abstração de uma estrutura na análise estrutural



Fonte: Martha (2010)

O modelo estrutural é a idealização matemática que descreve o comportamento da estrutura, considerando hipóteses sobre o material, as condições de contorno e o regime de deformações. A criação do modelo estrutural é uma das tarefas mais importantes da análise estrutural e pode ser bastante complexa a depender do tipo de estrutura e sua importância. Para estruturas reticuladas, o modelo estrutural tem características bastantes específicas. A modelagem matemática desse tipo de estrutura está fundamentada na teoria de Navier, que rege o comportamento de elementos estruturais submetidos à flexão, efeitos axiais e de torção. Nesse tipo de estruturas, as vigas e colunas são representadas por linhas e sua informação tridimensional fica atrelada as propriedades globais das suas seções transversais.

Os métodos de análise representam o comportamento da estrutura por meio de um conjunto de parâmetros. O modelo discreto representa a idealização do modelo estrutural por um número finito de variáveis. O processo de passagem do modelo matemático analítico para a modelagem de elementos discretos com parâmetros finitos é denominado discretização.

O modelo computacional, último nível de abstração apresentado por Martha (2010) implementa rotinas de cálculos numéricos que permitem resolver os sistemas de equações para determinação dos conjuntos de parâmetros dos elementos discretos. Dessa forma gerando resultados aproximados ao do comportamento da estrutura real.

Entre os métodos clássicos de solução de estruturas destacam-se o Método das Forças e o Método dos Deslocamentos. No primeiro, as incógnitas são forças e momentos, que podem ser reações de apoio ou esforços internos. Uma vez que as incógnitas principais tenham sido determinadas, as forças reativas restantes da estrutura são obtidas satisfazendo as exigências de equilíbrio e compatibilidade.

O Método dos Deslocamentos tem como incógnitas principais os deslocamentos e rotações. Com a obtenção dos deslocamentos, as forças são determinadas substituindo-as incógnitas principais nas equações de equilíbrio para serem resolvidas. A vasta maioria dos softwares de computador modernos para análise estrutural de hoje em dia são desenvolvidos usando o Método dos deslocamentos devido à sua formulação matricial (Hibbeler, 2015).

O principal diferencial do método dos deslocamentos, em específico o da rigidez direta, é sua compatibilidade com a implementação computacional, uma vez que as operações envolvem apenas manipulação de matrizes e vetores. Soriano (2014) observa que a formulação matricial sistematiza o processo de análise e facilita sua automatização, permitindo que modelos complexos, antes resolvidos apenas por aproximações manuais, possam ser tratados com precisão e eficiência numérica.

Diversos autores e trabalhos acadêmicos evidenciam o potencial didático e prático de uma abordagem sistematizada e orientada à implementação computacional. Almeida e Fonsaca (2018) desenvolveram o código CERBERUS, em linguagem *Python*, para a análise de estruturas reticuladas planas e espaciais pelo Método da Rigidez Direta. Segundo os autores, o programa gera relatórios com as matrizes de rigidez, vetores de forças e resultados numéricos, apresentando boa correlação com softwares comerciais, como o *SAP2000*. Em linha com esse caráter educacional, Soriano (2014) destaca que o desenvolvimento de rotinas próprias constitui um recurso pedagógico relevante, pois contribui para a compreensão da lógica interna dos programas de cálculo estrutural.

Assim, o Método da Rigidez Direta consolida-se como uma ferramenta tanto para o aprendizado voltado a aplicação da teoria das estruturas, quanto para a prática profissional da engenharia estrutural. Sua robustez matemática, associada à facilidade de automatização pelo implemento com linguagens de programação, torna-o ponto de partida ideal para o desenvolvimento de algoritmos voltados à análise de sistemas estruturais hiperestáticos e, por extensão, à análise dinâmica de estruturas reticuladas.

2.2 ANÁLISE MODAL

A análise modal é uma ferramenta fundamental na engenharia estrutural moderna, permite a compreensão do comportamento dinâmico das estruturas pela determinação de suas frequências naturais e modos de vibração associados. Tais parâmetros são essenciais para avaliar o risco de ressonância e projetar sistemas estruturalmente seguros e eficientes sob a ação de carregamentos dinâmicos como vento, tráfego, impactos, multidões e terremotos.

Conforme explica Soriano (2014), as estruturas reais estão continuamente sujeitas a ações externas que variam no tempo e podem ser idealizadas como determinísticas, quando podem ser definidas analíticas ou numericamente ou aleatórias, quando não tem valores definidos para cada instante, sendo tratadas como probabilísticas. Tais ações induzem forças de inércia relevantes, caracterizando um comportamento dinâmico. Vibrações demasiadas podem

não apenas comprometer o conforto e a funcionalidade, mas também ocasionar fadiga e danos estruturais, especialmente em sistemas esbeltos e com grandes vãos, por conta da baixa capacidade de dissipação de energia.

O interesse pelas vibrações tem origens históricas. Rao (2011) relembra que a busca pela compreensão dos fenômenos vibratórios remonta à antiguidade, ligada inicialmente ao estudo da música evoluiu até o século XX, para um campo rigoroso da engenharia. Essa evolução foi construída a partir das contribuições de Galileu, que em 1638 descreveu experimentalmente o comportamento oscilatório dos corpos, passando por Newton, que formalizou as leis do movimento e forneceu os recursos matemáticos para a formulação das equações diferenciais que regem sistemas vibratórios.

Ao longo dos séculos XVIII e XIX, avanços decisivos foram introduzidos por Euler e Daniel Bernoulli, que desenvolveram as primeiras equações para análise de barras finas que ficou conhecida como teoria de vigas de Euler–Bernoulli, estabelecendo a base matemática para a modelagem estrutural de elementos flexíveis. Bernoulli também demonstrou que uma corda vibrante pode oscilar simultaneamente em vários de seus harmônicos, princípio hoje reconhecido como superposição. Segundo a linha histórica apresentada por Rao (2011), a teoria moderna das vibrações se consolida com Rayleigh, que desenvolveu métodos energéticos para a estimativa das frequências naturais de sistemas conservativos e, em trabalhos posteriores, com o avanço de formulações que incorporam a análise numérica e computacional, bem como o estudo de sistemas não lineares e de problemas de vibração aleatória, ampliando o escopo de aplicação da teoria das vibrações na engenharia.

A preocupação prática com vibrações estruturais ganhou destaque após desastres emblemáticos, como o colapso da ponte Tacoma Narrows (1940), mostrado na Figura 2, o que consolidou a análise modal como um requisito no dimensionamento de estruturas suscetíveis a excitações dinâmicas.

Segundo Chopra (2014), a análise modal constitui o primeiro passo para solução de problemas dinâmicos de sistemas com múltiplos graus de liberdade, pois permite decompor o sistema em um conjunto de modos de vibração que podem ser tratados de forma independente. Essa decomposição modal simplifica a resolução das equações diferenciais de movimento, que passam a ser formuladas como um problema de autovalores e autovetores, respectivamente associados às frequências naturais e aos modos de vibração da estrutura.

Figura 2 – Colapso da ponte Tacoma Narrows em 1940



Fonte: PRELINGER ARCHIVES. Tacoma Narrows Bridge collapse

As equações de movimento de um sistema estrutural discreto com múltiplos graus de liberdade podem ser escritas, em notação matricial, na forma:

$$[M]\{\ddot{u}(t)\} + [C]\{\dot{u}(t)\} + [K]\{u(t)\} = \{f(t)\} \quad (1)$$

em que $[M]$ é a matriz global de massa, $[C]$ é a matriz de amortecimento, $[K]$ é a matriz global de rigidez, $\{u(t)\}$ é o vetor de deslocamentos nodais em função do tempo e $\{f(t)\}$ é o vetor de forças externas.

Na análise modal básica, que constitui o escopo deste trabalho, considera-se o caso de vibração livre não amortecida, onde não há carregamentos externos e o amortecimento é desprezado. Nessas condições, a equação de movimento reduz-se a:

$$[M]\{\ddot{u}(t)\} + [K]\{u(t)\} = \{0\} \quad (2)$$

Seguindo a formulação apresentada por Soriano, admite-se que a resposta do sistema, após uma perturbação inicial, possa ser descrita por uma solução harmônica de deslocamentos nodais da forma:

$$\{u(t)\} = \{\hat{\phi}_j\} \cos(\omega_j t - \varphi_j) \quad (3)$$

em que $\{\hat{\phi}_j\}$ é o j -ésimo modo natural de vibração, ω_j é a correspondente frequência natural de vibração livre e φ_j é o ângulo de fase associado. Essa forma de solução é compatível com a teoria de equações diferenciais lineares com coeficientes constantes, segundo a qual, para sistemas conservativos, as soluções da equação homogênea são periódicas e podem ser escritas como combinações de funções harmônicas no tempo.

Ao substituir a solução harmônica dada pela Equação (3) na equação de vibração livre não amortecida da Equação (2), obtém-se um sistema de equações algébricas homogêneas na forma:

$$(-[M]\omega_j^2 + [K])\{\hat{\phi}_j\} \cos(\omega_j t - \varphi_j) = \{0\} \quad (4)$$

Como o fator $\cos(\omega_j t - \varphi_j)$ não é identicamente nulo no tempo, a igualdade obtida após a substituição da solução harmônica na equação de movimento só pode ser satisfeita, de forma não trivial, se o vetor de deslocamentos modais $\{\hat{\phi}_j\}$ atender à relação:

$$([K] - \omega_j^2[M])\{\hat{\phi}_j\} = \{0\} \quad (5)$$

que corresponde ao problema generalizado de autovalores da estrutura. Essa expressão é normalmente escrita na seguinte forma:

$$[K]\{\hat{\phi}_j\} = \omega_j^2[M]\{\hat{\phi}_j\} \quad (6)$$

em que ω_j^2 é o autovalor e $\{\hat{\phi}_j\}$ é o autovetor associado. Cada par $(\omega_j^2, \{\hat{\phi}_j\})$ descreve um modo natural de vibração: o autovalor define a frequência natural daquele modo e o autovetor fornece o padrão relativo de deslocamentos nodais que caracteriza a forma modal correspondente.

O problema generalizado de autovalores pode ser interpretado como a busca de configurações de deslocamento para as quais as forças de restituição elástica, representadas por $[K]\{\hat{\phi}_j\}$, e as forças de inércia, representadas por $[M]\{\hat{\phi}_j\}$, mantêm uma relação estritamente proporcional. Com essas condições, a estrutura vibra segundo um padrão espacial fixo, apenas com variação periódica de sinal no tempo, o que caracteriza a vibração em modo natural.

A solução do problema $[K]\{\hat{\phi}_j\} = \omega_j^2[M]\{\hat{\phi}_j\}$ fornece, o conjunto de frequências naturais e modos de vibração que descreve o comportamento dinâmico fundamental da

estrutura. Esse é objetivo principal do programa desenvolvido neste trabalho, que, a partir das matrizes globais de rigidez e de massa obtidas pelo Método da Rigidez Direta, determina numericamente os autovalores e autovetores associados às estruturas reticuladas planas. No Capítulo 4, essa formulação é retomada no contexto das matrizes globais $[K]$ e $[M]$ obtidas pelo Método da Rigidez Direta e da implementação computacional desenvolvida.

Conforme apresentado por Rao (2011), qualquer resposta dinâmica linear de um sistema com múltiplos graus de liberdade pode ser representada como uma combinação linear dos seus modos naturais, conceito conhecido como superposição modal. Essa possibilidade decorre do fato de que os autovetores associados às diferentes frequências naturais apresentam propriedades de ortogonalidade em relação às matrizes de massa e de rigidez, o que permite desacoplar o sistema original de equações diferenciais em um conjunto de equações escalares independentes, uma para cada modo de vibração.

Além do seu papel teórico, a análise modal possui aplicações práticas diretas em engenharia civil e mecânica, como na calibração de modelos numéricos, no controle de vibrações, em análises sísmicas baseadas em espectros de resposta e em procedimentos de identificação de danos estruturais. Chopra (2014) destaca que a decomposição modal é a base dos métodos de superposição utilizados em análises sísmicas lineares, nas quais a resposta global da estrutura é obtida a partir da combinação adequada das respostas modais associadas a cada frequência natural considerada.

Do ponto de vista computacional, essa metodologia se integra de forma natural à formulação matricial do Método da Rigidez Direta, empregado neste trabalho. As matrizes $[K]$ e $[M]$ são montadas a partir da contribuição de cada elemento de barra do modelo estrutural, com uso de sistemas de coordenadas locais e globais, conforme a abordagem apresentada por Martha (2010) para estruturas reticuladas. Após a aplicação das condições de contorno, obtém-se o sistema global de equações que, no contexto estático, assume a forma $[K]\{U\} = \{F\}$ e, no contexto dinâmico, conduz ao problema de autovalores $[K]\{\hat{\phi}_j\} = \omega_j^2[M]\{\hat{\phi}_j\}$ cuja formulação matemática será detalhada no Capítulo 4 e cuja implementação computacional é apresentada no Capítulo 5.

A metodologia para análise modal adotada neste trabalho pode ser descrita pelas seguintes etapas principais:

- modelagem e idealização da estrutura, com definição dos graus de liberdade e das propriedades geométricas e mecânicas dos elementos;

-
- montagem das matrizes globais de rigidez e de massa, que traduzem, respectivamente, as propriedades elásticas e inerciais do sistema;
 - resolução do problema generalizado de autovalores, visando à obtenção das frequências naturais e das formas modais associadas;
 - organização e apresentação dos resultados modais, com listagem das frequências naturais obtidas e visualização das formas de vibração correspondentes.

Essas etapas são implementadas de forma sistemática no algoritmo desenvolvido neste trabalho. A rotina computacional, apresentada no Capítulo 5, constrói as matrizes $[K]$ e $[M]$ para estruturas reticuladas planas não amortecidas, aplica as restrições de apoio, resolve numericamente o problema de autovalores e organiza como saída as frequências naturais e as formas modais, que são apresentadas ao usuário por meio de tabelas dos deslocamentos nodais.

3 MÉTODO DA RIGIDEZ DIRETA

3.1 CONCEITOS FUNDAMENTAIS

O método da rigidez direta é uma formulação matricial do método dos deslocamentos em que as incógnitas do problema estrutural são os deslocamentos e rotações dos nós. Nessa abordagem, as condições de equilíbrio são impostas em cada nó da estrutura, de modo que as forças internas desenvolvidas nos elementos sejam compatíveis com os deslocamentos nodais considerados. A solução do problema resulta em um sistema linear na forma geral:

$$[K]\{U\} = \{F\} \quad (7)$$

em que $[K]$ é a matriz de rigidez global da estrutura, $\{U\}$ é o vetor de deslocamentos e rotações nodais e $\{F\}$ é o vetor de forças nodais externas.

As estruturas analisadas neste trabalho são reticuladas, ou seja, modelos compostos por elementos lineares retos, com uma dimensão predominante em relação às outras duas. Cada elemento é idealizado como uma barra que liga dois nós, e o conjunto de barras e nós define o modelo estrutural. Tais estruturas incluem vigas, treliças planas, treliças espaciais, pórticos planos, grelhas e pórticos espaciais.

Especificamente neste trabalho, considera-se o caso de estruturas reticuladas planas, situadas no plano XY . O pórtico plano é adotado como modelo mais geral, com três graus de liberdade por nó, correspondentes às translações u_x e u_y e à rotação em torno do eixo perpendicular ao plano (θ_z). Treliças planas podem ser interpretadas como casos particulares em que as rotações nodais não são tomadas como graus de liberdade e os elementos trabalham predominantemente à força axial.

A passagem da estrutura real para o modelo utilizado no método da rigidez direta envolve diferentes níveis de abstração. A estrutura real é idealizada em um modelo estrutural, do qual se obtém um modelo discreto em termos de deslocamentos nodais. Sobre esse modelo constrói-se o modelo computacional, no qual o método da rigidez direta é formulado em termos matriciais.

Na formulação em deslocamentos, as quantidades discretas que representam a configuração deformada da estrutura são os deslocamentos e rotações dos nós, denominados

deslocabilidades. Essas grandezas definem o comportamento cinemático do modelo, já que a deformada contínua dos elementos pode ser obtida por interpolação a partir dos valores nodais. No contexto do método da rigidez direta, é usual chamar essas componentes de graus de liberdade, termo que será adotado neste trabalho tanto para componentes livres como para aquelas restritas por apoios.

O comportamento de cada elemento de barra é descrito por uma relação matriz entre esforços e deslocamentos nos seus nós extremos. Essa matriz de rigidez local é inicialmente escrita em um sistema de coordenadas associado ao elemento e, posteriormente, transformada para o sistema de coordenadas global da estrutura por meio de uma matriz de rotação. A matriz de rigidez global resulta da superposição das contribuições de todos os elementos, obedecendo à regra de correspondência entre os graus de liberdade locais e globais.

Neste trabalho, adotam-se as hipóteses usuais da análise linear, estática e elástica. Admite-se comportamento geométrico e fisicamente linear, com pequenas deformações e regime elástico linear do material, além de ligações perfeitas entre os elementos. Com essas hipóteses, a matriz de rigidez é simétrica e definida positiva, e a Equação (7) possui solução única para um conjunto especificado de condições de contorno. Essa formulação estática constitui a base sobre a qual, nos capítulos seguintes, será acoplada a matriz de massa para a formulação do problema modal generalizado, cuja solução resultará nos modos e frequências naturais de vibração.

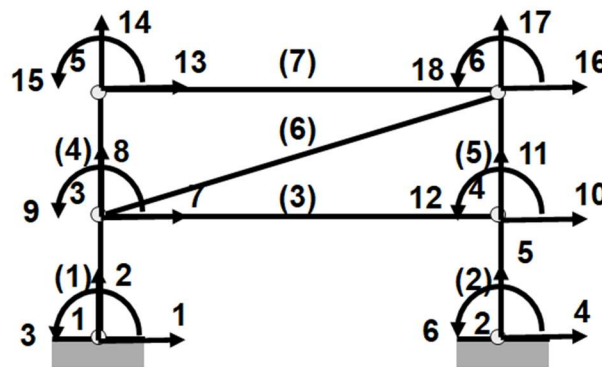
A aplicação do método da rigidez direta a estruturas reticuladas planas segue uma sequência bem definida: discretização da estrutura em nós e elementos, definição do vetor de forças nodais equivalentes, escolha dos sistemas de coordenadas local e global, determinação das matrizes de rigidez dos elementos, montagem da matriz de rigidez global e imposição das condições de contorno. As seções seguintes detalham cada uma dessas etapas para o caso de estruturas planas de barras.

3.2 DISCRETIZAÇÃO DA ESTRUTURA

A discretização da estrutura é a etapa em que se define o conjunto finito de nós, elementos de barra e graus de liberdade que irão representar o comportamento da estrutura real. Nessa etapa, a estrutura é decomposta em elementos mais simples, cujos extremos são identificados como nós, e são atribuídas propriedades geométricas e mecânicas a cada elemento. A partir desse modelo discreto é que se torna possível formular, em termos matriciais, as relações entre esforços e deslocamentos nodais.

O ponto de partida é a definição de um sistema de coordenadas global plano, adotado no escopo deste trabalho como um sistema cartesiano ortogonal XY . A partir desse sistema, procede-se à numeração dos elementos de barras e nós e ao registro de suas coordenadas globais (X_i, Y_i) , conforme ilustrado na Figura 3. Essa numeração deve ser feita de forma consistente, uma vez que será utilizada tanto na montagem das matrizes globais quanto na implementação computacional.

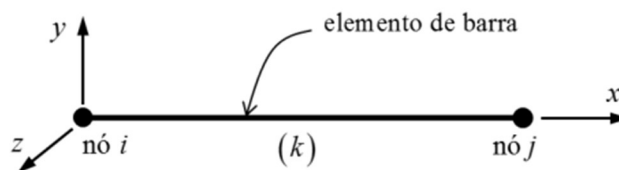
Figura 3 – Sistema de coordenadas global de estrutura e numeração dos elementos e nós



Fonte: Autor (2025)

Cada barra do modelo estrutural é substituída por um elemento de barra que liga dois nós, denominados nó inicial i e nó final j . Os elementos de barra k em um sistema de coordenadas local, conforme Figura 4, são definidos de modo que o comprimento e a inclinação do elemento reproduzam a geometria da barra real no plano, o que permite calcular o comprimento L e os cossenos diretores que caracterizam a orientação do eixo local do elemento em relação ao sistema global. Esses parâmetros geométricos serão empregados, nas seções seguintes, na construção da matriz de transformação entre os sistemas de coordenadas local e global.

Figura 4 – Sistema de coordenadas local de um elemento k



Fonte: Soares (2017)

A cada elemento são associados os parâmetros necessários para a definição de sua rigidez e de sua inércia de massa. Do ponto de vista geométrico, consideram-se, em geral, seções transversais prismáticas, caracterizadas por área A e momento de inércia I em relação ao eixo neutro. Do ponto de vista do material, são especificados o módulo de elasticidade longitudinal E e a densidade volumétrica ρ . Os parâmetros E , A e I são utilizados na formação da matriz de rigidez dos elementos de pórtico plano, enquanto ρ é empregado, em conjunto com A e L , na construção das matrizes de massa discutidas nos capítulos seguintes.

