

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

FELIPE CAMARGO DE OLIVEIRA KOBAYASHI

**DESENVOLVIMENTO DE CÓDIGO COMPUTACIONAL PARA ANÁLISE MODAL
DE TRELIÇAS PLANAS VIA MÉTODO DOS ELEMENTOS FINITOS**

CAMPO MOURÃO

2022

FELIPE CAMARGO DE OLIVEIRA KOBAYASHI

**DESENVOLVIMENTO DE CÓDIGO COMPUTACIONAL PARA ANÁLISE MODAL
DE TRELIÇAS PLANAS VIA MÉTODO DOS ELEMENTOS FINITOS**

**Development of computer code for modal analysis of plane trusses via Finite
Element Method**

Trabalho de Conclusão de Curso de Graduação
apresentada como requisito para obtenção do título de
Bacharel em Engenharia Civil da Universidade
Tecnológica Federal do Paraná (UTFPR).
Orientador(a): Prof. Dr. Leandro Waidemam.

CAMPO MOURÃO

2022



Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es). Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

FELIPE CAMARGO DE OLIVEIRA KOBAYASHI

**DESENVOLVIMENTO DE CÓDIGO COMPUTACIONAL PARA ANÁLISE MODAL
DE TRELIÇAS PLANAS VIA MÉTODO DOS ELEMENTOS FINITOS**

Trabalho de Conclusão de Curso de Graduação
apresentado como requisito para obtenção do título de
Bacharel em Engenharia Civil da Universidade
Tecnológica Federal do Paraná (UTFPR).

Data de aprovação: 29 de novembro de 2022

Leandro Waidemam
Doutorado
Universidade Tecnológica Federal do Paraná

Marcelo Rodrigo Carreira
Doutorado
Universidade Tecnológica Federal do Paraná

Ronaldo Rigobello
Doutorado
Universidade Tecnológica Federal do Paraná

CAMPO MOURÃO

2022

Dedico este trabalho à minha esposa, aos meus avós e minha mãe por todo o apoio que tive durante a graduação.

AGRADECIMENTOS

Gostaria de agradecer primeiramente a Deus por tudo o que tem feito em minha vida e por tudo que ainda fará. Sem Ele não sou nada.

Ao amor da minha vida, minha amada esposa, Endoli, que sempre acreditou em mim, mesmo nos momentos mais difíceis em que ninguém mais acreditava. Seu amor incondicional e todo seu companheirismo foram os principais combustíveis para mim em meio a todas as dificuldades. E por fim, ainda concebeu nosso maior presente de Deus, que é nossa filha Sophia.

À minha mãe, Rosangela, e meus avós maternos, Sebastiana e Miguel, que durante toda minha vida nunca me desampararam e sempre fizeram o máximo para que eu me desenvolvesse. E que ainda acolheram minhas amadas como se fossem suas filhas, e nos proporcionaram um lar e todo suporte durante minha graduação.

A todos os meus professores, mas em especial ao meu orientador Prof. Dr. Leandro Waidemam por todos os ensinamentos, paciência e dedicação durante todo esse trabalho. Sua contribuição ao longo de todo o curso foi essencial à minha formação, serei eternamente grato por tudo.

Aos pastores e membros de minha igreja que de alguma forma contribuíram para meu crescimento.

Changing an organization, a company, a country -
or a world - begins with the simple step of
changing yourself.
(Anthony Robbins)

RESUMO

Este trabalho tem como finalidade apresentar um código computacional capaz de realizar a análise modal de treliças planas, fornecendo as frequências naturais e os modos de vibração, que são elementos fundamentais no estudo da Dinâmica das Estruturas. Inicialmente, foram apresentadas as deduções das equações matemáticas necessárias para o desenvolvimento do programa. O código apresentado é baseado no Método dos Elementos Finitos (MEF), e para sua concepção foi utilizado o software MATLAB. Por meio do ambiente App Designer, foi desenvolvida uma interface gráfica intuitiva para facilitar a inserção de dados por parte do usuário. Para a validação do código desenvolvido, foram analisadas três treliças planas bastante comuns: Howe, Pratt e Warren. Para efeitos de comparação, fez-se uso do software Ansys Mechanical APDL 2022 R2 versão Academic Research, que é um dos softwares de Elementos Finitos mais utilizados globalmente. A partir dos resultados obtidos pelo código elaborado e pelo software comercial utilizado, é possível constatar que as frequências naturais e os modos de vibração (sendo os modos de vibrar avaliados de forma qualitativa) convergem, comprovando, dessa forma, a eficácia do código implementado.

Palavras-chave: análise modal; código computacional; Método dos Elementos Finitos; treliças planas.