Os graus de liberdade do modelo são definidos nos nós da estrutura. Para o pórtico plano, adotado neste trabalho como caso mais geral de estrutura reticulada plana, considera-se em cada nó dois deslocamentos de translação, u_x e u_y , e uma rotação θ_z em torno do eixo perpendicular ao plano. Assim, cada nó possui três graus de liberdade e cada elemento de pórtico plano, conectando dois nós, possui seis graus de liberdade nodais. Treliças planas são interpretadas como casos particulares desse modelo geral, obtidos pela eliminação das rotações nodais como incógnitas e pela consideração de que os elementos resistem apenas a esforços normais, de modo que cada nó possui apenas dois graus de liberdade de translação.

Na implementação computacional, torna-se conveniente numerar os graus de liberdade globais em função da numeração dos nós. Para estruturas de pórtico plano, uma convenção simples é associar ao nó n os graus de liberdade globais:

- $3n - 2$ (translação em X),
- $3n - 1$ (translação em Y)
- $3n$ (rotação em torno de Z).

Essa regra estabelece a correspondência entre os graus de liberdade locais de cada elemento e as posições das entradas na matriz de rigidez global, facilitando a etapa de montagem que será tratada adiante.

Embora a formulação do método da rigidez direta seja conduzida em termos de deslocamentos globais, cada elemento é inicialmente descrito em seu sistema de coordenadas local. Nesse sistema, os deslocamentos nodais são reunidos em um vetor de deslocamentos de extremidade, que contém as componentes de translação e rotação dos nós i e j ao longo e em torno dos eixos locais do elemento. A relação entre esse vetor de deslocamentos locais e o vetor de deslocamentos globais é estabelecida por uma matriz de transformação, função dos cossenos diretores do elemento, que será apresentada na Seção 3.4. Essa distinção entre grandezas locais e globais é fundamental para que se possa, a partir da rigidez formulada no sistema local, obter as contribuições de cada elemento na matriz de rigidez global da estrutura.

Cabe observar que, em formulações gerais do método da rigidez direta, a discretização também envolve a forma de representar os carregamentos externos. Em termos conceituais, ações distribuídas ou aplicadas ao longo dos elementos podem ser substituídas por forças e momentos nodais equivalentes, compatíveis com os graus de liberdade definidos para cada nó, porém neste trabalho, o foco recai sobre a análise modal de estruturas reticuladas planas em vibração livre, para a qual se considera a ausência de ações externas durante a etapa de solução. Assim, a modelagem de carregamentos é fica limitada, quando necessário, à definição de forças e momentos concentrados nos nós, organizados no vetor global de forças. O procedimento de montagem desse vetor é apresentado de forma resumida na Seção 3.3, apenas para elucidar formulação estática do método da rigidez direta e viabilizar o uso do programa caso sejam feitas análises lineares sob carregamentos nodais.

Concluídas essas etapas, fica definido o modelo discreto sobre o qual será aplicado o método da rigidez direta: um conjunto de nós com coordenadas conhecidas no sistema global, um conjunto de elementos de barra com propriedades geométricas e de material associadas, além de uma correspondência bem estabelecida entre os graus de liberdade locais de cada elemento e os graus de liberdade globais da estrutura. Quando necessário, esse modelo pode ainda incorporar a representação nodal de forças e momentos externos, organizados no vetor global de carregamentos. Esse conjunto de informações constitui a base de dados utilizada no Capítulo 5 para a montagem das matrizes globais de rigidez e de massa e para a formulação e solução do problema generalizado de autovalores associado às estruturas reticuladas planas analisadas neste trabalho.

3.3 VETOR DE FORÇAS

Na Equação (7), o vetor $\{F\}$ reúne as forças generalizadas associadas aos graus de liberdade da estrutura. Cada componente de deslocamento ou rotação nodal possui uma força conjugada correspondente: para as translações u_x e u_y , as forças nodais F_x e F_y , e, para a rotação θ_z , o momento nodal M_z . Assim, em um pórtico plano com n nós, o vetor global de forças é formado por $3n$ componentes, organizadas de maneira compatível com a metodologia de numeração dos graus de liberdade adotada na Seção 3.2.

No modelo de pórtico plano utilizado neste trabalho, o carregamento externo é representado exclusivamente por forças e momentos aplicados nos nós da estrutura. Cada nó pode receber componentes F_x , F_y e M_z , que são posicionadas em $\{F\}$ de acordo com o número

do nó e com a regra de numeração dos graus de liberdade globais. Para o nó n , por exemplo, as componentes F_x , F_y e M_z são associadas, respectivamente, aos graus de liberdade globais $3n - 2$, $3n - 1$ e $3n$. No caso particular de treliças planas, em que as rotações nodais não são consideradas incógnitas, o vetor de forças passa a conter apenas as componentes de força nas direções x e y globais, associadas aos dois graus de liberdade de translação de cada nó.

Na implementação computacional desenvolvida, o vetor global de forças nodais é montado a partir dos carregamentos especificados na tabela de nós. Para cada nó, o programa lê as colunas de forças nas direções X e Y e, quando existentes, os momentos nodais, acumulando essas contribuições nas posições correspondentes do vetor $\{F\}$. Não são considerados, nesta etapa, carregamentos distribuídos ao longo dos elementos de barra nem forças equivalentes obtidas a partir de soluções de engastamento perfeito, pois tais casos extrapolam o escopo do trabalho e não são necessários para a análise modal proposta.

Para este trabalho, o papel do vetor de forças é principalmente completar a formulação estática do método da rigidez direta e permitir que o programa realize, quando desejado, análises estáticas lineares sob carregamentos nodais. Nas análises de vibrações livres, que constituem o foco principal do trabalho, considera-se a ausência de ações externas, de modo que o vetor de forças é tido como nulo e a equação de equilíbrio estático é substituída pelo problema de autovalor generalizado em termos das matrizes de rigidez e de massa.

3.4 SISTEMA DE COORDENADAS

A distinção entre sistemas de coordenadas global e local é essencial para o método da rigidez direta. As equações de equilíbrio da estrutura, como a equação matricial apresentada na Seção 3.1, são formuladas no sistema global, enquanto as relações constitutivas dos elementos de barra (matriz de rigidez local, matriz de massa local) são mais simples quando escritas em sistemas locais associados a cada elemento. A montagem da matriz de rigidez global depende justamente da capacidade de converter, de forma consistente, deslocamentos e esforços entre esses dois sistemas de referência.

No escopo deste trabalho, a estrutura está contida em um plano XY e o sistema de coordenadas global adotado é cartesiano ortogonal, com eixos X e Y definidos conforme a discretização apresentada na Seção 3.2. Os nós da estrutura são identificados por um índice inteiro e têm suas posições descritas pelas coordenadas globais (X_i, Y_i) , que são utilizadas tanto na montagem das matrizes globais quanto na implementação computacional.

A cada elemento de barra que liga um nó inicial i a um nó final j associa-se um sistema de coordenadas local. Esse sistema é definido de modo que o eixo local x coincida com o eixo longitudinal do elemento, orientado do nó i para o nó j . O eixo local y é tomado no plano da estrutura, ortogonal ao eixo x , e o eixo local z é perpendicular ao plano XY , coincidindo com o eixo global Z . Com essa escolha, o elemento de pórtico plano possui, em cada nó, dois deslocamentos de translação ao longo dos eixos locais x e y e uma rotação em torno de z , compatíveis com os graus de liberdade globais definidos na Seção 3.2.

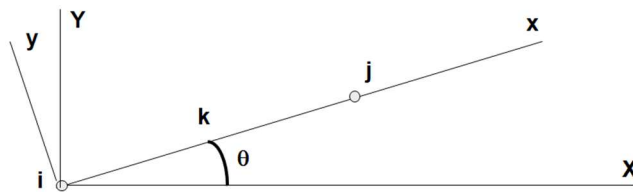
A orientação do sistema local em relação ao sistema global é determinada por um único ângulo θ , medido do eixo global X para o eixo local x , no sentido anti-horário, conforme Figura 5. A partir das coordenadas dos nós i e j no sistema global, esse ângulo é determinado por meio dos cossenos diretores do eixo local x , pelos seguintes parâmetros:

$$L = \sqrt{(X_j - X_i)^2 + (Y_j - Y_i)^2} \quad (8)$$

$$c = \cos \theta = \frac{X_j - X_i}{L} \quad (9)$$

$$s = \sin \theta = \frac{Y_j - Y_i}{L} \quad (10)$$

Figura 5 – Orientação de um elemento k em relação ao sistema global



Fonte: Autor (2025)

Esses parâmetros foram introduzidos na Seção 3.2, na descrição da geometria dos elementos em coordenadas globais, e serão utilizados de forma sistemática na construção da matriz de transformação entre sistemas de coordenadas local e global.

Para um elemento genérico de pórtico plano, o vetor de deslocamentos nodais em coordenadas globais pode ser escrito, por exemplo, na forma:

$$\{u^g\} = \begin{Bmatrix} u_{iX} \\ u_{iY} \\ \theta_i \\ u_{jX} \\ u_{jY} \\ \theta_j \end{Bmatrix} \quad (11)$$

em que u_{iX} e u_{iY} são as translações do nó i nas direções globais X e Y , e θ_i é a rotação do nó i em torno do eixo Z que sai do plano; analogamente tem-se u_{jX} , u_{jY} e θ_j para o nó j . Em coordenadas locais, o mesmo estado de deslocamentos é representado por:

$$\{u^l\} = \begin{Bmatrix} u_{ix} \\ u_{iy} \\ \theta_i \\ u_{jx} \\ u_{jy} \\ \theta_j \end{Bmatrix} \quad (12)$$

em que u_{ix} e u_{iy} são as translações do nó i nas direções dos eixos locais x e y , e assim por diante. Em estruturas estritamente planas, a rotação em torno do eixo Z é a mesma quando medida nos sistemas local e global, o que justifica a presença direta de θ_i e θ_j em ambos os vetores.

A relação entre deslocamentos locais e globais é linear e pode ser escrita por meio de uma matriz de transformação $[T]$, função dos cossenos diretores do elemento:

$$\{u^l\} = [T]\{u^g\} \quad (13)$$

No caso de um elemento de pórtico plano, a matriz $[T]$ assume a forma:

$$[T] = \begin{bmatrix} c & s & 0 & 0 & 0 & 0 \\ -s & c & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & c & s & 0 \\ 0 & 0 & 0 & -s & c & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (14)$$

em que $c = \cos \theta$ e $s = \sin \theta$. As duas primeiras linhas realizam a rotação das componentes de translação do nó i , preservando sua rotação θ_i . As linhas 4 e 5 fazem o mesmo para o nó j ,

preservando θ_j . Essa estrutura em blocos repete, em dimensão 3×3 , a transformação clássica de vetores planares entre sistemas de coordenadas rotacionados.

A matriz T é ortogonal, ou seja, satisfaz $[T]^T [T] = [I]$. Nesta seção, essa propriedade é utilizada para garantir que a mudança de base entre sistemas de coordenadas local e global não altera o conteúdo físico das grandezas, permitindo empregar a mesma matriz de transformação tanto sobre deslocamentos quanto sobre esforços nodais, quando necessário.

De forma análoga ao que foi feito para os deslocamentos, vetores de forças e momentos nas extremidades do elemento podem ser representados em coordenadas locais ou globais e convertidos entre si por meio da matriz T . Assim, quando se conhece o estado de esforços em um sistema de referência, torna-se possível obtê-lo no outro sistema de forma consistente com a cinemática adotada, sem que seja necessário detalhar aqui a transformação componente a componente.

Tendo essas definições, a mudança de base entre sistemas local e global para as relações constitutivas dos elementos torna-se direta. Se a matriz de rigidez de um elemento é inicialmente formulada no sistema local, na forma:

$$[k^l]\{u^l\} = \{f^l\} \quad (15)$$

Então, ao se substituir nas relações constitutivas locais as expressões de transformação de deslocamentos e esforços, obtém-se, em coordenadas globais:

$$[k^g]\{u^g\} = \{f^g\} \quad (16)$$

em que

$$[k^g] = [T]^T [k^l] [T] \quad (17)$$

Uma expressão análoga é empregada para transformar as matrizes de massa locais para o sistema global, assegurando que rigidez e inércia sejam tratadas de forma consistente na formulação do problema modal generalizado.

No caso particular de treliças planas, em que os elementos trabalham predominantemente à força axial e as rotações nodais não são tratadas como incógnitas, a mesma estrutura de transformação se aplica a vetores e matrizes de dimensão reduzida. Cada

nó passa a contribuir apenas com duas translações, de modo que os vetores $\{u^g\}$ e $\{u^l\}$ possuem quatro componentes por elemento, e a matriz $[T]$ é reduzida para uma transformação bidimensional das translações, construída a partir dos mesmos coeficientes $\cos \theta$ e $\sin \theta$. A formulação do pórtico plano, portanto, é mantida como caso mais geral, e as treliças planas são obtidas pela redução do número de graus de liberdade considerados.

Do ponto de vista computacional, a distinção entre sistemas de coordenadas e a definição explícita da matriz de transformação $[T]$ permitem desacoplar duas tarefas: de um lado, a dedução de matrizes de rigidez e de massa em coordenadas locais, diretamente a partir das soluções básicas de barras de pórtico plano; de outro, a montagem das matrizes globais da estrutura, formuladas em termos dos graus de liberdade globais definidos na discretização da estrutura. Esta seção estabelece, portanto, o elo geométrico entre o modelo de elemento e o modelo global da estrutura, que será utilizado de forma recorrente na implementação computacional.

3.5 MATRIZ DE RIGIDEZ

Conforme apresentado na Seção 3.1, a aplicação do Método da Rigidez Direta a uma estrutura reticulada plana leva, em nível global, ao sistema linear da Equação (7). Em que $[K]$ é a matriz de rigidez global da estrutura, $\{U\}$ é o vetor de deslocamentos e rotações nodais e $\{F\}$ é o vetor de forças nodais externas. A construção de $[K]$ é feita a partir da contribuição de cada elemento de barra, cuja rigidez é inicialmente formulada em coordenadas locais e, em seguida, transformada para o sistema de coordenadas global, conforme a metodologia usual em estruturas de barras apresentada por Martha (2010) para o Método da Rigidez Direta.

A rigidez de cada elemento é obtida a partir de soluções fundamentais da barra isolada, em que se impõem deslocamentos unitários em cada grau de liberdade de extremidade, mantendo-se os demais deslocamentos nulos. As forças de reação correspondentes, determinadas em regime elástico linear, definem os coeficientes de rigidez do elemento, que podem ser organizados na forma matricial. Essa interpretação corresponde a uma generalização da lei de Hooke para sistemas com múltiplos graus de liberdade. Cada grau de liberdade está associado a uma equação de equilíbrio, e o acoplamento entre esses graus é descrito pelos coeficientes da matriz de rigidez.

No caso do pórtico plano, adotado como modelo mais geral de estrutura reticulada plana, cada elemento liga dois nós com três graus de liberdade por nó: translações u_x e u_y e rotação

θ_z . Assim, o elemento possui seis graus de liberdade nodais e é conveniente reunir as incógnitas de extremidades do elemento em um vetor de deslocamentos nodais no sistema local:

$$\{u_e^l\} = \begin{Bmatrix} u_{ix} \\ u_{iy} \\ \theta_i \\ u_{jx} \\ u_{jy} \\ \theta_j \end{Bmatrix} \quad (18)$$

Para os esforços correspondentes em um vetor de forças nodais locais:

$$\{f_e^l\} = \begin{Bmatrix} N_i \\ V_i \\ M_i \\ N_j \\ V_j \\ M_j \end{Bmatrix} \quad (19)$$

em que N , V e M representam, respectivamente, o esforço normal, o esforço cortante e o momento fletor nas extremidades do elemento.

A relação linear entre esforços e deslocamentos nas direções locais, que representa o sistema de equilíbrio do elemento no seu próprio sistema de coordenadas, é escrita na forma:

$$[k_e^l]\{u_e^l\} = \{f_e^l\} \quad (20)$$

A matriz $[k_e^l]$ é denominada matriz de rigidez local do elemento e depende apenas das propriedades geométricas e mecânicas da barra e do seu comprimento. A partir dessa formulação elementar, pode-se prosseguir para determinação explícita dos coeficientes de rigidez e da transformação da matriz local para o sistema global, etapa necessária para a montagem de $[K]$ pela regra de correspondência dos graus de liberdade.

3.5.1 Coeficientes de rigidez

Os coeficientes de rigidez de um elemento são definidos a partir da resposta da barra engastada em ambas as extremidades quando se impõe, isoladamente, um deslocamento unitário em um dos seus graus de liberdade nodais, mantendo os demais iguais a zero. O esforço

de reação que surge em um determinado grau de liberdade i devido a um deslocamento unitário aplicado no grau de liberdade j é o coeficiente k_{ij} da matriz de rigidez. Organizando esses coeficientes em uma matriz quadrada, obtém-se a matriz $[k_e^l]$ para o elemento.

Para um elemento de pórtico plano prismático, de comprimento L , área da seção transversal A , momento de inércia I em relação ao eixo z perpendicular ao plano da estrutura e módulo de elasticidade longitudinal E , a matriz de rigidez local no sistema de coordenadas do elemento pode ser escrita, na ordenação $(u_{ix}, u_{iy}, \theta_i, u_{jx}, u_{jy}, \theta_j)$, como (Soriano, 2014):

$$[k_e^l] = \begin{bmatrix} \frac{EA}{L} & 0 & 0 & -\frac{EA}{L} & 0 & 0 \\ \cdot & \frac{12EI_z}{L^3} & \frac{6EI_z}{L^2} & 0 & -\frac{12EI_z}{L^3} & \frac{6EI_z}{L^2} \\ \cdot & \cdot & \frac{4EI_z}{L} & 0 & -\frac{6EI_z}{L^2} & \frac{2EI_z}{L} \\ \cdot & \cdot & \cdot & \frac{EA}{L} & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \frac{12EI_z}{L^3} & -\frac{6EI_z}{L^2} \\ sim. & \cdot & \cdot & \cdot & \cdot & \frac{4EI_z}{L} \end{bmatrix} \quad (21)$$

Essa matriz resulta da superposição de duas contribuições independentes: uma associada à deformação axial, proporcional a $\frac{EA}{L}$, e outra associada à flexão no plano, proporcional a $\frac{EI}{L^3}$. A presença de termos fora da diagonal principal traduz o acoplamento entre deslocamentos verticais e rotações nas extremidades, efeito resultante do comportamento à flexão de vigas engastadas.

Para treliças planas, os elementos resistem apenas a esforços normais e as rotações nodais não são consideradas como graus de liberdade. Nessas condições, o elemento passa a ter apenas dois deslocamentos úteis em cada nó no sistema local, associados à direção axial. Mantida a ordenação $(u_{ix}, u_{iy}, u_{jx}, u_{jy})$, a matriz de rigidez local pode ser escrita, para um elemento alinhado com o eixo local x , como (Soriano, 2014):

$$[k_e^l]_{treliça} = \frac{EA}{L} \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (22)$$

Pode-se observar que apenas os termos associados à deformação axial são diferentes de zero, enquanto os graus de liberdade transversais têm rigidez nula.

3.5.2 Transformação de coordenadas

A matriz de rigidez local $[k_e^l]$ e os coeficientes apresentados na Subseção 3.5.1 são definidos em um sistema de coordenadas associado ao elemento, no qual o eixo x coincide com o eixo da barra e o eixo y é ortogonal a esse eixo no plano da estrutura. Nesse sistema, os graus de liberdade de extremidade estão diretamente relacionados às deformações axial e à flexão no plano do pórtico, o que simplifica a dedução das expressões analíticas dos termos da matriz de rigidez a partir das soluções fundamentais de barras engastadas em ambas as extremidades. Para que as contribuições de todos os elementos possam ser reunidas em uma única matriz global $[K]$, torna-se necessário representar essas relações esforço–deslocamento também no sistema de coordenadas global XY adotado para o modelo estrutural.

A mudança de base referencial é feita por meio da matriz de transformação definida na Seção 3.4, que relaciona os vetores de deslocamentos nodais locais e globais do elemento conforme a Equação (13), construída a partir dos cossenos diretores do eixo longitudinal da barra.

A mesma matriz de transformação é utilizada para compatibilizar as forças nodais de extremidade com o sistema global. Dessa forma, o equilíbrio interno do elemento, representado em coordenadas locais pela Equação (20), pode ser expresso em termos dos graus de liberdade globais substituindo-se nas relações constitutivas as transformações de deslocamentos e de esforços. O resultado desse procedimento é a expressão geral de transformação da matriz de rigidez local para o sistema global apresentada na Equação (17), que fornece a matriz de rigidez do elemento escrita diretamente nas coordenadas globais da estrutura.

No caso das treliças planas consideradas neste trabalho, a transformação de coordenadas é aplicada de forma análoga, restringindo-se às componentes de translação dos nós e utilizando os mesmos cossenos diretores definidos na Seção 3.4 para converter a matriz de rigidez axial local para o sistema de coordenadas global.

A etapa de transformação é realizada elemento a elemento antes da montagem da matriz de rigidez global. Para cada barra, o programa computacional utiliza a matriz de rigidez local, formulada no sistema de coordenadas do elemento, e a matriz de transformação geométrica associada à sua orientação para obter a matriz de rigidez em coordenadas globais. As matrizes

transformadas são então utilizadas na Subseção 3.5.3, em conjunto com a regra de correspondência entre graus de liberdade, para a construção da matriz de rigidez global.

3.5.3 Montagem da matriz de rigidez global

Uma vez determinadas, para cada elemento de barra, a matriz de rigidez no sistema local $[k_e^l]$ e sua correspondente no sistema global $[k_e^g]$, obtida pela transformação de coordenadas apresentada na Subseção 3.5.2, o passo seguinte do Método da Rigidez Direta consiste na montagem da matriz de rigidez global $[K]$ da estrutura. Nessa etapa, as contribuições de todos os elementos são superpostas em uma única matriz, expressa em termos dos graus de liberdade globais definidos na Seção 3.2.

A ideia central é que cada coeficiente de $[k_e^g]$, que é a matriz transformada para o sistema global, representa a força nodal (ou momento) na direção do grau de liberdade global, resultante de um deslocamento unitário imposto na direção do grau de liberdade global. Assim, para que a estrutura seja tratada como um sistema único, é necessário acumular cada bloco $[k_e^g]$ dentro de $[K]$, de acordo com a correspondência entre os graus de liberdade locais do elemento e os graus de liberdade globais associados aos nós conectados por esse elemento. Essa acumulação é realizada com base na correspondência entre o índice de cada grau de liberdade do elemento e o índice do grau de liberdade global correspondente na estrutura.

Considerando uma barra de um pórtico plano com um número n de nós como modelo mais geral, cada nó possui três graus de liberdade globais, correspondentes às translações u_x , u_y e à rotação θ_z . A numeração global adotada associa ao nó n os graus de liberdade da seguinte forma:

- $3n - 2$ (translação em X),
- $3n - 1$ (translação em Y),
- $3n$ (rotação em torno de Z),

conforme já estabelecido na Seção 3.2. Para um elemento k que liga os nós i e j , é conveniente definir um vetor de correspondência ou vetor de espalhamento, conforme demonstrado por Martha (2010):

$$\{g_e\} = \begin{Bmatrix} 3i - 2 \\ 3i - 1 \\ 3i \\ 3j - 2 \\ 3j - 1 \\ 3j \end{Bmatrix} \quad (23)$$

Desta forma, pode-se associar cada grau de liberdade local do elemento à sua posição correspondente na matriz global. Esse vetor implementa, de forma compacta, a montagem por barra do Método da Rigidez Direta.

A matriz de rigidez global $[K]$ é inicialmente considerada nula. Em seguida, para cada elemento k , considera-se a sua matriz de rigidez em coordenadas globais $[k_e^g]$. Sendo α e β os índices locais de linha e coluna dessa matriz, variando de 1 a 6 e associados aos seis graus de liberdade nodais do elemento de barra de pórtico plano. O mapeamento entre esses índices locais (α, β) e os índices globais correspondentes (g_α, g_β) é estabelecido pelo vetor de correspondência do elemento, construído a partir da numeração dos nós e da regra adotada para os graus de liberdade globais. Para cada par (α, β) , a contribuição do elemento é acumulada no coeficiente global correspondente de acordo com

$$K_{g_\alpha g_\beta} = K_{g_\alpha g_\beta} + k_{e,\alpha\beta}^g, \quad \alpha, \beta = 1, \dots, 6 \quad (24)$$

Esse procedimento é repetido para todos os elementos da estrutura até que todas as contribuições tenham sido incorporadas na matriz $[K]$. Em termos matriciais, a montagem pode ser interpretada como a soma, para todos os elementos, de matrizes $[k_e^g]$ “espalhadas” em $[K]$ por meio de matrizes de incidência ou de seleção de graus de liberdade. Conforme apresenta Martha (2010), a formulação matricial do método dos deslocamentos realiza essa montagem direta da matriz de rigidez global por simples soma dos blocos de rigidez elementares, o que torna o procedimento adequado para estruturas com múltiplos graus de liberdade.

Do ponto de vista computacional, essa regra de correspondência é implementada de forma direta com base na Equação (24). Para cada elemento k , o programa monta o vetor de correspondência, que reúne os seis índices globais associados aos graus de liberdade locais do elemento. Com a matriz $[k_e^g]$ previamente calculada, percorrem-se seus coeficientes por meio de dois laços aninhados em que α e β variam de 1 a 6, atualizando-se os termos de $[K]$ nas posições (g_α, g_β) .

Embora, a montagem seja descrita sobre todos os pares (α, β) , a simetria das matrizes de rigidez locais e da matriz global permite, na implementação, operar apenas sobre uma de suas metades (por exemplo, a triangular superior), reduzindo o custo de armazenamento e o número de operações. Ao final do processo de montagem elemento a elemento, obtém-se a matriz de rigidez global $[K]$ da estrutura ainda “sem restrições”, isto é, considerando todos os graus de liberdade nodais ativos. Essa matriz é simétrica e esparsa, com a maior parte dos coeficientes não nulos concentrada em uma faixa próxima à diagonal principal, característica típica de modelos de barras e relevante para as estratégias de armazenamento e para os métodos numéricos de solução do sistema de equações. Na etapa seguinte, $[K]$ é modificada pela imposição das condições de contorno, por meio da eliminação ou da alteração dos graus de liberdade restringidos pelos apoios. A matriz de rigidez resultante, em conjunto com a matriz de massa global $[M]$, constitui a base para a formulação e solução do problema modal generalizado desenvolvido no Capítulo 4.

3.6 CONDIÇÕES DE CONTORNO

As condições de contorno da estrutura neste trabalho são representadas pelas restrições de deslocamentos e rotações impostas pelos apoios. Na formulação em deslocamentos do método da rigidez direta, essas condições também são chamadas de condições geométricas, pois especificam quais componentes de deslocamento nodal são prescritas e quais permanecem livres. No modelo adotado, essas condições são representadas por deslocamentos nulos em determinados graus de liberdade globais, correspondentes a nós apoiados ou engastados.

Ao final do processo de montagem descrito na Subseção 3.5.3, obtém-se a matriz de rigidez global $[K]$ da estrutura ainda “sem restrições”, considerando todos os graus de liberdade nodais como livres. Nessa situação, a matriz $[K]$ é singular, refletindo a existência de movimentos de corpo rígido compatíveis com o modelo estrutural e impossibilitando a obtenção de uma solução estática única. A imposição das condições de apoio, por meio da prescrição de deslocamentos nulos em determinados graus de liberdade, remove esses modos de corpo rígido e conduz a uma matriz de rigidez restringida positiva definida, o que é requisito tanto para a solução estática quanto para a formulação consistente do problema modal generalizado.

No modelo de pórtico plano adotado como caso mais geral de estrutura reticulada plana, cada nó possui três graus de liberdade: duas translações u_x e u_y e uma rotação θ_z em torno do

eixo perpendicular ao plano XY . As condições de contorno geométricas são impostas ao classificar cada uma dessas componentes como grau de liberdade livre ou restrito. Os graus de liberdade livres compõem o vetor de deslocamentos nodais desconhecidos, enquanto os graus de liberdade restritos correspondem a deslocamentos ou rotações prescritos pelos apoios, usualmente considerados nulos.

Na formulação matricial global, a aplicação das condições de contorno equivale a separar o vetor de deslocamentos nodais globais em dois subconjuntos, associados aos graus de liberdade livres e restritos. A partir da numeração dos nós e da convenção de três graus de liberdade por nó adotada na Seção 3.2, o programa identifica, para cada apoio especificado em um nó, quais componentes u_x , u_y e θ_z são restringidas. Em seguida, constrói-se o conjunto de índices dos graus de liberdade livres e, por complementaridade, o conjunto de índices dos graus de liberdade restritos.

As matrizes globais de rigidez e de massa são então reduzidas por seleção de linhas e colunas. Denotando por $[K_{ll}]$ e $[M_{ll}]$ as submatrizes obtidas a partir de $[K]$ e $[M]$ ao considerar apenas os graus de liberdade livres, o problema modal generalizado é resolvido, neste trabalho, na forma:

$$[K_{ll}] \{\hat{\phi}_j\} = \omega_j^2 [M_{ll}] \{\hat{\phi}_j\} \quad (25)$$

onde $\{\hat{\phi}_j\}$ representa o vetor modal associado ao modo j , escrito apenas em termos dos graus de liberdade livres, e ω_j é a frequência natural correspondente. As componentes dos vetores de modos completos associadas aos graus de liberdade restritos são nulas por construção. A técnica adotada na análise modal é o particionamento do sistema e a formação das submatrizes $[K_{ll}]$ e $[M_{ll}]$, reduzindo a dimensão do problema de autovalor ao conjunto de deslocabilidades livres.

Para problemas puramente estáticos, uma alternativa é a chamada técnica dos “zeros e uns”, em que as linhas e colunas associadas a deslocamentos prescritos são modificadas diretamente na matriz $[K]$, com anulação dos coeficientes fora da diagonal e atribuição de valor unitário na posição diagonal correspondente, acompanhada da anulação da componente respectiva do vetor de forças. Esse procedimento permite obter diretamente deslocamentos nulos nos graus de liberdade restringidos, preservando a forma global do sistema linear, mas não é o procedimento adotado aqui para a solução do problema modal, em que se utiliza explicitamente a redução às submatrizes $[K_{ll}]$ e $[M_{ll}]$.

Uma generalização usual é a representação de apoios flexíveis por meio de molas concentradas ligadas aos nós da estrutura. Nessa modelagem, cada apoio elástico é caracterizado por um coeficiente de rigidez k_s associado a um grau de liberdade específico, e a presença da mola é incorporada somando-se k_s ao termo diagonal correspondente da matriz $[K]$. Embora esses recursos sejam úteis em análises mais gerais, o programa desenvolvido neste trabalho considera, na análise modal de vibrações livres, apenas apoios idealmente rígidos em estruturas reticuladas planas, sem interação solo-estrutura.

4 ANÁLISE MODAL

4.1 MATRIZ DE MASSA

Conforme discutido na Seção 2.2, a formulação dinâmica de sistemas estruturais com múltiplos graus de liberdade é construída a partir da equação de movimento matricial em termos das matrizes globais de massa $[M]$ e de rigidez $[K]$. Nessas equações, $[M]$ representa a distribuição da inércia da estrutura associada aos graus de liberdade nodais do modelo discreto, enquanto $[K]$ traduz a rigidez obtida pelo Método da Rigidez Direta. A análise modal de vibrações livres, que constitui o foco deste capítulo, consiste em determinar as frequências naturais e os modos de vibração a partir da solução do problema generalizado de autovalores envolvendo essas duas matrizes.

A construção da matriz global de massa $[M]$ segue uma lógica análoga à utilizada para a matriz de rigidez global $[K]$, apresentada no Capítulo 3. Parte-se de uma matriz de massa definida em nível de elemento, escrita inicialmente em um sistema de coordenadas local associado ao eixo da barra, realiza-se a transformação para o sistema de coordenadas global adotado para a estrutura e, em seguida, monta-se a matriz global por superposição das contribuições de todos os elementos. Nesta seção, apresenta-se a formulação da matriz de massa de elemento a partir da energia cinética do meio contínuo e justifica-se a adoção da matriz de massa consistente para elementos de pórtico plano, em coerência com a discretização utilizada para a matriz de rigidez.

4.1.1 Formulação da matriz de massa de elemento

O ponto de partida para a obtenção da matriz de massa é a expressão da energia cinética de um corpo contínuo. Considerando um sólido com massa específica ρ , volume V e campo de velocidades \dot{u} , a energia cinética total é dada por:

$$E_c = \frac{1}{2} \int_V \rho \dot{u}^T \dot{u} dV \quad (26)$$

Na formulação em deslocamentos adotada neste trabalho, procede-se de forma análoga ao que é feito na dedução da matriz de rigidez: em cada elemento, o campo de deslocamentos (e, conseqüentemente, de velocidades) é aproximado por funções de forma em função das

coordenadas locais e pelos valores nodais generalizados. Utilizando essa aproximação na Equação (26), demonstra-se que a energia cinética associada ao elemento e pode ser escrita, em termos das velocidades nodais $\{\dot{u}_e^l\}$, na forma:

$$E_{c,e} = \frac{1}{2} \{\dot{u}_e^l\}^T [M_e^l] \{\dot{u}_e^l\} \quad (27)$$

em que a matriz de massa do elemento, denominada matriz de massa consistente, é definida por:

$$[M_e^l] = \int_{V_e} \rho N^T(x) N(x) dV \quad (28)$$

As Equações (27) e (28) mostram que a matriz de massa consistente é obtida pela mesma estratégia de interpolação em deslocamentos adotada na formulação dos elementos de barra.

Para elementos de barra, como os considerados neste trabalho, o volume do elemento é dado pelo produto da área da seção transversal A pelo comprimento L , de modo que a integral em (28) se reduz a uma integral unidimensional ao longo do eixo da barra. Nesses casos, é usual introduzir a massa por unidade de comprimento $m' = \rho A$ e escrever:

$$[M_e^l] = \int_0^L m' N^T(x) N(x) dx \quad (29)$$

No caso de elementos de pórtico plano baseados na teoria de Euler–Bernoulli, as funções de forma associadas aos deslocamentos transversais e rotações são polinômios cúbicos, o que conduz à matriz de massa consistente clássica proporcional a $\frac{m'L}{420}$, com coeficientes obtidos à luz da teoria dos elementos finitos (SORIANO, 2014).

4.1.2 Matriz de massa consistente e montagem global

No modelo adotado neste trabalho, cada nó da estrutura de pórtico plano possui três graus de liberdade: as translações u_x e u_y e a rotação θ_z em torno do eixo perpendicular ao plano da estrutura. Consequentemente, cada elemento de pórtico plano que liga dois nós possui seis graus de liberdade nodais, e a matriz de massa consistente local é uma matriz quadrada de

ordem 6, escrita no sistema de coordenadas local do elemento. A estrutura dessa matriz resulta da combinação dos efeitos de inércia axial e de flexão no plano, sendo expressa em termos da massa por unidade de comprimento $m' = \rho A$ e do comprimento L do elemento.

No caso específico do elemento de pórtico plano implementado neste trabalho, a matriz de massa consistente local é escrita no sistema de coordenadas do elemento, na ordenação $(u_{ix}, u_{iy}, \theta_i, u_{jx}, u_{jy}, \theta_j)$, como:

$$[M_e^l] = \frac{m'}{420} \begin{bmatrix} 140 & 0 & 0 & 70 & 0 & 0 \\ \cdot & 156 & 22L & 0 & 54 & -13L \\ \cdot & \cdot & 4L^2 & 0 & 13L & -3L^2 \\ \cdot & \cdot & \cdot & 140 & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & 156 & -22L \\ sim. & \cdot & \cdot & \cdot & \cdot & 4L^2 \end{bmatrix} \quad (30)$$

De forma análoga ao que foi feito para a matriz de rigidez no Capítulo 3, a matriz de massa consistente de cada elemento é inicialmente formulada em um sistema de coordenadas local, associado ao eixo da barra. Para que as contribuições de todos os elementos possam ser reunidas em uma matriz global única $[M]$, é necessário expressar essas matrizes locais no sistema de coordenadas global XY adotado para o modelo estrutural. Essa mudança de base é realizada por meio da mesma matriz de transformação $[T]$ apresentada na Seção 3.4, que relaciona os vetores de deslocamentos nodais locais e globais do elemento.

Se $[M_e^l]$ denota a matriz de massa consistente do elemento e $[M_e^g]$ a matriz de massa do mesmo elemento escrita em coordenadas globais, a relação entre essas matrizes é dada por:

$$M_e^g = [T]^T [M_e^l] [T] \quad (31)$$

Expressão análoga foi utilizada, no Capítulo 3, para transformar a matriz de rigidez local para o sistema global, conforme a Equação (17), o que garante que rigidez e inércia sejam tratadas de maneira consistente na formulação matricial do problema geral.

O espalhamento na matriz de massa global $[M]$ é realizado por meio do vetor de espalhamento $\{g_e\}$ de forma semelhante ao que foi feito com a matriz de rigidez $[K]$, ficando expresso como:

$$M_{g_\alpha g_\beta} = M_{g_\alpha g_\beta} + M_{e,\alpha\beta}^g, \quad \alpha, \beta = 1, \dots, 6 \quad (32)$$

Do ponto de vista numérico, existem duas escolhas principais para representar a inércia em modelos discretos: a matriz de massa consistente, obtida a partir da integração das funções de forma e a matriz de massa concentrada, em que a massa do elemento é distribuída em valores concentrados nos nós. Embora a matriz de massa concentrada apresente vantagens computacionais em alguns casos, por ser diagonal e simplificar o problema de autovalores (Soriano, 2014), a matriz de massa consistente fornece estimativas mais acuradas das frequências naturais, especialmente para modos de ordem mais elevada, além de manter a compatibilidade entre a formulação de $[M]$ e $[K]$.

Nos testes realizados durante o desenvolvimento do programa computacional deste trabalho, verificou-se que a utilização de uma matriz de massa concentrada implicava dificuldades numéricas na solução do problema generalizado de autovalores e conduzia a resultados menos estáveis para os modos superiores. Por outro lado, a adoção da matriz de massa consistente, com a estrutura 6×6 clássica de elementos de pórtico plano, mostrou-se coerente com a formulação teórica e levou a resultados mais regulares na análise modal. Por essa razão, e em consonância com a literatura de dinâmica de estruturas baseada em formulações de deslocamentos, optou-se neste trabalho pela utilização exclusiva da matriz de massa consistente na montagem da matriz global $[M]$.

4.2 PROBLEMA GENERALIZADO DE AUTOVALOR

Com as matrizes globais de rigidez $[K]$ e de massa $[M]$ já estabelecidas, a formulação da análise modal de estruturas reticuladas planas baseia-se na equação de movimento para sistemas com múltiplos graus de liberdade em vibração livre não-amortecida. O objetivo desta seção é apresentar, de forma concisa, a passagem dessa equação diferencial matricial para o problema generalizado de autovalores que fornece as frequências naturais e os modos de vibração utilizados no programa computacional desenvolvido neste trabalho.

4.2.1 Equação de movimento e solução harmônica

Para um sistema estrutural discreto sem amortecimento e sem carregamentos externos, a equação de movimento, já apresentada na Seção 2.2, é reescrita aqui na forma:

$$[M]\{\ddot{U}(t)\} + [K]\{U(t)\} = \{0\} \quad (33)$$

em que $[M]$ é a matriz global de massa, $[K]$ é a matriz global de rigidez e $\{U(t)\}$ é o vetor de deslocamentos nodais em coordenadas generalizadas em função do tempo.

Na análise modal, procura-se por movimentos livres que ocorram em regime harmônico. Adota-se, então, a hipótese de que cada modo de vibração pode ser representado por uma solução do tipo:

$$\{U(t)\} = \{\hat{\phi}\} \cos(\omega t + \psi) \quad (34)$$

em que $\{\hat{\phi}\}$ é um vetor constante de amplitudes nodais (modo de vibração), ω é a frequência natural associada ao modo considerado e ψ é uma fase inicial arbitrária. Derivando (34) duas vezes no tempo, obtém-se:

$$\{\ddot{U}(t)\} = -\omega^2 \{\hat{\phi}\} \cos(\omega t + \psi) \quad (35)$$

Substituindo $\{U(t)\}$ e $\{\ddot{U}(t)\}$ na equação de movimento (33) e fatorando o termo $\cos(\omega t + \psi)$, que não é identicamente nulo, resulta a relação algébrica:

$$([K] - \omega^2 [M])\{\hat{\phi}\} = \{0\} \quad (36)$$

4.2.2 Equação característica e interpretação dos autopares

A Equação (36) representa um sistema homogêneo de equações lineares. Para que exista solução não-trivial $\{\hat{\phi}\} \neq \{0\}$, é necessário que o determinante da matriz dos coeficientes seja nulo, o que conduz à equação característica:

$$\det([K] - \omega^2 [M]) = 0 \quad (37)$$

A Equação (37) é um polinômio em ω^2 cujo grau, em geral, é igual ao número de graus de liberdade livres do sistema. Suas raízes ω_j^2 são os autovalores do par de matrizes $([K], [M])$ e correspondem aos quadrados das frequências naturais de vibração da estrutura.

No contexto de estruturas modeladas pelo Método da Rigidez Direta, as matrizes $[K]$ e $[M]$ são simétricas e, sob hipóteses usuais de análise linear elástica com vínculos adequados (estrutura estável e bem apoiada), são também definidas positivas. Nessas condições, o

problema generalizado de autovalores admite autovalores reais e positivos ω_j^2 , o que garante que as frequências naturais ω_j sejam reais e que a resposta modal associada seja fisicamente admissível.