ABSTRACT

This work aims to present a computational code capable of performing the modal analysis of plane trusses, providing the natural frequencies and vibration modes, which are essential elements in the Structural Dynamics field. Firstly, the equations required for the development of the computer program were presented. The introduced computer code is based on the Finite Element Method (FEM), and the MATLAB software was used for its design. Through the App Designer environment, an interface was developed to make it easier to insert all the data. To validate the results from the developed code, three very common plane trusses were chosen: Howe, Pratt and Warren. For comparison purposes, the Ansys Mechanical APDL 2022 R2 Academic Research was chosen, which is one of the most well-known Finite Elements Analysis software. From the results obtained by the developed computer program and Ansys Mechanical APDL, it can be state that all the results converge, proving the accuracy of the developed computer code.

Keywords: modal analysis; computer code; Finite Element Method; plane trusses.

LISTA DE ILUSTRAÇÕES

Figura 1 - Treliça plana aplicada em pontes	20
Figura 2 - Sistema massa-mola.....	22
Figura 3 - Função de interpolação	31
Figura 4 - Transformação de coordenadas	33
Figura 5 - Montagem da matriz de rigidez de uma estrutura com 3 GDL	40
Figura 6 - Nós de uma treliça e os respectivos GDLs	41
Figura 7 - Fluxograma para elaboração do código	45
Figura 8 - Interface gráfica do código computacional	46
Figura 9 - Interface do Ansys Mechanical APDL 2022 R2 versão Academic Research	48
Figura 10 - Treliça Howe	52
Figura 11 - Primeiro modo de vibrar da treliça Howe obtido por meio do código desenvolvido	54
Figura 12 - Primeiro modo de vibrar da treliça Howe no software Ansys	54
Figura 13 - Segundo modo de vibrar da treliça Howe obtido por meio do código desenvolvido	55
Figura 14 - Segundo modo de vibrar da treliça Howe no software Ansys	55
Figura 15 - Terceiro modo de vibrar da treliça Howe obtido por meio do código desenvolvido	56
Figura 16 - Terceiro modo de vibrar da treliça Howe no software Ansys	56
Figura 17 - Treliça Pratt.....	58
Figura 18 - Primeiro modo de vibrar da treliça Pratt obtido por meio do código desenvolvido	60
Figura 19 - Primeiro modo de vibrar da treliça Pratt no software Ansys	60
Figura 20 - Segundo modo de vibrar da treliça Pratt obtido por meio do código desenvolvido	61
Figura 21 - Segundo modo de vibrar da treliça Pratt no software Ansys.....	61
Figura 22 - Terceiro modo de vibrar da treliça Pratt obtido por meio do código desenvolvido	62
Figura 23 - Terceiro modo de vibrar da treliça Pratt no software Ansys.....	62
Figura 24 - Treliça Warren	64
Figura 25 - Primeiro modo de vibrar da treliça Warren obtido por meio do código desenvolvido.....	66
Figura 26 - Primeiro modo de vibrar da treliça Warren no software Ansys	66
Figura 27 - Segundo modo de vibrar da treliça Warren obtido por meio do código desenvolvido.....	67
Figura 28 - Segundo modo de vibrar da treliça Warren no software Ansys	67
Figura 29 - Terceiro modo de vibrar da treliça Warren obtido por meio do código desenvolvido.....	68
Figura 30 - Terceiro modo de vibrar da treliça Warren no software Ansys.....	68

LISTA DE TABELAS

Tabela 1 - Frequências naturais da treliça Howe.....	53
Tabela 2 - Frequências naturais da treliça Pratt.....	59
Tabela 3 - Frequências naturais da treliça Warren.....	65

LISTA DE SÍMBOLOS

$F(t)$	Carregamento variante ao longo do tempo
F_{elast}	Força elástica
k	Rigidez do elemento
u	Deslocamento na direção axial da mola
F_{am}	Força de amortecimento
c	Coefficiente de amortecimento do sistema
\dot{u}	Velocidade do sistema
F_{in}	Força de inércia
m	Massa do sistema
\ddot{u}	Aceleração do sistema
C	Amplitude máxima do deslocamento
ω	Frequência natural
t	Tempo
φ_0	Ângulo de fase
$\{\delta d\}^{*T}$	Vetor transposto contendo os deslocamentos nodais virtuais
$\{\delta \psi\}^{*T}$	Vetor transposto contendo as funções deslocamento virtual nas direções x, y e z
$\{\delta \psi_A\}^{*T}$	Vetor transposto contendo as funções deslocamento virtual dos pontos pertencentes à superfície do elemento nas direções em que as forças de superfície atuam
$\{P\}$	Vetor contendo as forças nodais
$\{T_A\}$	Vetor contendo as forças de superfície por unidade de área