Para cada autovalor ω_j^2 , a Equação (36) admite uma solução não-trivial $\{\hat{\phi}_j\}$, denominada autovetor ou modo natural de vibração associado ao modo j . A relação entre rigidez, massa, frequência natural e modo de vibração pode ser escrita como:

$$[K]\{\hat{\phi}_j\} = \omega_j^2[M]\{\hat{\phi}_j\} \quad (38)$$

Como a Equação (38) é homogênea, qualquer múltiplo escalar de $\{\hat{\phi}_j\}$ também é solução. Por isso, a amplitude absoluta do modo não é determinada pela análise modal; apenas a forma do modo, as razões entre as componentes do vetor $\{\hat{\phi}_j\}$, possui significado físico. O modo associado à menor frequência natural ω_1 é denominado modo fundamental de vibração, e os demais modos estão associados a frequências naturais progressivamente mais elevadas.

4.2.3 Formulação em termos dos graus de liberdade livres

No Capítulo 3, a imposição das condições de contorno levou ao particionamento das matrizes globais de rigidez e de massa em submatrizes associadas a graus de liberdade livres e restritos. Denotando por $[K_{ll}]$ e $[M_{ll}]$ as submatrizes correspondentes ao subconjunto de graus de liberdade livres, a equação de movimento (33) pode ser reescrita, após a eliminação dos deslocamentos restritos, na forma reduzida:

$$[K_{ll}]\{\ddot{U}_l(t)\} + [M_{ll}]\{U_l(t)\} = \{0\} \quad (39)$$

Em que $\{U_l(t)\}$ contém apenas os deslocamentos e rotações nodais livres. O procedimento de redução adotado aqui é inteiramente análogo ao utilizado na análise estática pelo Método da Rigidez Direta: os graus de liberdade associados a deslocamentos prescritos são eliminados do sistema, restando um conjunto menor de equações envolvendo apenas as deslocabilidades livres.

Admitindo novamente uma solução harmônica do tipo:

$$\{U_l(t)\} = \{\hat{\phi}_j\} \cos(\omega_j t + \psi_j) \quad (40)$$

obtem-se, por procedimento idêntico ao da Subseção 4.2.1, o problema generalizado de autovalores escrito diretamente em termos das submatrizes reduzidas:

$$[K_{ll}]\{\hat{\phi}_j\} = \omega_j^2 [M_{ll}]\{\hat{\phi}_j\} \quad (41)$$

Essa relação é formalmente idêntica à Equação (25), apresentada no Capítulo 3 a partir do particionamento das matrizes globais, sendo aqui rerepresentada apenas para manter a formulação modal deste capítulo auto-contida.

É essa forma reduzida que é efetivamente utilizada no programa computacional, uma vez que os graus de liberdade associados a deslocamentos nulos por condição de apoio não participam da dinâmica da estrutura.

4.2.4 Resolução numérica no programa computacional

Do ponto de vista numérico, o problema da Equação (41) pode ser escrito como:

$$[K_{ll}]\{\hat{\phi}_j\} = \lambda_j [M_{ll}]\{\hat{\phi}_j\} \quad (42)$$

Introduziu-se $\lambda_j = \omega_j^2$ como autovalores do par de matrizes $([K_{ll}], [M_{ll}])$. Uma vez obtidos os autovalores λ_j , as frequências naturais circulares e em hertz são calculadas por:

$$\omega_j = \sqrt{\lambda_j}, \quad f_j = \frac{\omega_j}{2\pi} \quad (43)$$

O conjunto de pares $(\omega_j, \{\hat{\phi}_j\})$ caracteriza as propriedades dinâmicas inerentes da estrutura modelada, uma vez que depende apenas da distribuição de massa e de rigidez representadas por $[M]$ e $[K]$. Em termos físicos, essas grandezas podem ser interpretadas como uma “assinatura modal” do sistema, que sintetiza suas principais características de vibração.

A solução do problema generalizado de autovalores da Equação (42) é realizada, neste trabalho, por meio de rotinas numéricas apropriadas para matrizes simétricas definidas positivas, disponíveis em bibliotecas científicas de álgebra linear. No programa implementado

em linguagem *Python*, apresentado no Capítulo 5, as matrizes reduzidas $[K_u]$ e $[M_u]$ são montadas a partir das contribuições de cada elemento de pórtico plano e, em seguida, o par $([K_u], [M_u])$ é passado a um solucionador de autovalores generalizado, que retorna os autovalores λ_j e os autovetores $\{\hat{\phi}_j\}$.

Os resultados da análise modal — frequências naturais e formas de vibração — são então organizados e utilizados tanto para fins de interpretação física do comportamento dinâmico das estruturas reticuladas planas estudadas quanto para a validação do programa computacional, apresentada nos capítulos seguintes.

5 PROGRAMA COMPUTACIONAL

O presente capítulo descreve o programa computacional desenvolvido em apoio à análise modal de estruturas reticuladas planas, implementado com base nas formulações do Método da Rigidez Direta e do problema generalizado de autovalores apresentadas nos Capítulos 3 e 4. O código automatiza as etapas de montagem das matrizes globais de rigidez $[K]$ e de massa $[M]$, aplicação das condições de contorno e solução numérica do sistema modal em termos das submatrizes reduzidas $[K_{ll}]$ e $[M_{ll}]$, fornecendo como saída as frequências naturais e os modos de vibração da estrutura considerada.

Além da análise modal, o programa também permite realizar, de forma opcional, a análise estática linear da estrutura sob carregamentos nodais, utilizando a mesma discretização adotada para o estudo dinâmico. Essa funcionalidade opcional reforça o vínculo entre a formulação estática clássica do Método da Rigidez Direta e a obtenção dos parâmetros modais, favorecendo a interpretação física dos resultados.

5.1 CONSIDERAÇÕES GERAIS

O programa computacional desenvolvido consiste em um conjunto de rotinas de cálculo em linguagem *Python* destinadas à análise estática e modal de estruturas reticuladas planas, adotando o elemento de pórtico plano como modelo mais geral de estrutura bidimensional. Cada nó do modelo possui três graus de liberdade, dois de translação (u_x, u_y) e uma rotação em torno do eixo normal ao plano (θ_z), o que permite representar, em um mesmo ambiente, tanto treliças planas quanto pórticos, bastando adequar as propriedades dos elementos e as condições de restrição.

A escolha da linguagem *Python* está alinhada aos objetivos do trabalho. Trata-se de uma linguagem de alto nível, com sintaxe relativamente simples e ampla disponibilidade de bibliotecas científicas para álgebra linear e visualização. No código implementado são utilizadas, em especial, as bibliotecas *pandas*, para leitura e organização dos dados de entrada em planilhas *Excel*, *NumPy*, para manipulação matricial, *Scipy*, para a solução do problema generalizado de autovalores, e *Matplotlib*, para eventual plotagem da geometria da estrutura e das formas modais calculadas. Essa opção contrasta com o uso de programas comerciais de análise estrutural tratados como “caixa preta”, pois permite ao usuário acompanhar explicitamente como as matrizes são montadas e como o problema de autovalor é resolvido.

Do ponto de vista estrutural, o programa adota as hipóteses compatíveis com a formulação de pórtico plano em regime linear elástico. Os elementos são modelados como barras de Euler–Bernoulli com seções prismáticas, caracterizadas pelos parâmetros geométricos área A e momento de inércia I , e pelo módulo de elasticidade E e densidade ρ do material. A matriz de rigidez local 6×6 de cada elemento é construída segundo a formulação clássica de pórtico plano, em coordenadas locais, e posteriormente rotacionada para o sistema global por meio da matriz de transformação T associada aos cossenos diretores do elemento.

A inércia de massa da estrutura é representada por matrizes de massa consistentes em nível de elemento, também de ordem 6×6 , obtidas a partir da massa total associada à barra. Essa matriz local é formulada no sistema de coordenadas do elemento e , em seguida, transformada para o sistema global utilizando a mesma matriz de rotação aplicada à rigidez. A adoção exclusiva da matriz de massa consistente, em vez de uma matriz de massa concentrada, está em conformidade com a formulação teórica discutida no Capítulo 4.

O escopo do programa é restrito a estruturas planas, em vibração livre não amortecida, sem consideração de ações externas na etapa de solução modal. As condições de apoio são modeladas como apoios idealmente rígidos, por meio da imposição de deslocamentos nulos em determinados graus de liberdade, o que conduz à formação das submatrizes $[K_{ll}]$ e $[M_{ll}]$ associadas ao subconjunto de graus de liberdade livres. Nas análises estáticas opcionais, podem ser consideradas forças e momentos nodais concentrados, organizados no vetor global $\{F\}$, mas não são contemplados carregamentos distribuídos ao longo dos elementos, interação solo–estrutura ou efeitos não lineares geométricos e físicos.

O escopo do programa é restrito a estruturas planas, em vibração livre não amortecida, sem consideração de ações externas na etapa de solução modal. As condições de apoio são modeladas como apoios idealmente rígidos, por meio da imposição de deslocamentos nulos em determinados graus de liberdade, o que conduz à formação das submatrizes $[K_{ll}]$ e $[M_{ll}]$ associadas ao subconjunto de graus de liberdade livres. Nas análises estáticas opcionais, podem ser consideradas forças e momentos nodais concentrados, organizados no vetor global $\{F\}$, mas não são contemplados carregamentos distribuídos ao longo dos elementos, interação solo–estrutura ou efeitos não lineares geométricos e físicos.

Do ponto de vista numérico, o programa resolve o problema generalizado de autovalores apresentado no Capítulo 3 e expresso pela Equação (25), em que $[K_{ll}]$ e $[M_{ll}]$ são as submatrizes de rigidez e de massa associadas apenas aos graus de liberdade livres. Na implementação em *Python*, essas submatrizes são obtidas a partir de $[K]$ e $[M]$ por seleção das linhas e colunas correspondentes aos graus de liberdade livres e armazenadas em variáveis internas do código.

A solução do problema generalizado é realizada, por rotinas numéricas específicas para matrizes simétricas definidas positivas disponíveis na biblioteca *Scipy*, por meio de um solucionador de autovalores generalizado adequado à estrutura do problema. Quando essa biblioteca não está disponível, o programa recorre a um procedimento alternativo baseado na reescrita do problema na forma padrão, o que permite a execução também em ambientes com recursos mais limitados, ainda que com eficiência numérica inferior. Os autovalores obtidos são filtrados de modo a considerar apenas os valores positivos, ordenados em ordem crescente e utilizados para o cálculo das frequências naturais ω_j (em rad/s) e das frequências em hertz f_j .

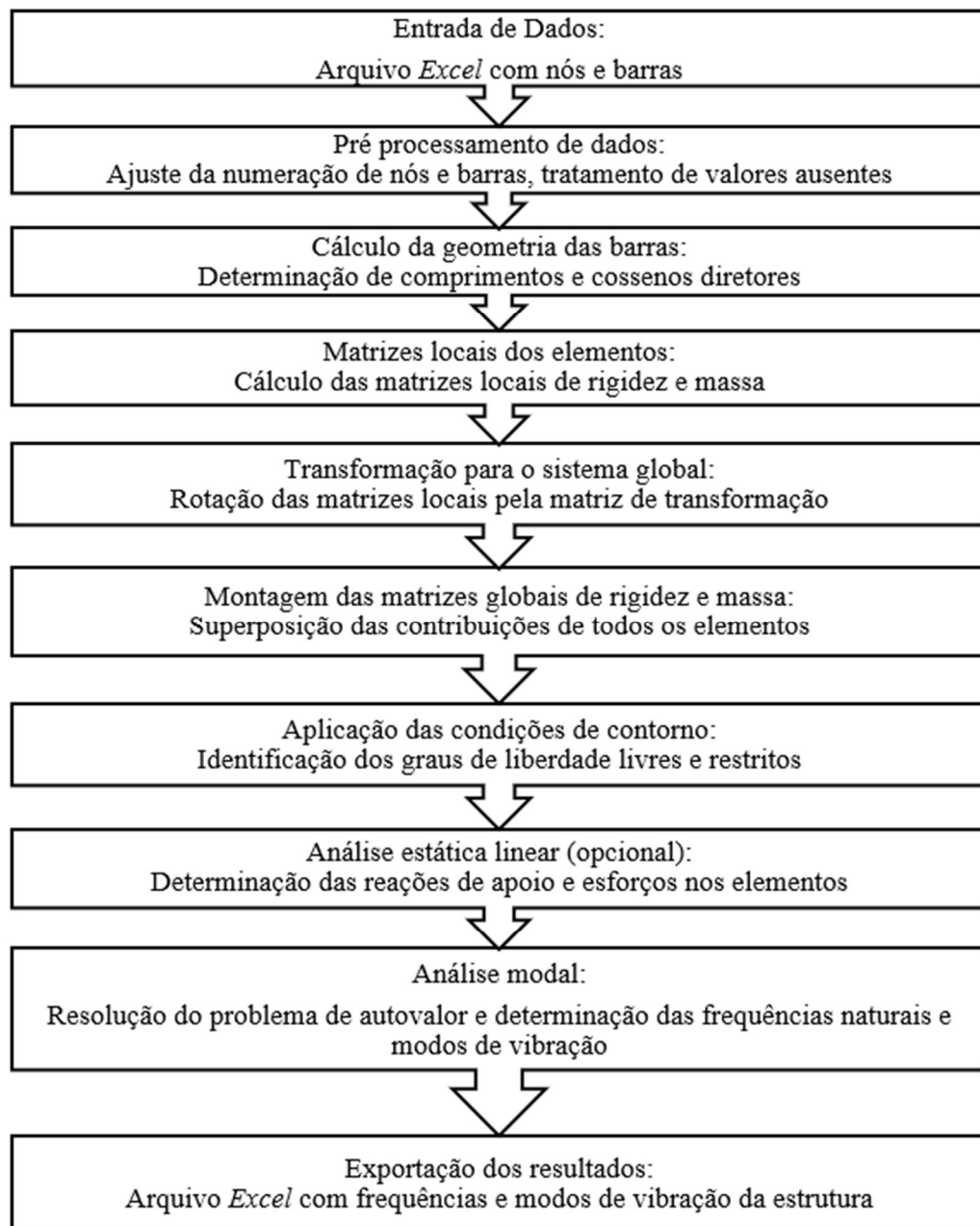
Os autovetores associados, definidos à escala arbitrária fornecida pelo algoritmo numérico, são inicialmente obtidos apenas para o subconjunto de graus de liberdade livres e, em seguida, reconstruídos pelo programa na forma de vetores modais completos, com inserção de componentes nulas nos graus de liberdade restritos, compatibilizando a solução numérica com a idealização física da estrutura.

Por fim, o programa foi estruturado em torno de uma função principal, que coordena as etapas de leitura dos dados, montagem das matrizes, aplicação das condições de contorno, solução estática e modal, exportação dos resultados para arquivos *Excel* e geração de gráficos de apoio. Nas seções seguintes serão detalhados o funcionamento do programa, o fluxo de informações entre as rotinas e a forma como cada etapa da formulação teórica apresentada nos Capítulos 3 e 4 é traduzida em instruções computacionais.

5.2 FUNCIONAMENTO DO PROGRAMA

A presente seção descreve o fluxo de execução do programa, desde a leitura dos dados de entrada até a geração dos resultados em arquivos *Excel*. A organização segue a estrutura em blocos utilizada no código. A Figura 6 apresenta o esquema do processo executado pelo programa, cujo código completo em *Python* está no Apêndice A.

Figura 6 – Etapas do programa



Fonte: Autor (2025)

5.2.1 Estrutura geral do código

O programa foi implementado em um único algoritmo principal em *Python* que está na íntegra no Apêndice A. O programa é organizado em grupos com funções que reproduzem o procedimento do Método da Rigidez Direta. A função de mais alto nível é *executar_analise*, que coordena todas as etapas da análise a partir de um arquivo de entrada em formato *Excel*. Quando o código é executado diretamente, o usuário informa o caminho do arquivo extensão

“.xlsx” e essa função é automaticamente chamada, a partir daí realiza a leitura dos dados, a montagem das matrizes globais, a aplicação das condições de contorno, opcionalmente análise estática, a análise modal e a exportação dos resultados.

As rotinas são agrupadas em seis blocos principais. O primeiro bloco corresponde às funções de leitura e pré-processamento dos dados, sendo destacadas:

- *ler_dados_entrada*, que importa as abas “Nós” e “Barras” de uma planilha *Excel* e ajusta a indexação para que a numeração de nós e barras comece em 1;
- *calcular_geometria_barras*, que calcula o comprimento e os cossenos diretores de cada elemento a partir das coordenadas nodais.

Em seguida, um segundo bloco reúne as funções responsáveis pela definição das matrizes locais de rigidez e de massa e da matriz de rotação dos elementos. O terceiro bloco lida com a montagem das matrizes globais de rigidez $[K]$ e de massa $[M]$, por meio da função *montar_matrizes_globais*, que percorre todos os elementos, calcula suas contribuições em coordenadas globais e as acumula nas posições apropriadas das matrizes globais, conforme o processo de montagem da matriz global do Método da Rigidez Direta. O quarto bloco agrupa as rotinas associadas às condições de contorno, destaca-se a função *obter_graus_livres*, que, com base nas restrições RX , RY e $R\theta$, identifica os graus de liberdade restritos e livres e prepara o sistema para ser particionado conforme discutido no Capítulo 3.

O quinto bloco engloba a análise estática, com as funções:

- *montar_vetor_forcas*, que constrói o vetor global de carregamentos nodais $\{F\}$;
- *resolver_estatico*, que resolve o sistema reduzido em termos dos graus de liberdade livres e reconstrói o vetor completo de deslocamentos;
- *calcular_esforços_elementares*, que determina os esforços de extremidade em cada barra, e *exportar_resultados_estatica*, que organiza esses resultados em planilhas de saída.

Por fim, o sexto bloco corresponde à análise modal, com a função *resolver_problema_modal*, que soluciona o problema generalizado de autovalores, *reconstruir_modos_completos*, que obtém os vetores modais completos e *exportar_resultados_modal*, que gera as frequências e modos em um arquivo *Excel*.

A estrutura do código faz com que cada grupo de funções corresponda diretamente a uma etapa da formulação teórica: definição do modelo e dos graus de liberdade, montagem das matrizes elementares e globais, imposição das condições de contorno, solução dos problemas estático e modal e pós-processamento dos resultados. Esse paralelismo facilita tanto a

compreensão do código quanto a verificação da consistência entre o modelo matemático e a sua implementação computacional.

5.2.2 Entrada e pré-processamento dos dados

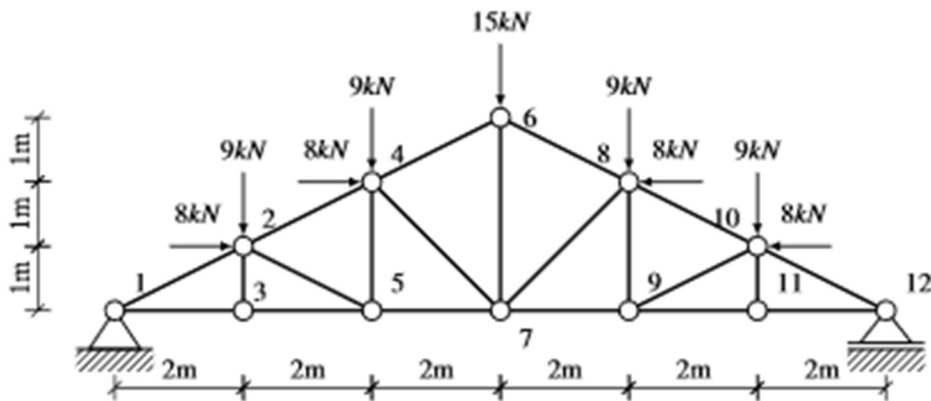
A entrada de dados do programa é feita por meio de uma planilha em formato “.xlsx”, que deve conter obrigatoriamente, duas abas denominadas “Nós” e “Barras”. Cada linha da aba “Nós” representa um nó da estrutura e as colunas definem suas coordenadas, restrições e, opcionalmente, carregamentos nodais. As colunas mínimas esperadas são X e Y , que armazenam as coordenadas do nó no sistema global, RX e RY , que indicam se as translações nas direções X e Y estão restritas (valor 1) ou livres (valor 0), e $R\theta$, que desempenha o mesmo papel para a rotação θ_z . Podem ser incluídas ainda as colunas FX e FY , correspondentes às forças concentradas aplicadas no nó nas direções X e Y , e MZ , associada ao momento nodal em torno do eixo normal ao plano. Caso essas colunas de carregamento não estejam presentes, o programa assume que os respectivos valores são nulos. A Tabela 1 mostra um exemplo da aba “Nós” para a estrutura exibida na Figura 7.

Tabela 1 – Exemplo da aba “Nós”

X	Y	RX	RY	Rθ	FX	FY	MZ
0	0	1	1	0	0	0	
2	1				8000	-9000	
2	0						
4	2				8000	-9000	
4	0						
6	3				0	-15000	
6	0						
8	2				-8000	-9000	
8	0						
10	1				-8000	-9000	
10	0						
12	0	0	1	0			

Fonte: Autor (2025)

Figura 7 – Exemplo de estrutura



Fonte: Matos e Titello (2020)

Na aba “Barras”, cada linha representa um elemento de barra ligando dois nós do modelo. As colunas $N1$ e $N2$ identificam o nó inicial e o nó final de cada elemento, de acordo com a numeração adotada na aba “Nós”. As propriedades geométricas e mecânicas do elemento são definidas pelas colunas A , E e I , correspondentes à área da seção transversal, ao módulo de elasticidade e ao momento de inércia fletor. Além disso, a planilha deve conter uma coluna associada à massa do elemento, que pode ser denominada “rho”, quando se utiliza densidade volumétrica ρ em unidades de massa por volume, ou “ $m_especifica$ ”, quando a informação de entrada é a massa linear m' em unidades de massa por comprimento. A escolha do parâmetro de massa efetivamente utilizado é controlada, na chamada da função principal, pelo argumento *mass_is_linear*. A Tabela 2 mostra um exemplo da aba “Barras” para a estrutura exibida da Figura 7.

Na etapa de leitura, a função *ler_dados_entrada* importa as duas abas, ajusta os índices das linhas de forma que a numeração de nós e barras comece em 1, e substitui eventuais valores ausentes por zero. Em seguida, a função *calcular_geometria_barras* utiliza as coordenadas X e Y dos nós indicados em $N1$ e $N2$ para calcular o comprimento L de cada barra e os cossenos diretores c associados à sua orientação no plano, armazenando esses valores em novas colunas da aba “Barras”. Esses parâmetros geométricos são consistentes com os empregados na formulação do elemento de pórtico plano discutida no Capítulo 3 e são utilizados nas etapas seguintes para construir as matrizes locais e a matriz de rotação de cada elemento.

Tabela 2 – Exemplo da aba “Barras”

N1	N2	A (m ²)	E (GPa)	m_especifica (kg/m ³)	I (m ⁴)
1	2	0,01	2,10E+11	7850	8,33333E-06
1	3	0,01	2,10E+11	7850	8,33333E-06
2	3	0,01	2,10E+11	7850	8,33333E-06
2	4	0,01	2,10E+11	7850	8,33333E-06
2	5	0,01	2,10E+11	7850	8,33333E-06
3	5	0,01	2,10E+11	7850	8,33333E-06
4	5	0,01	2,10E+11	7850	8,33333E-06
4	6	0,01	2,10E+11	7850	8,33333E-06
4	7	0,01	2,10E+11	7850	8,33333E-06
5	7	0,01	2,10E+11	7850	8,33333E-06
6	7	0,01	2,10E+11	7850	8,33333E-06
6	8	0,01	2,10E+11	7850	8,33333E-06
7	8	0,01	2,10E+11	7850	8,33333E-06
7	9	0,01	2,10E+11	7850	8,33333E-06
8	9	0,01	2,10E+11	7850	8,33333E-06
8	10	0,01	2,10E+11	7850	8,33333E-06
9	10	0,01	2,10E+11	7850	8,33333E-06
9	11	0,01	2,10E+11	7850	8,33333E-06
10	11	0,01	2,10E+11	7850	8,33333E-06
10	12	0,01	2,10E+11	7850	8,33333E-06
11	12	0,01	2,10E+11	7850	8,33333E-06

Fonte: Autor (2025)

5.2.3 Montagem das matrizes globais de rigidez e de massa

A partir dos dados pré-processados, o programa procede à montagem das matrizes globais de rigidez $[K]$ e de massa $[M]$ da estrutura. Essa etapa é implementada na função *montar_matrizes_globais*, que percorre todos os elementos listados na aba “Barras”, obtém para cada um os nós extremos, as coordenadas globais desses nós, as propriedades A , E e I e o parâmetro de massa, e constrói as respectivas matrizes locais de rigidez e de massa no sistema de coordenadas do elemento.

A matriz de rigidez local é obtida a partir da formulação clássica do elemento de pórtico plano, resultando em uma matriz 6×6 no sistema local definido pelos deslocamentos de translação e rotação nos nós inicial e final. A matriz de massa local também é construída na forma 6×6 , a partir da massa total da barra, calculada com base na densidade volumétrica e na área, ou na massa linear, conforme a coluna de massa disponível na planilha e a opção escolhida pelo usuário. Em seguida, para cada elemento, é construída uma matriz de rotação $[T]$, dependente dos cossenos diretores c , que relaciona os deslocamentos locais aos deslocamentos no sistema global.

A contribuição de cada elemento para as matrizes globais é obtida por meio das transformações clássicas:

$$[k_e^g] = [T]^T [k_e^l] [T] \quad \text{e} \quad [M_e^g] = [T]^T [M_e^l] [T] \quad (44)$$

em que $[k_e^l]$ e $[M_e^l]$ são as matrizes locais de rigidez e de massa no sistema do elemento e $[k_e^g]$ e $[M_e^g]$ representam as mesmas matrizes expressas no sistema global. Para cada barra, o programa determina o conjunto de seis graus de liberdade globais associados aos nós N1 e N2, monta a matriz de índices que mapeia as posições locais para as posições globais e adiciona as contribuições $[k_e^g]$ e $[M_e^g]$ às posições correspondentes das matrizes $[K]$ e $[M]$. O procedimento é repetido para todos os elementos da estrutura, resultando na forma global das matrizes de rigidez e massa compatível com a discretização adotada.

5.2.4 Aplicação das condições de contorno e solução dos problemas estático e modal

Com as matrizes globais $[K]$ e $[M]$ montadas, o programa aplica as condições de contorno e resolve, em sequência, o problema estático e o problema modal. A identificação dos graus de liberdade livres e restritos é realizada pela função *obter_graus_livres*, que percorre a aba “Nós” e, para cada nó, avalia os valores das colunas R_X , R_Y e R_θ . Quando o valor de uma dessas colunas é igual a 1, o grau de liberdade correspondente é considerado restrito, representando um apoio idealmente rígido; quando é igual a 0, o grau de liberdade é considerado livre. A partir dessa classificação, o programa constrói os vetores de índices de graus de liberdade livres e restritos e determina, por diferença, o conjunto de todos os graus de liberdade do sistema.

Na análise estática, que é uma funcionalidade complementar devido aos objetivos principais desse trabalho, o vetor global de forças nodais $\{F\}$ é montado pela função *montar_vetor_forcas*, que lê as colunas F_X , F_Y e M_Z da aba “Nós” e distribui os carregamentos nos graus de liberdade associados às translações e rotações de cada nó. A função *resolver_estatico* então particiona o sistema global da Equação (7) em termos dos graus de liberdade livres e resolve o sistema reduzido:

$$[K_u]\{U_l\} = \{F_l\} \quad (45)$$

Impondo-se deslocamentos nulos nos graus de liberdade restritos. A partir da solução $\{U_l\}$, o programa reconstrói o vetor completo de deslocamentos $\{U\}$ e calcula as reações de apoio. Em seguida, a função *calcular_esforços_elementares* avalia os esforços de extremidade em cada barra a partir dos deslocamentos nodais, permitindo obter forças normais, cortantes e momentos fletores em nível de elemento.

Na análise modal, a função *resolver_problema_modal* utiliza os vetores de graus de liberdade livres para extrair das matrizes globais as submatrizes $[K_{ll}]$ e $[M_{ll}]$ associadas apenas a esses graus, conforme a Equação (25). O problema generalizado de autovalores é resolvido com o auxílio de rotinas numéricas específicas para matrizes simétricas definidas positivas, disponibilizadas pela biblioteca *Scipy*. Os autovalores obtidos são filtrados para remover valores negativos ou nulos decorrentes de imperfeições numéricas, ordenados em ordem crescente e utilizados para calcular as frequências naturais ω_j e as frequências em hertz f_j , onde o subíndice j indica o número do modo natural. Os autovetores correspondentes, inicialmente obtidos apenas para o subconjunto de graus de liberdade livres, são normalizados de forma que o valor absoluto máximo de cada vetor seja igual a 1, o que facilita a comparação entre modos e a exportação para eventual visualização gráfica das deformadas modais.

A reconstrução dos vetores modais completos é realizada pela função *reconstruir_modos_completos*, que insere componentes nulas nos graus de liberdade restritos e distribui os valores dos autovetores reduzidos nas posições dos graus de liberdade livres, resultando em um conjunto de modos $\{\hat{\phi}_j\}$ compatível com a idealização física da estrutura. Dessa forma, garante-se que os deslocamentos associados a apoios e demais restrições permaneçam nulos nos modos de vibração, em conformidade com as condições de contorno.

Para fins de visualização, os modos são apresentados com normalização e amplificação gráfica, de modo que as amplitudes plotadas não representam deslocamentos físicos reais. Além disso, a deformada modal é traçada ligando-se os nós deformados por segmentos retos, a partir dos deslocamentos nodais, sem interpolação ao longo do elemento, o que pode resultar em aparência segmentada quando comparada a softwares que utilizam suavização por pós-processamento.

5.2.5 Saída de resultados e interface com o usuário

A etapa final de execução corresponde ao pós-processamento e à apresentação dos resultados ao usuário. Os resultados da análise modal são organizados e exportados pela função

exportar_resultados_modal, que gera um arquivo *Excel*, no qual a primeira planilha reúne, em forma de tabela, as frequências naturais em *rad/s* e em hertz e, na aba seguinte, são armazenados os componentes dos vetores modais completos, organizados por grau de liberdade e por modo.

Do ponto de vista do usuário, o fluxo de utilização do programa resume-se a quatro passos principais. Em primeiro lugar, é necessário preparar a planilha de entrada com as abas “Nós” e “Barras” no formato descrito na Subseção 5.2.2. Em seguida, o usuário executa o programa em ambiente *Python*. Quando solicitado pelo programa, informa o caminho completo do arquivo *Excel* e define o parâmetro *mass_is_linear* de acordo com o tipo de informação de massa disponível na planilha. Após a execução, o usuário dispõe dos arquivos *Excel* gerados para uma análise quantitativa.

6 EXEMPLOS NUMÉRICOS

Neste capítulo são apresentados exemplos com o objetivo de verificar o funcionamento do programa computacional desenvolvido e discutir sua aplicação prática. A validação é conduzida por comparação com resultados de referência obtidos no software comercial Autodesk *Robot Structural Analysis Professional*, adotando-se modelos equivalentes em termos de geometria, propriedades dos elementos, condições de contorno e hipóteses de análise.

A comparação contempla resultados de análise modal, em especial as frequências naturais e as formas modais associadas, que constituem a principal saída do programa e sintetizam a distribuição conjunta de rigidez e massa do sistema.

Serão considerados dois exemplos representativos, ambos modelados como pórticos planos. O primeiro utiliza uma discretização reduzida, permitindo verificar de forma objetiva a montagem das matrizes globais e a solução do problema de autovalores. O segundo amplia a complexidade estrutural, com maior número de nós e elementos, avaliando a robustez do procedimento numérico e a consistência dos resultados em uma configuração estrutural com muitos graus de liberdade.

6.1 VALIDAÇÃO DO PROGRAMA

A validação do programa é realizada por meio da comparação direta entre os resultados numéricos fornecidos pelo código em *Python* e aqueles obtidos *no Robot*, adotado como referência externa. Para cada exemplo, a estrutura é modelada nos dois ambientes com a mesma discretização, mantendo-se consistência na numeração de nós e elementos, nas propriedades mecânicas e geométricas e nas restrições nodais. Embora o programa tenha rotinas estáticas, a validação apresentada foca no objetivo principal da análise modal, onde são comparadas as frequências naturais dos primeiros modos.

A diferença entre resultados é quantificada por meio do erro relativo percentual. Para uma grandeza genérica x , define-se:

$$\varepsilon_x = \frac{|x_{\text{prog}} - x_{\text{ref}}|}{|x_{\text{ref}}|} \cdot 100 \quad (46)$$

onde x_{prog} representa o valor fornecido pelo programa e x_{ref} o valor de referência obtido no *Robot*. Para frequências naturais, aplica-se a mesma definição a cada modo j , substituindo-se x por f_j ou ω_j , conforme a unidade adotada na apresentação dos resultados.

6.2 COMPARAÇÃO QUALITATIVA DAS FORMAS MODAIS

Além da comparação quantitativa das frequências, é apresentada uma comparação qualitativa das formas modais por meio de pares de figuras geradas pelo programa em *Python* e pelo *Autodesk Robot Structural Analysis Professional*. Visualmente compara-se o padrão relativo do modo, configuração global do movimento e a relação entre deslocamentos nos nós, e não a amplitude absoluta.

As deformadas apresentadas pelo *Robot* tendem a ser visualmente mais suaves porque o *software* representa a linha média deformada do elemento com interpolação ao longo do seu comprimento, associada às funções de forma do elemento. No programa, a deformada é construída a partir dos deslocamentos nodais do autovetor e traçada por segmentos retos unindo os nós deformados, o que produz uma aparência segmentada.

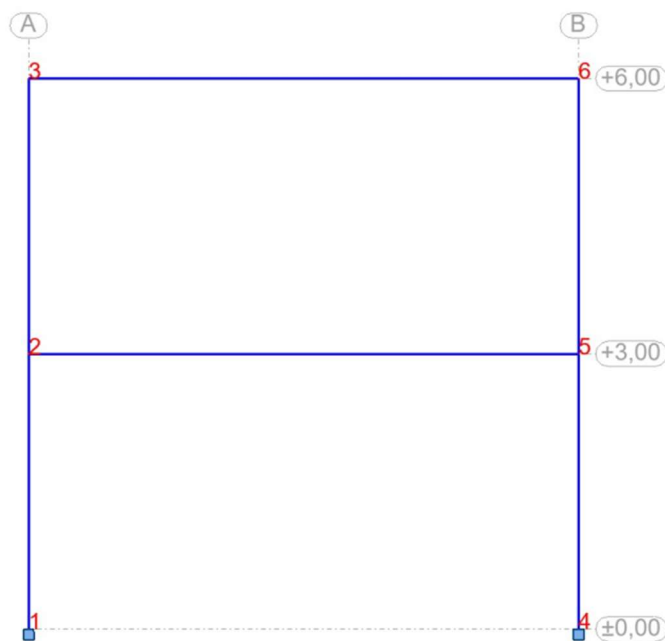
A diferença visual observada entre as deformadas geradas pelo código desenvolvido e o do *Robot* é majoritariamente decorrente do pós-processamento e da forma de representação gráfica, não implicando necessariamente divergência nos resultados numéricos do problema modal. Ressalta-se ainda que os autovetores são normalizados para fins de visualização, tendo seu máximo absoluto igual a 1, e as deformadas são amplificadas por um fator de escala, portanto, as amplitudes apresentadas não representam deslocamentos físicos reais. A inversão de sinal do autovetor associado a um modo também não altera sua interpretação.

6.3 EXEMPLO 1

Para a estrutura apresentada na Figura 8, deseja-se determinar as frequências naturais dos três primeiros modos e as três primeiras formas modais de vibração, com o objetivo de validar as rotinas de montagem das matrizes globais de rigidez e de massa e a resolução do problema de autovalores generalizado. A validação é realizada por comparação direta entre os resultados obtidos pelo programa e aqueles fornecidos pelo software comercial *Autodesk Robot Structural Analysis Professional*, adotado como referência.

Por se tratar de um trabalho com foco na análise modal, não é conduzida análise estática neste exemplo. Além disso, para complementar a comparação quantitativa, apresentam-se também as deformadas modais obtidas no programa em *Python* e no *Autodesk Robot Structural Analysis Professional*. As figuras têm finalidade de comparação qualitativa do padrão global de deformação de cada modo, enquanto a comparação quantitativa é feita por meio das tabelas de frequências e erros relativos, conforme a Equação (46).

Figura 8 – Estrutura de pórtico para o Exemplo 1



Fonte: Autor (2025)

6.3.1 Geometria e conectividade

A estrutura é definida como um pórtico de dois pavimentos e um vão, com distância entre eixos dos pilares de 6,0 m e altura total de 6,0 m, sendo 3,0 m por pavimento. O modelo é discretizado em seis elementos de pórtico, com dois trechos por pilar e um elemento por viga em cada nível, de modo a permitir a representação do comportamento por flexão e esforço normal com um número reduzido de barras. As coordenadas e restrições do exemplo são exibidas no formato do arquivo de entrada na Tabela 3.

Tabela 3 – Coordenadas nodais e restrições do Exemplo 1

Nó	X	Y	RX	RY	RΘ
1	0,0	0,0	1	1	1
2	0,0	3,0	0	0	0
3	0,0	6,0	0	0	0
4	6,0	0,0	1	1	1
5	6,0	3,0	0	0	0
6	6,0	6,0	0	0	0

Fonte: Autor (2025)

A conectividade dos elementos é indicada na Tabela 4.

Tabela 4 – Conectividade dos elementos

Barra	N1	N2
1	1	2
2	2	3
3	4	5
4	5	6
5	2	5
6	3	6

Fonte: Autor (2025)

6.3.2 Propriedades do material, seção e massa

Adota-se concreto armado estrutural de $30MPa$, com propriedades usuais na prática e que podem ser definíveis no *Robot*:

- $E = 27 \times 10^6$ kPa;
- $\nu = 0,20$;
- $\rho = 2500$ kg/m³.

Para reduzir ambiguidades de orientação de eixos locais e facilitar a correspondência entre os modelos, todas as barras são adotadas com seção quadrada de $0,40 \times 0,40$ m, resultando em:

- $A = 0,16$ m²;
- $I = \frac{0,40^4}{12} = 2,133 \times 10^{-3}$ m⁴.

A massa específica linear associada ao elemento, utilizada na formulação de massa consistente do programa, é:

- $m_{\text{específica}} = \rho A = 2500 \cdot 0,16 = 400 \text{ kg/m}$.

Os valores de E , A , I e $m_{\text{específica}}$ devem ser atribuídos a todas as barras na planilha de entrada do programa e definidos de forma equivalente no *Robot* por meio do material e da seção adotados.

6.3.3 Condições de contorno e hipótese de modelo plano

As bases dos pilares são engastadas, impondo-se deslocamentos e rotação nulos nos nós 1 e 4:

- $u_x = u_y = \theta_z = 0$ em $\{1,4\}$.

No *Robot*, a análise é realizada em ambiente tridimensional. Para manter equivalência com o modelo plano do programa, o pórtico é modelado no plano XZ do *Robot*, adotando Z como direção vertical. Assim, a coordenada Y do programa corresponde à coordenada Z no *Robot*. Para evitar a obtenção de modos fora do plano, restringem-se em todos os nós os graus de liberdade fora do plano, mantendo apenas as componentes compatíveis com o modelo bidimensional (translações no plano e rotação normal ao plano).

6.3.4 Análise modal e critérios de comparação

A análise modal é realizada a partir do problema de autovalores generalizado, considerando as matrizes globais reduzidas aos graus de liberdade livres. Como o modelo possui 6 nós com 3 graus de liberdade por nó e 2 nós engastados, o sistema reduzido resulta em 12 graus de liberdade livres, produzindo 12 autovalores não nulos. Neste exemplo, são extraídas as três menores frequências naturais não nulas e as formas modais correspondentes.

A validação é feita pela comparação entre o programa e o *Robot*, sendo estabelecida pela comparação quantitativa das frequências naturais dos três primeiros modos, apresentada em forma tabular, com cálculo do erro relativo percentual conforme a Equação (46).

A Tabela 5 apresenta os valores das três primeiras frequências naturais obtidas no *Robot* e no programa, bem como os erros relativos:

Tabela 5 – Comparação dos valores de frequência

Modo j	f_{Robot} (Hz)	$f_{programa}$ (Hz)	ε_f (%)
1	7,00	7,069	0,98
2	24,70	24,984	1,15
3	34,13	34,546	1,22

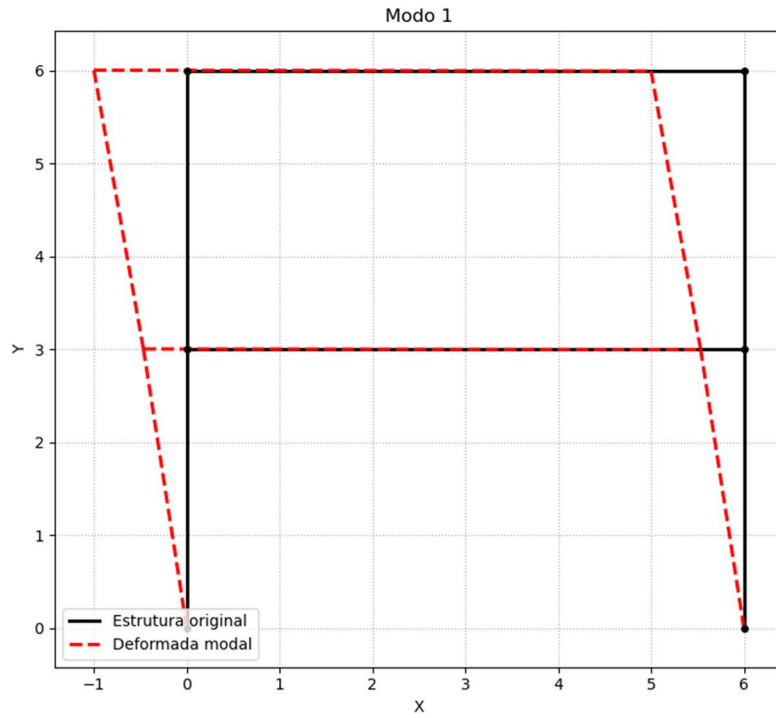
Fonte: Autor (2025)

Observa-se concordância muito próxima entre os resultados, com erros relativos inferiores a 1,3% para os três primeiros modos. Essa discrepância de valores é compatível com pequenas diferenças numéricas e de implementação entre os ambientes (tratamento interno de massa distribuída e convenções de modelagem), e indica que os resultados obtidos pelas rotinas do programa estão coerentes com as de uma referência comercial.

As Figuras 9 a 14 apresentam, para os três primeiros modos, as deformadas modais do Exemplo 1 obtidas no programa desenvolvido e no *Robot*. Visualmente compara-se o padrão relativo de deslocamentos e a configuração global do modo, e não a amplitude absoluta. O *Robot* apresenta deformadas mais suaves por interpolação ao longo do elemento, representando a linha média deformada com base nas funções de forma do elemento. No programa, a deformada é construída apenas com deslocamentos translacionais nodais e ligação reta entre nós, resultando em uma aparência segmentada. Assim, a diferença observada é predominantemente de visualização, não implicando necessariamente divergência nos resultados numéricos do problema modal. Os autovetores são normalizados para fins de plotagem e as deformadas são amplificadas por um fator de escala; portanto, as amplitudes apresentadas não representam deslocamentos físicos reais, e o sinal de um modo pode ser invertido sem alterar seu significado.

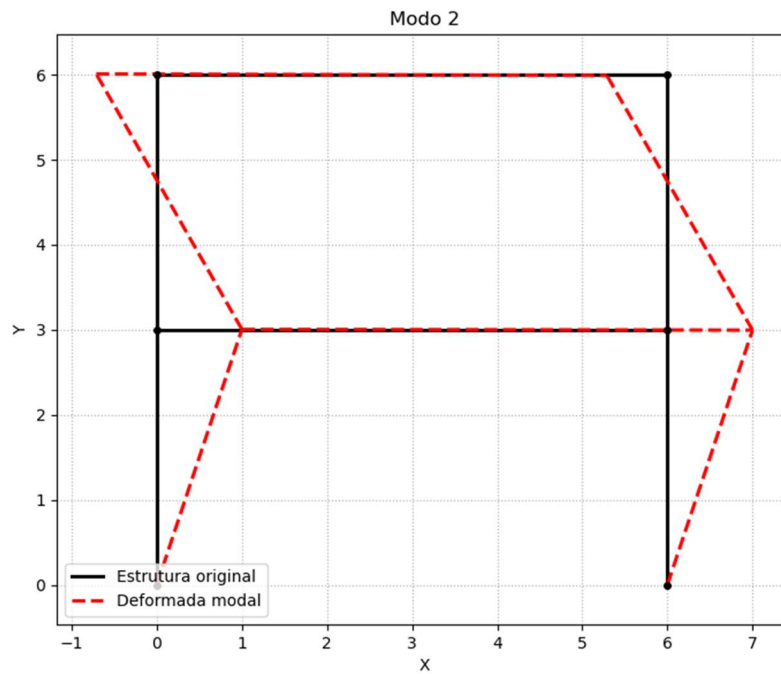
A comparação qualitativa das formas modais é interpretada a partir da configuração relativa do modo, observando-se: coerência com as condições de contorno, com deslocamentos nulos nos apoios; padrão global do movimento, com regiões de maior participação modal, inversões de curvatura e simetrias quando aplicáveis; e compatibilidade entre os modos apresentados no programa e no *Robot*. Nesta comparação não se avalia a amplitude absoluta da deformada.

Figura 9 – Modo 1 do Exemplo 1, deformada modal no programa



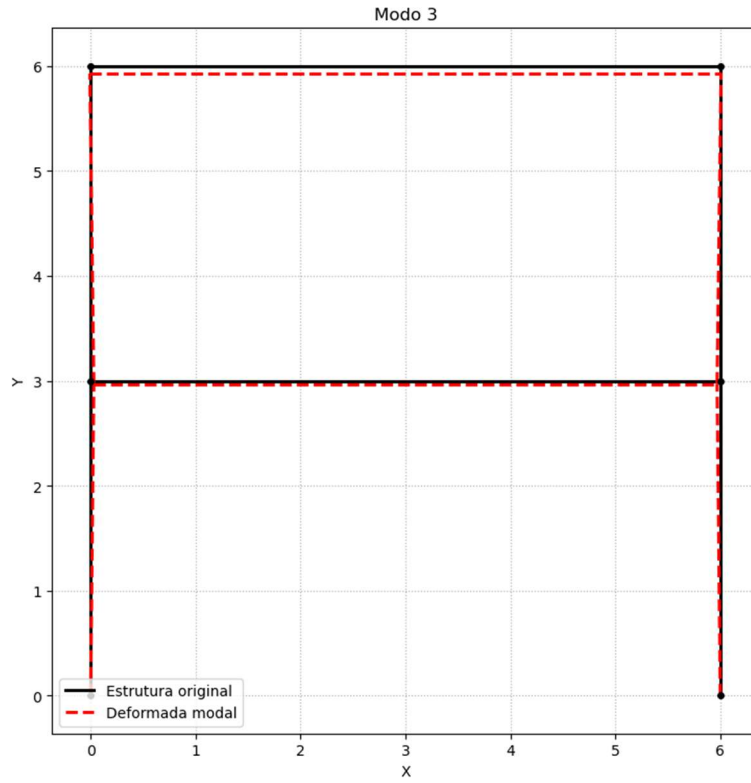
Fonte: Autor (2025)

Figura 10 – Modo 2 do Exemplo 1, deformada modal no programa



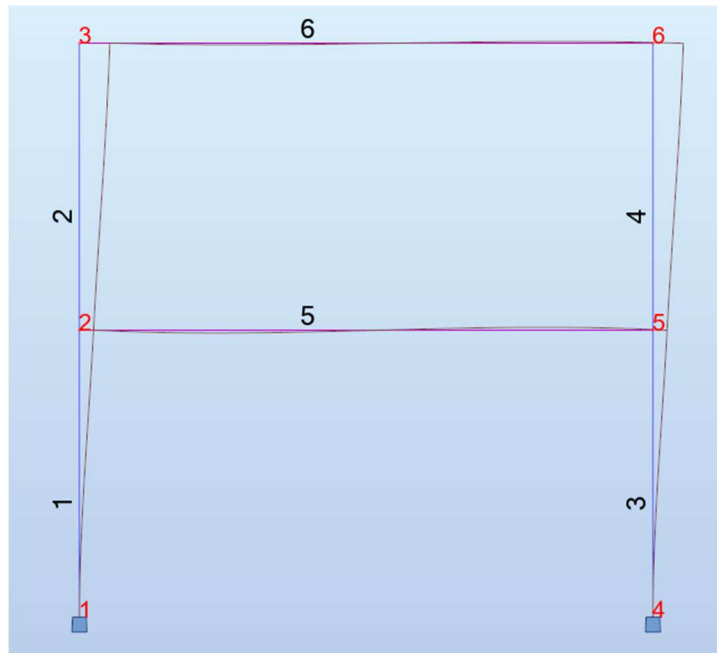
Fonte: Autor (2025)

Figura 11 – Modo 3 do Exemplo 1, deformada modal no programa



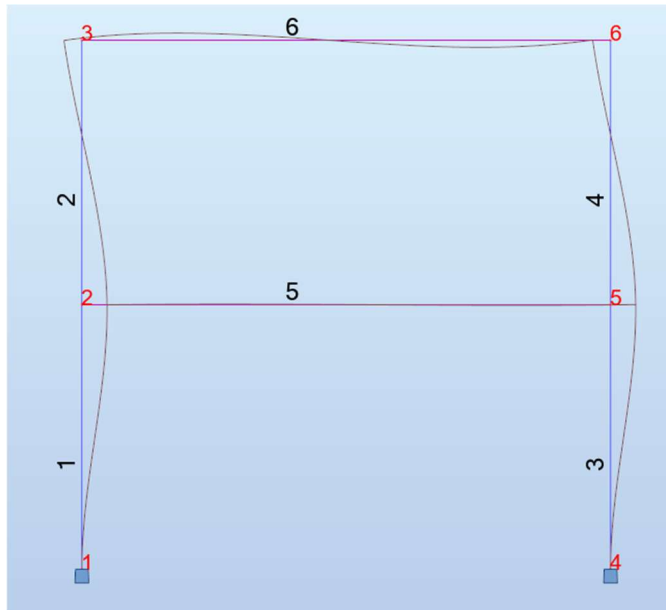
Fonte: Autor (2025)

Figura 12 – Modo 1 do Exemplo 1, deformada modal no *Robot*



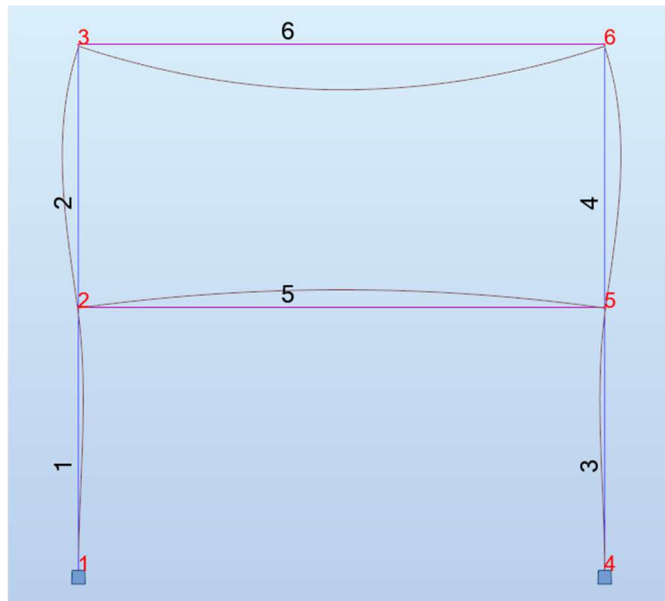
Fonte: Autor (2025)

Figura 13 – Modo 2 do Exemplo 1, deformada modal no *Robot*



Fonte: Autor (2025)

Figura 14 – Modo 3 do Exemplo 1, deformada modal no *Robot*



Fonte: Autor (2025)

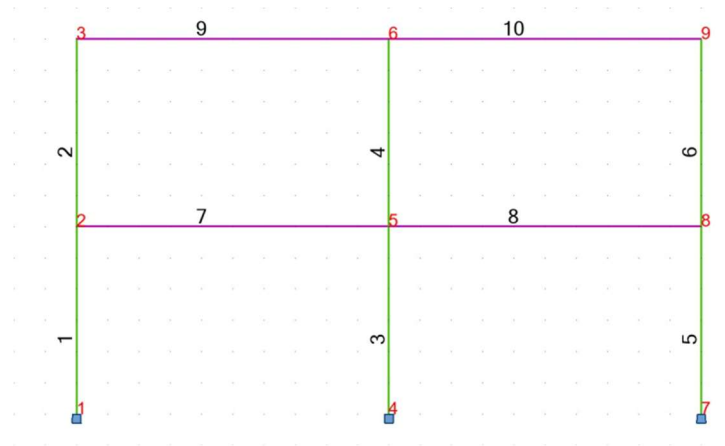
Com base na proximidade das frequências naturais e nas formas modais com configuração condizente com a do *Robot*, considera-se o Exemplo 1 validado para os propósitos deste trabalho, servindo como verificação objetiva da implementação do elemento de pórtico plano com matriz de massa consistente e da solução numérica do problema de autovalores generalizado.

6.4 EXEMPLO 2

Este exemplo tem como objetivo complementar a validação apresentada no Exemplo 1, verificando o desempenho do programa em uma estrutura com maior nível de requisição computacional, maior número de nós, elementos e graus de liberdade livres. Enquanto o Exemplo 1 utiliza um pórtico de dois pavimentos e um vão, aqui adota-se um pórtico de dois pavimentos e dois vãos, com três alinhamentos de pilares, conforme mostrado na Figura 15.

Mantêm-se as hipóteses de análise, propriedades de material, seção transversal e critérios de comparação adotados no Exemplo 1, modificando-se apenas a geometria, a conectividade e as condições de contorno do pórtico. A validação é conduzida exclusivamente no âmbito da análise modal, sem realização de análise estática, por meio da comparação direta entre as três primeiras frequências naturais obtidas no programa e no *Robot*, com quantificação das diferenças mediante erros relativos calculados conforme a Equação (46). Para a comparação qualitativa das formas modais, adotam-se os mesmos critérios de interpretação apresentados no Exemplo 1, com foco no padrão relativo do modo e na coerência física do movimento global.

Figura 15 – Estrutura de pórtico para o Exemplo 2



Fonte: Autor (2025)

6.4.1 Geometria e conectividade

A estrutura é definida como um pórtico plano de dois pavimentos e dois vãos, com vãos de 5,0 m e altura total de 6,0 m, sendo 3,0 m por pavimento. O modelo é composto por 9 nós e 10 elementos de pórtico, com dois trechos por pilar em cada alinhamento vertical e vigas representadas por elementos em cada nível, garantindo correspondência direta com o arquivo de entrada utilizado no programa.

Tabela 6 – Coordenadas nodais e restrições do Exemplo 2

Nó	X (m)	Y (m)	RX	RY	RΘ
1	0,0	0,0	1	1	1
2	0,0	3,0	0	0	0
3	0,0	6,0	0	0	0
4	5,0	0,0	1	1	1
5	5,0	3,0	0	0	0
6	5,0	6,0	0	0	0
7	10,0	0,0	1	1	1
8	10,0	3,0	0	0	0
9	10,0	6,0	0	0	0

Fonte: Autor (2025)

Tabela 7 – Conectividade dos elementos

Barra	N1	N2
1	1	2
2	2	3
3	4	5
4	5	6
5	7	8
6	8	9
7	2	5
8	5	8
9	3	6
10	6	9

Fonte: Autor (2025)

6.4.2 Propriedades do material, seção e massa

Mantêm-se as hipóteses de análise e os critérios de comparação estabelecidos no Exemplo 1, alterando-se apenas a geometria, a conectividade e as condições de contorno correspondentes. A validação é conduzida exclusivamente no âmbito modal, por comparação direta com o *Robot* das três primeiras frequências naturais, organizadas em tabelas com cálculo de erros relativos segundo a Equação (46).

No *Robot*, a estrutura é modelada em um plano vertical e restrita para comportamento plano, de forma compatível com o modelo bidimensional do programa. Adota-se o plano XZ no *Robot*, como o eixo Z sendo direção vertical, mantendo-se equivalência geométrica com o plano XY do programa.

6.4.3 Resultados e discussão

A Tabela 8 apresenta as três primeiras frequências naturais obtidas no *Robot* e no programa, calculados conforme a Equação (46). Os valores do programa são extraídos diretamente do arquivo de saída gerado pela rotina de análise modal.

Tabela 8 – Comparação dos valores de frequência

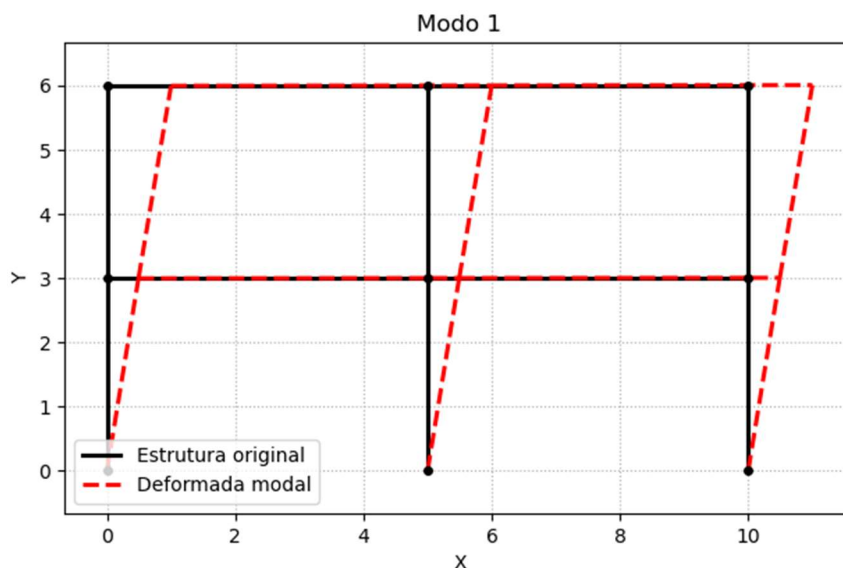
Modo j	f_{Robot} (Hz)	$f_{programa}$ (Hz)	ϵ_f (%)
1	7,45	7,5281	1,05
2	24,73	25,0117	1,14
3	40,98	41,5196	1,32

Fonte: Autor (2025)

As Figuras 16 a 21 apresentam os três primeiros modos do Exemplo 2 obtidos no programa e no *Robot*, para fins de comparação qualitativa. A leitura dessas figuras deve ser feita com base no padrão relativo do modo e na compatibilidade com as restrições do modelo, lembrando que as amplitudes são normalizadas e escaladas apenas para visualização.

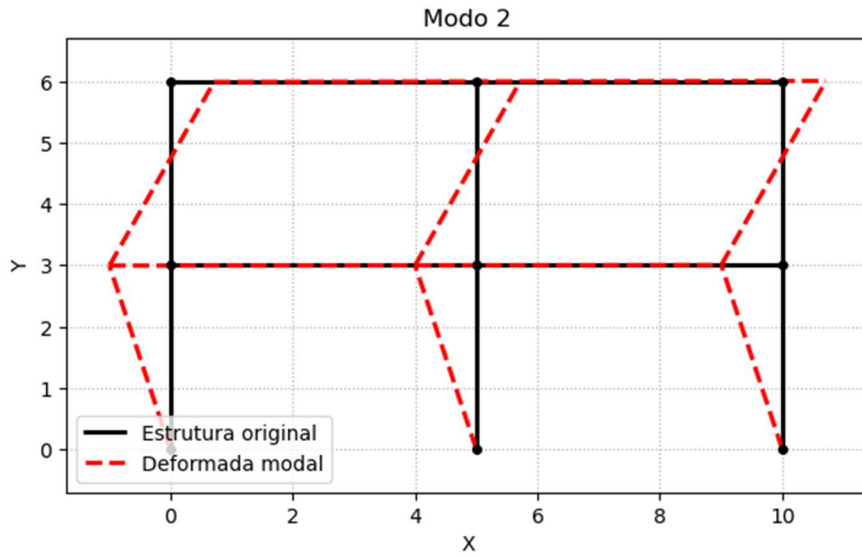
Conforme discutido no Exemplo 1, diferenças de suavização entre deformadas decorrem do pós-processamento, pois o *Robot* interpola a linha média ao longo do elemento, enquanto o programa traça a deformada por segmentos entre nós, não implicando necessariamente divergência numérica dos resultados modais

Figura 16 – Modo 1 do Exemplo 2, deformada modal no programa



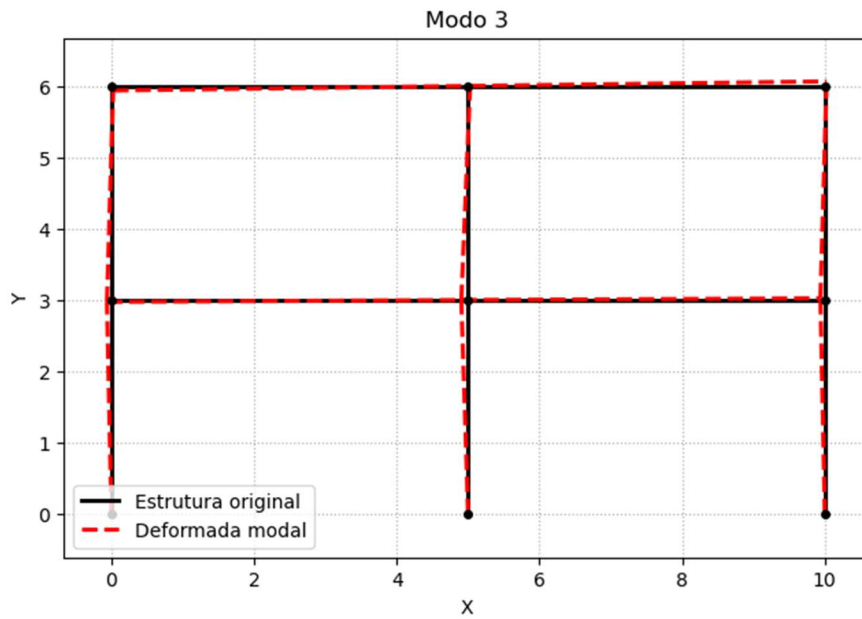
Fonte: Autor (2025)

Figura 17 – Modo 2 do Exemplo 2, deformada modal no programa



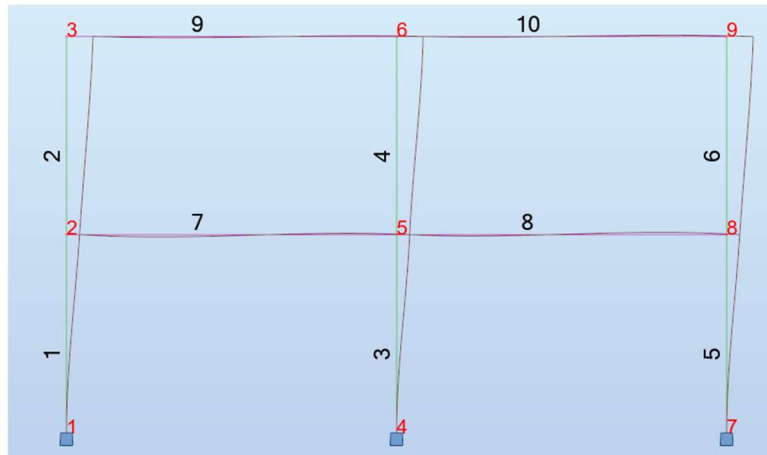
Fonte: Autor (2025)

Figura 18 – Modo 3 do Exemplo 2, deformada modal no programa



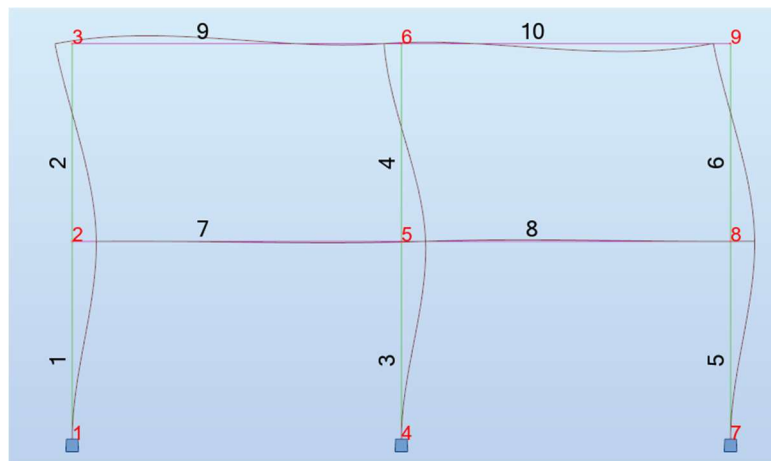
Fonte: Autor (2025)

Figura 19 – Modo 1 do Exemplo 2, deformada modal no *Robot*



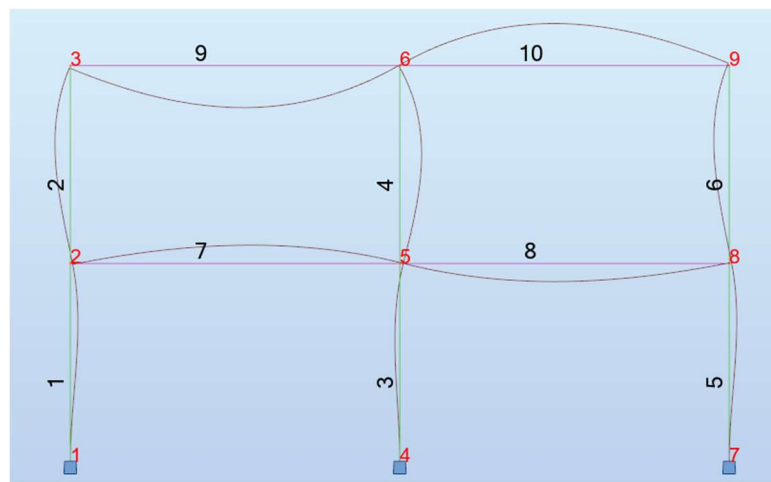
Fonte: Autor (2025)

Figura 20 – Modo 2 do Exemplo 2, deformada modal no *Robot*



Fonte: Autor (2025)

Figura 21 – Modo 3 do Exemplo 2, deformada modal no *Robot*



Fonte: Autor (2025)

Observa-se concordância consistente entre as frequências naturais, com erros relativos inferiores a 1,4% para os três primeiros modos. A ordem desses desvios permanece muito próxima a do Exemplo 1, o que reforça que a proximidade dos resultados não é circunstancial: o programa mantém aderência ao software comercial mesmo com aumento do número de elementos e graus de liberdade livres, e com redistribuição de rigidez e massa devido à presença do pilar central e do segundo vão.

7 CONSIDERAÇÕES FINAIS

7.1 CONCLUSÕES

O presente trabalho teve como objetivo desenvolver um programa computacional em linguagem *Python* para a análise modal de estruturas reticuladas planas, com ênfase na determinação das frequências naturais e das formas modais associadas, a partir das matrizes globais de rigidez e de massa do sistema.

Para isso, foi implementada a formulação do Método da Rigidez Direta aplicada ao elemento de pórtico plano, contemplando o cálculo das matrizes locais, a transformação para o sistema global e a montagem das matrizes $[K]$ e $[M]$. A entrada de dados foi padronizada em planilha eletrônica em formato *Excel* e a saída do programa é organizada em arquivos de resultados, de modo a facilitar a reprodutibilidade e a conferência das etapas numéricas.

A análise modal é conduzida pela resolução do problema generalizado de autovalores do sistema reduzido aos graus de liberdade livres, obtendo-se os autovalores e os autovetores necessários ao cálculo das frequências naturais e à reconstrução das formas modais no modelo completo.

A validação numérica foi realizada por comparação direta com o *Autodesk Robot Structural Analysis Professional*, adotado como referência. Foram analisados dois pórticos planos com discretizações compatíveis entre os ambientes: um pórtico com dois pavimentos e um vão e um pórtico com dois pavimentos e dois vãos. Dessa forma foi possível verificar o desempenho do programa em topologias com diferentes distribuições de rigidez e massa.

Em ambos os exemplos, as três primeiras frequências naturais apresentaram erros relativos baixos em relação ao *Robot*, com desvios da ordem de 1%. No Exemplo 1, os erros ficaram inferiores a 1,3% e, no Exemplo 2, inferiores a aproximadamente 1,4%. Esses resultados indicam consistência na implementação da montagem das matrizes, da aplicação das condições de contorno e da solução do problema modal. Assim, considera-se que os objetivos propostos foram atingidos útil tanto para fins didáticos quanto para estudos preliminares do comportamento dinâmico de estruturas planas.

Ressalta-se que os resultados devem ser interpretados dentro das hipóteses adotadas no desenvolvimento: comportamento linear elástico, pequenas deformações e vibração livre não amortecida, compatíveis com a formulação utilizada e com o escopo estabelecido para este trabalho. Adicionalmente, as formas modais foram comparadas de maneira qualitativa com o

Robot, observando-se configurações compatíveis entre os modos, sendo as diferenças visuais atribuídas predominantemente ao pós-processamento e à forma de representação das deformadas.

7.2 SUGESTÕES PARA FUTUROS TRABALHOS

Com base no desenvolvimento realizado e nas limitações assumidas, apresentam-se a seguir sugestões para a continuidade do trabalho e ampliação das funcionalidades do programa:

- Ampliação do conjunto de validações;
- Inclusão de um banco de seções transversais usuais e ferramentas para cálculo automático de propriedades geométricas a partir de parâmetros de seção;
- Expansão do programa para análise tridimensional, contemplando pórticos espaciais e grelhas, com inclusão dos graus de liberdade fora do plano, transformação completa entre sistemas locais e global e montagem das matrizes globais correspondentes;
- Ampliação do escopo de análise dinâmica para além da vibração livre não amortecida;
- Desenvolvimento de uma interface gráfica que organize a criação do modelo, a execução e a leitura dos resultados.

REFERÊNCIAS BIBLIOGRÁFICAS

ALMEIDA, E. D.; FONSACA, M. **Criação de código computacional didático para cálculo de estruturas reticuladas espaciais via método dos deslocamentos em formulação matricial (CERBERUS)**. 2018. Trabalho de Conclusão de Curso (Engenharia Civil) – Universidade Positivo, Curitiba, 2018.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 6118: Projeto de estruturas de concreto – Procedimento**. Rio de Janeiro, 2023.

CHOPRA, A. K. **Dynamics of structures: theory and applications to earthquake engineering**. 4 ed. Harlow: Pearson, 2014.

HIBBELER, R. C. **Structural analysis**. 9 ed. Boston: Pearson, 2015.

MARTHA, L. F. **Análise de estruturas: conceitos e métodos básicos**. Rio de Janeiro: Elsevier, 2010.

MATOS, D. B. M.; TITELLO, E. P. **Python aplicado à análise de estruturas**. Porto Alegre: UFRGS, 2020.

PRELINGER ARCHIVES. **Tacoma-Narrows Bridge collapse**. 1940. Disponível em: <<https://archive.org/details/Pa2096Tacoma>>. Acesso em: 13 dez. 2025.

RAO, S. S. **Mechanical vibrations**. 5 ed. Upper Saddle River: Prentice Hall, 2011.

SOARES, H. B. **Desenvolvimento de um programa para análise de pórticos tridimensionais considerando a flexibilidade das fundações**. 2017. Trabalho de Conclusão de Curso (Engenharia Civil) – Faculdade de Tecnologia, Universidade Federal do Amazonas, Manaus, 2017.

SORIANO, H. L. **Introdução à dinâmica das estruturas**. Rio de Janeiro: Elsevier, 2014.

PYTHON SOFTWARE FOUNDATION. **Python**. Wilmington: Python Software Foundation, 2025. Software. Disponível em: <<https://www.python.org>>. Acesso em: 13 dez. 2025.

APÊNDICE A – CÓDIGO EM *PYTHON* DESENVOLVIDO

Apresenta-se aqui o programa computacional desenvolvido no trabalho para implementação da teoria do Método da Rigidez e Análise Modal:

```

"""
Análise estática e modal de pórtico plano 2D
-----

Este módulo implementa:

- Leitura dos dados em formato Excel (abas "Nós" e "Barras");
- Montagem da matriz global de rigidez [K];
- Montagem da matriz global de massa [M] (massa consistente, rotacionada);
- Análise estática (vetor de deslocamentos globais {U}, reações de apoio e esforços de
extremidade);
- Análise modal (frequências naturais e modos de vibração, a partir de  $[K_{ll}]{\Phi_l} =
\omega^2[M_{ll}]{\Phi_l}$ );
- Exportação de resultados para arquivos .xlsx;
- Plotagem da estrutura original e dos primeiros modos de vibração.

Convenções de entrada (planilha Excel)
-----

Aba "Nós": colunas esperadas
- X, Y : coordenadas do nó
- RX, RY : restrições de deslocamento (1 = restrito, 0 = livre)
- Rθ : restrição de rotação (1 = restrito, 0 = livre)
- FX, FY : forças nodais aplicadas (opcional: se ausentes, assumem-se zero)
- MZ : momento nodal (opcional)

Aba "Barras": colunas esperadas
- N1, N2 : nós inicial e final
- A : área da seção transversal
- E : módulo de elasticidade
- I : momento de inércia fletor
- rho OU m_especifica : parâmetro de massa
  * Se for densidade volumétrica [kg/m³], usa-se  $m_{total} = \rho * A * L$ 
  * Se for massa linear [kg/m], basta ajustar o parâmetro mass_is_linear=True

O programa adapta-se automaticamente se a coluna se chamar "rho" ou "m_especifica".

Observação importante:
  A análise modal exige a biblioteca SciPy instalada no mesmo ambiente
  Python em que este módulo é executado (scipy.linalg.eigh).

```

```
"""

import sys
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# =====
# LEITURA E PRÉ-PROCESSAMENTO
# =====

def ler_dados_entrada(caminho_arquivo: str):
    """
    Lê o arquivo Excel com as abas "Nós" e "Barras" e ajusta índices
    para começarem em 1, compatíveis com a numeração de nós e barras
    usada no texto.
    """
    caminho_arquivo = caminho_arquivo.strip("'\"")

    nos = pd.read_Excel(caminho_arquivo, sheet_name="Nós")
    barras = pd.read_Excel(caminho_arquivo, sheet_name="Barras")

    # Índice começando em 1, compatível com a formulação teórica
    nos.index = nos.index + 1
    barras.index = barras.index + 1

    # Substituí NaN por zero
    nos = nos.fillna(0)
    barras = barras.fillna(0)

    return nos, barras

def calcular_geometria_barras(nos: pd.DataFrame, barras: pd.DataFrame):
    """
    Calcula comprimento L e cossenos diretores (c, s) para cada barra,
    conforme as equações de geometria do elemento.
    """
    Ls = []
    senos = []
    cossenos = []

    for barra in barras.index:
        n1 = int(barras.loc[barra, "N1"])
        n2 = int(barras.loc[barra, "N2"])

        x1, y1 = nos.loc[n1, ["X", "Y"]]
        x2, y2 = nos.loc[n2, ["X", "Y"]]

        dx = x2 - x1
        dy = y2 - y1
```

```

L = float((dx**2 + dy**2) ** 0.5)
if L == 0.0:
    raise ValueError(f"Barra {barra} com comprimento zero (verifique N1/N2).")

c = dx / L
s = dy / L

Ls.append(L)
cossenos.append(c)
senos.append(s)

barras = barras.copy()
barras["L"] = Ls
barras["cos"] = cossenos
barras["sin"] = senos

return barras

# =====
# MATRIZES LOCAIS E ROTAÇÃO
# =====

def matriz_rigidez_local_portico(E: float, A: float, I: float, L: float) -> np.ndarray:
    """
    Matriz de rigidez local [k]^1 (6x6) de barra de pórtico plano,
    no sistema LOCAL [u_i, v_i, θ_i, u_j, v_j, θ_j].

    Forma clássica para barra de pórtico Euler-Bernoulli.
    """
    k_ax = E * A / L
    k_flex = E * I / (L**3)

    k_l = np.array([
        [ k_ax,      0.0,      0.0,    -k_ax,      0.0,      0.0],
        [ 0.0, 12.0*k_flex,  6.0*L*k_flex,  0.0, -12.0*k_flex,  6.0*L*k_flex],
        [ 0.0,  6.0*L*k_flex, 4.0*L**2*k_flex, 0.0, -6.0*L*k_flex, 2.0*L**2*k_flex],
        [-k_ax,      0.0,      0.0,     k_ax,      0.0,      0.0],
        [ 0.0, -12.0*k_flex, -6.0*L*k_flex,  0.0, 12.0*k_flex, -6.0*L*k_flex],
        [ 0.0,  6.0*L*k_flex, 2.0*L**2*k_flex, 0.0, -6.0*L*k_flex, 4.0*L**2*k_flex],
    ], dtype=float)
    return k_l

def matriz_massa_local_consistente(massa_param: float, A: float, L: float,
                                   mass_is_linear: bool = False) -> np.ndarray:
    """
    Matriz de massa local consistente [M]^1 (6x6) para pórtico plano,
    no sistema LOCAL [u_i, v_i, θ_i, u_j, v_j, θ_j].

    - Se mass_is_linear = False: 'massa_param' é densidade volumétrica ρ [kg/m³]
      e a massa linear é m' = ρ * A [kg/m].
    """

```

- Se `mass_is_linear = True` : 'massa_param' é diretamente a massa linear m' [kg/m].

A matriz é escrita na forma clássica proporcional a $(m' * L / 420)$.

```
"""
if mass_is_linear:
    m_linha = massa_param          # m' [kg/m]
else:
    m_linha = massa_param * A      # m' = ρ A

coef = (m_linha * L) / 420.0
M_l = coef * np.array([
    [140.0,  0.0,  0.0,  70.0,  0.0,  0.0],
    [ 0.0, 156.0, 22.0*L,  0.0,  54.0, -13.0*L],
    [ 0.0, 22.0*L, 4.0*L**2,  0.0, 13.0*L, -3.0*L**2],
    [ 70.0,  0.0,  0.0, 140.0,  0.0,  0.0],
    [ 0.0,  54.0, 13.0*L,  0.0, 156.0, -22.0*L],
    [ 0.0, -13.0*L, -3.0*L**2,  0.0, -22.0*L,  4.0*L**2],
], dtype=float)
return M_l
```

```
def matriz_rotacao_portico(c: float, s: float) -> np.ndarray:
    """
    Matriz de transformação [T] (6x6) para pórtico plano.
    Transforma vetores [u, v, θ] do sistema LOCAL para o GLOBAL.

    {u}^1 = [T]{u}^g
    """
    R = np.array([
        [ c, -s, 0.0],
        [ s,  c, 0.0],
        [0.0, 0.0, 1.0]
    ], dtype=float)

    T = np.zeros((6, 6), dtype=float)
    T[0:3, 0:3] = R
    T[3:6, 3:6] = R
    return T
```

```
# =====
# MONTAGEM GLOBAL [K] E [M]
# =====
```

```
def montar_matrizes_globais(nos: pd.DataFrame, barras: pd.DataFrame,
                             mass_is_linear: bool = False):
    """
    Monta as matrizes globais de rigidez [K] e de massa [M] (sistema GLOBAL),
    com rotação das matrizes locais  $[k]^1$  e  $[M]^1$  por meio de  $[T]^T[k]^1[T]$ 
    e  $[T]^T[M]^1[T]$ .
```

- `mass_is_linear`: se `True`, considera que a coluna de massa é massa linear m' [kg/m];

```

        se False, considera densidade volumétrica  $\rho$  [kg/m3].
"""
n_nos = len(nos.index)
n_gl = 3 * n_nos
K = np.zeros((n_gl, n_gl), dtype=float)
M = np.zeros((n_gl, n_gl), dtype=float)

# Identifica a coluna de massa na planilha de barras
if "rho" in barras.columns:
    nome_col_massa = "rho"
elif "m_especifica" in barras.columns:
    nome_col_massa = "m_especifica"
else:
    raise KeyError(
        "Nenhuma coluna de massa encontrada em 'Barras'. "
        "Crie uma coluna 'rho' ou 'm_especifica' com a massa específica "
        "(volumétrica ou linear)."
    )

for barra in barras.index:
    n1 = int(barras.loc[barra, "N1"])
    n2 = int(barras.loc[barra, "N2"])

    E = float(barras.loc[barra, "E"])
    A = float(barras.loc[barra, "A"])
    I = float(barras.loc[barra, "I"])
    L = float(barras.loc[barra, "L"])
    c = float(barras.loc[barra, "cos"])
    s = float(barras.loc[barra, "sin"])

    massa_param = float(barras.loc[barra, nome_col_massa])

    # Matrizes locais
    K_l = matriz_rigidez_local_portico(E, A, I, L)
    M_l = matriz_massa_local_consistente(
        massa_param, A, L, mass_is_linear=mass_is_linear
    )

    # Rotação para o sistema global:  $[k]^g = [T]^T [k]^l [T]$ 
    T = matriz_rotacao_portico(c, s)
    K_g = T.T @ K_l @ T
    M_g = T.T @ M_l @ T

    # Graus de liberdade globais (0-based)
    g11 = 3*(n1-1) + 0
    g12 = 3*(n1-1) + 1
    g13 = 3*(n1-1) + 2
    g14 = 3*(n2-1) + 0
    g15 = 3*(n2-1) + 1
    g16 = 3*(n2-1) + 2

```

```

        idx = np.array([g11, g12, g13, g14, g15, g16], dtype=int)

        # Montagem por superposição (forma padrão do MRD)
        K[np.ix_(idx, idx)] += K_g
        M[np.ix_(idx, idx)] += M_g

    return K, M

# =====
# CONDIÇÕES DE CONTORNO
# =====

def obter_graus_livres(nos: pd.DataFrame):
    """
    Determina os graus de liberdade livres e restritos a partir de RX, RY, Rθ.

    Retorna:
        gdl_livres    - índices dos graus de liberdade livres
        gdl_restritos - índices dos graus de liberdade restritos
    """
    n_nos = len(nos.index)
    n_gl = 3 * n_nos

    gdl_restritos = []
    for no in nos.index:
        RX = int(nos.loc[no, "RX"])
        RY = int(nos.loc[no, "RY"])
        Rth = int(nos.loc[no, "Rθ"])

        if RX == 1:
            gdl_restritos.append(3*(no-1) + 0) # u_x
        if RY == 1:
            gdl_restritos.append(3*(no-1) + 1) # u_y
        if Rth == 1:
            gdl_restritos.append(3*(no-1) + 2) # θ_z

    gdl_restritos = np.array(sorted(set(gdl_restritos)), dtype=int)
    gdl_todos = np.arange(n_gl, dtype=int)
    gdl_livres = np.setdiff1d(gdl_todos, gdl_restritos)

    return gdl_livres, gdl_restritos

# =====
# ANÁLISE ESTÁTICA
# =====

def montar_vetor_forcas(nos: pd.DataFrame):
    """
    Monta o vetor global de forças nodais {F} (tamanho 3*n_nos).

```

```

Usa colunas FX, FY e, se existir, MZ.
"""
n_nos = len(nos.index)
n_gl = 3 * n_nos
F = np.zeros(n_gl, dtype=float)

tem_FX = "FX" in nos.columns
tem_FY = "FY" in nos.columns
tem_MZ = "MZ" in nos.columns

for no in nos.index:
    FX = float(nos.loc[no, "FX"]) if tem_FX else 0.0
    FY = float(nos.loc[no, "FY"]) if tem_FY else 0.0
    MZ = float(nos.loc[no, "MZ"]) if tem_MZ else 0.0

    gl_u = 3*(no-1) + 0
    gl_v = 3*(no-1) + 1
    gl_t = 3*(no-1) + 2

    F[gl_u] += FX
    F[gl_v] += FY
    F[gl_t] += MZ

return F

def resolver_estatico(K: np.ndarray, F: np.ndarray,
                    gdl_livres: np.ndarray, gdl_restritos: np.ndarray):
    """
    Resolve o problema estático global

     $[K]\{U\} = \{F\}$ 

    impondo  $\{U_r\} = 0$  nos graus de liberdade restritos (apoios indeformáveis).

    Em termos de submatrizes associadas aos graus de liberdade livres:

     $[K_{11}]\{U_1\} = \{F_1\}$ 

    em que  $[K_{11}]$  e  $\{F_1\}$  são obtidos por seleção de linhas e colunas de  $[K]$  e  $\{F\}$ .
    """
    n_gl = K.shape[0]

    # Submatriz e vetor dos GL livres
    K_11 = K[np.ix_(gdl_livres, gdl_livres)]
    F_1 = F[gdl_livres]

    # Deslocamentos nos GL livres
    U_1 = np.linalg.solve(K_11, F_1)

    # Reconstrói vetor de deslocamentos completo  $\{U\}$ 

```

```

U = np.zeros(n_gl, dtype=float)
U[gdl_livres] = U_l
U[gdl_restritos] = 0.0 # apoios indeformáveis

# Reações de apoio {R} = [K]{U} - {F}
R = K @ U - F

return U, R

def calcular_esforços_elementares(U: np.ndarray,
                                  nos: pd.DataFrame,
                                  barras: pd.DataFrame):
    """
    Calcula esforços nodais de extremidade em cada barra no sistema LOCAL,
    usando  $\{q\}^1 = [k]^1 \{d\}^1$ .

    Retorna DataFrame com N1x, V1, M1, N2x, V2, M2 por barra.
    """
    resultados = []

    for barra in barras.index:
        n1 = int(barras.loc[barra, "N1"])
        n2 = int(barras.loc[barra, "N2"])

        E = float(barras.loc[barra, "E"])
        A = float(barras.loc[barra, "A"])
        I = float(barras.loc[barra, "I"])
        L = float(barras.loc[barra, "L"])
        c = float(barras.loc[barra, "cos"])
        s = float(barras.loc[barra, "sin"])

        # Graus globais da barra (0-based)
        gl1 = 3*(n1-1) + 0
        gl2 = 3*(n1-1) + 1
        gl3 = 3*(n1-1) + 2
        gl4 = 3*(n2-1) + 0
        gl5 = 3*(n2-1) + 1
        gl6 = 3*(n2-1) + 2
        idx = np.array([gl1, gl2, gl3, gl4, gl5, gl6], dtype=int)

        # Deslocamentos globais da barra
        d_g = U[idx]

        # Transforma para sistema local
        T = matriz_rotacao_portico(c, s)
        d_l = T @ d_g

        # Esforços nodais locais
        K_l = matriz_rigidez_local_portico(E, A, I, L)
        q_l = K_l @ d_l # [N_i, V_i, M_i, N_j, V_j, M_j]

```

```

    resultados.append({
        "Barra": barra,
        "N1": n1,
        "N2": n2,
        "N1x": q_1[0],
        "V1": q_1[1],
        "M1": q_1[2],
        "N2x": q_1[3],
        "V2": q_1[4],
        "M2": q_1[5],
    })

df_esforços = pd.DataFrame(resultados)
return df_esforços

def exportar_resultados_estatica(nos: pd.DataFrame,
                                U: np.ndarray,
                                R: np.ndarray,
                                df_esforços: pd.DataFrame,
                                caminho_saida: str = "resultados_estatica.xlsx"):
    """
    Exporta deslocamentos nodais {U}, reações de apoio {R} e esforços de extremidade
    em barras para um arquivo Excel.
    """
    n_nos = len(nos.index)

    # Deslocamentos por nó
    dados_desloc = []
    for no in nos.index:
        u = U[3*(no-1) + 0]
        v = U[3*(no-1) + 1]
        t = U[3*(no-1) + 2]
        dados_desloc.append({
            "No": no,
            "ux": u,
            "uy": v,
            "theta": t
        })
    df_desloc = pd.DataFrame(dados_desloc)

    # Reações por nó
    dados_reac = []
    for no in nos.index:
        RX = int(nos.loc[no, "RX"])
        RY = int(nos.loc[no, "RY"])
        Rth = int(nos.loc[no, "Rθ"])

        rux = R[3*(no-1) + 0] if RX == 1 else 0.0
        ruy = R[3*(no-1) + 1] if RY == 1 else 0.0

```

```

rmz = R[3*(no-1) + 2] if Rth == 1 else 0.0

dados_reac.append({
    "No": no,
    "RX": RX,
    "RY": RY,
    "Rθ": Rth,
    "Reac_ux": rux,
    "Reac_uy": ruy,
    "Reac_mz": rmz
})
df_reac = pd.DataFrame(dados_reac)

with pd.ExcelWriter(caminho_saida) as writer:
    df_desloc.to_Excel(writer, sheet_name="Deslocamentos", index=False)
    df_reac.to_Excel(writer, sheet_name="Reacoes", index=False)
    df_esforços.to_Excel(writer, sheet_name="Esforços_barras", index=False)

print(f"[Estatica] Resultados salvos em '{caminho_saida}'.")

# =====
# ANÁLISE MODAL
# =====

def resolver_problema_modal(K: np.ndarray, M: np.ndarray,
                           gdl_livres: np.ndarray):
    """
    Resolve o problema generalizado de autovalor restrito aos graus de liberdade livres:

     $[K_{ll}]\{\Phi_l\} = \omega^2 [M_{ll}]\{\Phi_l\}$ 

    com
     $[K_{ll}] = [K]$  reduzida aos GL livres,
     $[M_{ll}] = [M]$  reduzida aos GL livres.

    Retorna:
    autovalores ->  $\lambda_j = \omega_j^2$ 
    omega        ->  $\omega_j$  [rad/s]
    freq         ->  $f_j$  [Hz]
    autovetores ->  $\{\Phi_l\}$  escritos apenas nos GL livres.
    """
    from math import pi
    try:
        from scipy.linalg import eigh
    except ImportError as e:
        raise ImportError(
            "SciPy não está disponível neste ambiente. "
            "A análise modal requer scipy.linalg.eigh. "
            "Instale o pacote 'scipy' no mesmo Python usado para executar este módulo."
        ) from e

```

```

# Submatrizes associadas aos GL livres
K_ll = K[np.ix_(gdl_livres, gdl_livres)]
M_ll = M[np.ix_(gdl_livres, gdl_livres)]

# Resolve o problema generalizado simétrico
autovalores, autovetores = eigh(K_ll, M_ll)

# Filtra autovalores positivos
tol = 1e-8
mask = autovalores > tol
autovalores = autovalores[mask]
autovetores = autovetores[:, mask]

# Ordena em ordem crescente
idx_sort = np.argsort(autovalores)
autovalores = autovalores[idx_sort]
autovetores = autovetores[:, idx_sort]

# Normaliza autovetores para visualização (máximo absoluto igual a 1)
for j in range(autovetores.shape[1]):
    norma = np.max(np.abs(autovetores[:, j]))
    if norma > 0.0:
        autovetores[:, j] /= norma

# Frequências
omega = np.sqrt(autovalores)      # rad/s
freq = omega / (2.0 * pi)        # Hz

return autovalores, omega, freq, autovetores

def reconstruir_modos_completos(autovetores_red: np.ndarray,
                                gdl_livres: np.ndarray,
                                n_gl: int):
    """
    Reconstrói os vetores de modos completos {} incluindo zeros nos GL restritos.
    """
    n_modos = autovetores_red.shape[1]
    modos_completos = np.zeros((n_gl, n_modos), dtype=float)

    for j in range(n_modos):
        modos_completos[gdl_livres, j] = autovetores_red[:, j]

    return modos_completos

def exportar_resultados_modal(omega, freq, modos_completos,
                              caminho_saida: str = "resultados_modal.xlsx"):
    """
    Exporta frequências naturais e modos de vibração completos para Excel.
    """

```

```

n_gl, n_modos = modos_completos.shape

df_freq = pd.DataFrame({
    "Modo": np.arange(1, n_modos+1),
    "omega_rad_s": omega,
    "f_Hz": freq
})

df_modos = pd.DataFrame(
    modos_completos,
    index=[f"GL_{i+1}" for i in range(n_gl)],
    columns=[f"Modo_{j+1}" for j in range(n_modos)]
)

with pd.ExcelWriter(caminho_saida) as writer:
    df_freq.to_Excel(writer, sheet_name="Frequencias", index=False)
    df_modos.to_Excel(writer, sheet_name="Modos", index=True)

print(f"[Modal] Resultados salvos em '{caminho_saida}'.")

# =====
# FUNÇÕES AUXILIARES PARA PLOTAGEM
# =====

def limites_estrutura(nos: pd.DataFrame, margem_relativa: float = 0.05):
    """
    Determina limites de plotagem (xmin, xmax, ymin, ymax) a partir das
    coordenadas dos nós, acrescentando uma margem relativa em torno da
    estrutura para melhorar a visualização.
    """
    x_min = float(nos["X"].min())
    x_max = float(nos["X"].max())
    y_min = float(nos["Y"].min())
    y_max = float(nos["Y"].max())

    dx = x_max - x_min
    dy = y_max - y_min

    # Evita intervalos nulos (estrutura alinhada em linha)
    if dx <= 0.0:
        dx = 1.0
        x_min -= 0.5
        x_max += 0.5
    if dy <= 0.0:
        dy = 1.0
        y_min -= 0.5
        y_max += 0.5

    d_ref = max(dx, dy)
    margem = margem_relativa * d_ref

```

```

return (x_min - margem,
        x_max + margem,
        y_min - margem,
        y_max + margem,
        dx, dy)

def escala_modal_automatica(nos: pd.DataFrame,
                            D: np.ndarray,
                            frac_altura: float = 0.10) -> float:
    """
    Calcula automaticamente um fator de escala para a deformada modal,
    de modo que o deslocamento máximo represente aproximadamente uma
    fração 'frac_altura' da dimensão dominante da estrutura.

    Caso frac_altura seja muito pequeno ou os deslocamentos sejam nulos,
    retorna 1.0.
    """
    x_min = float(nos["X"].min())
    x_max = float(nos["X"].max())
    y_min = float(nos["Y"].min())
    y_max = float(nos["Y"].max())

    d_ref = max(x_max - x_min, y_max - y_min, 1.0)

    max_amp = 0.0
    for no in nos.index:
        ux = float(D[3*(no-1) + 0])
        uy = float(D[3*(no-1) + 1])
        amp = (ux**2 + uy**2) ** 0.5
        if amp > max_amp:
            max_amp = amp

    if max_amp == 0.0:
        return 1.0

    return frac_altura * d_ref / max_amp

def _coords_deformados_nodais(nos: pd.DataFrame,
                              D: np.ndarray,
                              escala_local: float):
    """
    Gera coordenadas deformadas nodais a partir do vetor modal completo {D},
    aplicando apenas translações (ux, uy) nos nós.

    Retorna:
        x_def: dict {no: x_deformado}
        y_def: dict {no: y_deformado}
    """
    x_def = {}

```

```
y_def = {}

for no in nos.index:
    x, y = nos.loc[no, ["X", "Y"]]
    ux = float(D[3*(no-1) + 0])
    uy = float(D[3*(no-1) + 1])

    x_def[no] = float(x) + escala_local * ux
    y_def[no] = float(y) + escala_local * uy

return x_def, y_def

def _limites_com_deformada(nos: pd.DataFrame,
                           x_def: dict,
                           y_def: dict,
                           margem_relativa: float = 0.05):
    """
    Calcula limites de plotagem considerando a estrutura original e a deformada,
    evitando corte da deformada no gráfico.
    """
    x0_min = float(nos["X"].min())
    x0_max = float(nos["X"].max())
    y0_min = float(nos["Y"].min())
    y0_max = float(nos["Y"].max())

    xd_vals = np.array(list(x_def.values()), dtype=float)
    yd_vals = np.array(list(y_def.values()), dtype=float)

    x_min = min(x0_min, float(xd_vals.min()))
    x_max = max(x0_max, float(xd_vals.max()))
    y_min = min(y0_min, float(yd_vals.min()))
    y_max = max(y0_max, float(yd_vals.max()))

    dx = x_max - x_min
    dy = y_max - y_min

    # Evita intervalos nulos (casos degenerados)
    if dx <= 0.0:
        dx = 1.0
        x_min -= 0.5
        x_max += 0.5
    if dy <= 0.0:
        dy = 1.0
        y_min -= 0.5
        y_max += 0.5

    d_ref = max(dx, dy)
    margem = margem_relativa * d_ref

    return (x_min - margem,
```

```

        x_max + margem,
        y_min - margem,
        y_max + margem,
        dx, dy)

# =====
# PLOTAGEM DOS MODOS
# =====

def plotar_modos(nos: pd.DataFrame,
                barras: pd.DataFrame,
                modos_completos: np.ndarray,
                escala: float = 1.0,
                n_modos_plotar: int = 3,
                usar_escala_automatica: bool = False,
                freq: np.ndarray = None,
                mostrar_nos: bool = True,
                mostrar_ids_nos: bool = False,
                salvar_em: str = None,
                dpi_png: int = 300,
                largura_base: float = 7.5,
                margem_relativa: float = 0.06):
    """
    Plota a estrutura original e os primeiros modos de vibração.

    Representação adotada:
    - A deformada modal é gerada aplicando o vetor modal nos nós
      (ux, uy) e conectando os nós deformados por segmentos retos.

    Parâmetros:
        escala                : fator de amplificação da deformada modal
                               (usado se usar_escala_automatica=False).
        n_modos_plotar        : número máximo de modos a serem plotados.
        usar_escala_automatica: se True, a escala é calculada por
                               'escala_modal_automatica' para cada modo.
        freq                  : vetor de frequências [Hz] (opcional) para título.
        mostrar_nos           : se True, marca nós com pontos.
        mostrar_ids_nos       : se True, escreve o id de cada nó.
        salvar_em               : pasta para salvar SVG e PNG (opcional).
        dpi_png                : resolução do PNG (se salvar_em não for None).
        largura_base           : largura da figura em polegadas.
        margem_relativa        : margem relativa para limites de plotagem.
    """
    import os

    _, n_modos = modos_completos.shape
    n_plot = min(n_modos_plotar, n_modos)

    if salvar_em is not None:
        os.makedirs(salvar_em, exist_ok=True)

```

```
for j in range(n_plot):
    D = modos_completos[:, j].astype(float)

    if usar_escala_automatica:
        escala_local = escala_modal_automatica(nos, D)
    else:
        escala_local = float(escala)

    # Coordenadas deformadas nodais
    x_def, y_def = _coords_deformados_nodais(nos, D, escala_local)

    # Limites considerando a deformada (evita corte)
    x_min, x_max, y_min, y_max, dx, dy = _limites_com_deformada(
        nos, x_def, y_def, margem_relativa=margem_relativa
    )

    # Tamanho da figura proporcional à geometria
    razao = dy / dx
    altura = max(3.8, min(9.0, largura_base * razao))

    fig, ax = plt.subplots(figsize=(largura_base, altura), constrained_layout=True)

    # Estrutura original
    for barra in barras.index:
        n1 = int(barras.loc[barra, "N1"])
        n2 = int(barras.loc[barra, "N2"])

        x1, y1 = nos.loc[n1, ["X", "Y"]]
        x2, y2 = nos.loc[n2, ["X", "Y"]]

        label_original = "Estrutura original" if barra == barras.index[0] else ""
        ax.plot([x1, x2], [y1, y2],
                "k-", linewidth=2.2, label=label_original)

    # Estrutura deformada (modo j) conectando nós deformados
    for barra in barras.index:
        n1 = int(barras.loc[barra, "N1"])
        n2 = int(barras.loc[barra, "N2"])

        x1_def = x_def[n1]
        y1_def = y_def[n1]
        x2_def = x_def[n2]
        y2_def = y_def[n2]

        label_def = "Deformada modal" if barra == barras.index[0] else ""
        ax.plot([x1_def, x2_def], [y1_def, y2_def],
                "r--", linewidth=2.2, label=label_def)

    # Nós e ids (opcional)
```

```

if mostrar_nos or mostrar_ids_nos:
    for no in nos.index:
        x, y = nos.loc[no, ["X", "Y"]]
        if mostrar_nos:
            ax.plot([x], [y], "ko", markersize=4)
        if mostrar_ids_nos:
            ax.text(float(x), float(y), f"{no}", fontsize=8,
                    ha="center", va="bottom",
                    bbox=dict(boxstyle="round,pad=0.15",
                              fc="w", ec="none", alpha=0.75))

ax.set_xlim(x_min, x_max)
ax.set_ylim(y_min, y_max)
ax.set_aspect("equal", adjustable="box")
ax.grid(True, linestyle=":", linewidth=0.8)

ax.set_xlabel("X")
ax.set_ylabel("Y")

if freq is not None and j < len(freq):
    ax.set_title(f"Modo {j+1} | f = {float(freq[j]):.4f} Hz")
else:
    ax.set_title(f"Modo {j+1}")

ax.legend(loc="lower left", frameon=True)

if salvar_em is not None:
    nome_svg = os.path.join(salvar_em, f"modo_{j+1:02d}.svg")
    nome_png = os.path.join(salvar_em, f"modo_{j+1:02d}.png")
    fig.savefig(nome_svg, bbox_inches="tight", pad_inches=0.05)
    fig.savefig(nome_png, dpi=int(dpi_png), bbox_inches="tight", pad_inches=0.05)

plt.show()

# =====
# FUNÇÃO PRINCIPAL (INTEGRA ESTÁTICA E MODAL)
# =====

def executar_analise(caminho_Excel: str,
                    mass_is_linear: bool = False,
                    arquivo_estatica: str = "resultados_estatica.xlsx",
                    arquivo_modal: str = "resultados_modal.xlsx",
                    plotar_modos_flag: bool = True):
    """
    Executa a análise estática e modal a partir de um arquivo Excel de entrada.

    Parâmetros:
        caminho_Excel : caminho para a planilha de entrada.
        mass_is_linear : True se o parâmetro de massa na planilha for massa linear m' [kg/m];
                       False se for densidade volumétrica ρ [kg/m³].
    """

```

```

    arquivo_estatica : nome do arquivo .xlsx para salvar resultados estáticos.
    arquivo_modal     : nome do arquivo .xlsx para salvar resultados modais.
    plotar_modos_flag: se True, plota os primeiros modos de vibração.
"""
# 1. Leitura dos dados
nos, barras = ler_dados_entrada(caminho_Excel)
barras = calcular_geometria_barras(nos, barras)

# 2. Montagem de [K] e [M]
K, M = montar_matrizes_globais(nos, barras, mass_is_linear=mass_is_linear)

# 3. Determinação dos GL livres e restritos
gdl_livres, gdl_restritos = obter_graus_livres(nos)
n_gl = K.shape[0]

# -----
# ANÁLISE ESTÁTICA
# -----
F = montar_vetor_forcas(nos)
U, R = resolver_estatico(K, F, gdl_livres, gdl_restritos)
df_esforços = calcular_esforços_elementares(U, nos, barras)
exportar_resultados_estatica(nos, U, R, df_esforços,
                             caminho_saida=arquivo_estatica)

# -----
# ANÁLISE MODAL
# -----
autovalores, omega, freq, autovetores_red = resolver_problema_modal(
    K, M, gdl_livres
)
modos_completos = reconstruir_modos_completos(
    autovetores_red, gdl_livres, n_gl
)
exportar_resultados_modal(omega, freq, modos_completos,
                          caminho_saida=arquivo_modal)

print("\nFrequências naturais [Hz]:")
for i, f in enumerate(freq, start=1):
    print(f"Modo {i:2d}: f = {f:.4f} Hz (omega = {omega[i-1]:.4f} rad/s)")

# Plotar os primeiros modos (se desejado)
if plotar_modos_flag:
    try:
        plotar_modos(nos, barras, modos_completos,
                     escala=1.0,
                     n_modos_plotar=min(3, len(freq)),
                     usar_escala_automatica=False)
    except Exception as e:
        print("Falha ao plotar modos (verifique matplotlib):", e)

```

```
print("Módulo carregado. Para executar a análise, chame por exemplo:\n")
print("executar_analise(r'Caminho/para/seu_arquivo.xlsx', mass_is_linear=False)")
print(f"[Info] Python em uso: {sys.executable}")

caminho_Excel = input("Informe o caminho completo do arquivo Excel (.xlsx):
").strip().strip("\n")
if not caminho_Excel:
    raise ValueError("Caminho do arquivo não fornecido.")
executar_analise(caminho_Excel)
```