A	Área da superfície do elemento onde as forças de superfície atuam
$\{X\}$	Vetor contendo as forças de corpo por unidade de volume
$\{\ddot{\psi}\}$	Vetor contendo as funções aceleração nas direções x, y e z
ρ	Massa específica
V	Volume do elemento
$\{\delta\varepsilon\}^{*T}$	Vetor transposto contendo as componentes de deformação virtual
$\{\sigma\}$	Vetor contendo as componentes de tensão
$\{\psi_A\}$	Vetor contendo as funções de deslocamento de superfície
$\{\psi\}$	Vetor contendo as funções de deslocamento volumétrico
$\{N_A\}$	Vetor contendo as funções de forma de elemento de superfície
$\{N\}$	Vetor contendo as funções de forma de elemento volumétrico
$\{d\}$	Vetor contendo os deslocamentos nodais
$\{\ddot{d}\}$	Vetor contendo as acelerações nodais
$\{\varepsilon\}$	Vetor contendo as deformações específicas
$[L]$	Matriz que contém os operadores diferenciais
$[B]$	Matriz que contém os operadores diferenciais aplicados na função de forma do elemento
$[E]$	Matriz constitutiva
E	Módulo de elasticidade longitudinal
$\{f\}$	Vetor que contém as parcelas dos carregamentos

$[k]$	Matriz de rigidez do elemento
$[m]$	Matriz de massa do elemento
$[\bar{m}_{concentrada}]$	Matriz de massa concentrada
$[\bar{k}]$	Matriz de rigidez do elemento (4x4)
$\{\bar{d}\}$	Vetor contendo os deslocamentos nodais (4x1)
$[\bar{m}]$	Matriz de massa do elemento (4x4)
$\{\bar{\ddot{d}}\}$	Vetor contendo as acelerações nodais (4x1)
$\{\bar{f}\}$	Vetor que contém as parcelas dos carregamentos (4x1)
$[T]$	Matriz de transposição
$\{F_G\}$	Vetor que contém as parcelas dos carregamentos, em coordenadas globais
$\{d_G\}$	Vetor que contém os deslocamentos nodais, em coordenadas globais
$\{\ddot{d}_G\}$	Vetor que contém as acelerações nodais, em coordenadas globais
$[k_G]$	Matriz de rigidez do elemento, em coordenadas globais
$[m_G]$	Matriz de massa do elemento, em coordenadas globais
$[M]$	Matriz de massa do sistema
$[\ddot{\Delta}]$	Vetor de aceleração nodal
$[K]$	Matriz de rigidez do sistema
$[\Delta]$	Vetor de deslocamento nodal
$\{\phi_i\}$	Vetor dos modos de vibração

SUMÁRIO

1	INTRODUÇÃO	15
2	OBJETIVOS	17
2.1	Objetivo geral	17
2.2	Objetivos específicos	17
3	JUSTIFICATIVA	18
4	REFERENCIAL TEÓRICO	20
4.1	Treliças planas	20
4.2	Análise Modal via Método dos Elementos Finitos	21
4.2.1	Sistemas de um grau de liberdade	22
4.2.2	Sistemas de múltiplos graus de liberdade	25
4.2.3	Análise modal de treliças planas	30
<u>4.2.3.1</u>	<u>Matriz de transformação</u>	<u>32</u>
<u>4.2.3.2</u>	<u>Análise modal de treliças planas via MEF</u>	<u>37</u>
<u>4.2.3.3</u>	<u>Montagem das matrizes de massa e de rigidez globais</u>	<u>39</u>
4.3	MATLAB	41
4.4	Ansys	42
5	METODOLOGIA	44
5.1	Aspectos gerais	44
5.2	Aspectos computacionais	44
5.2.1	Algoritmo implementado	44
5.2.2	Desenvolvimento do código computacional	46
5.2.3	Utilizando o programa desenvolvido.....	46
5.2.4	Utilizando Ansys Mechanical APDL.....	48
6	ANÁLISE NUMÉRICA	51
6.1	Treliça Howe	51
6.1.1	Frequências naturais da treliça Howe	52
6.1.2	Modos de vibração da treliça Howe.....	53
6.2	Treliça Pratt	57
6.2.1	Frequências naturais da treliça Pratt	58
6.2.2	Modos de vibração da treliça Pratt	59
6.3	Treliça Warren	63
6.3.1	Frequências naturais da treliça Warren	64
6.3.2	Modos de vibração da treliça Warren	65

7	CONCLUSÃO	70
	REFERÊNCIAS.....	71
	APÊNDICE A - Código fonte do programa desenvolvido	72

1 INTRODUÇÃO

Os sistemas estruturais são formados por um conjunto de elementos com a finalidade de resistirem aos esforços solicitantes, provenientes de ações de origem estática ou dinâmica. Esses esforços podem ser tanto forças (normal e cortante) como também momentos (fletor e torção). Dentre os tipos de sistemas estruturais, vale destacar a relevância das treliças na concepção de pontes, coberturas e em torres de energia e telecomunicações, além de diversas outras aplicações. Um exemplo muito famoso de estrutura treliçada é a Torre Eiffel.

Ao contrário dos carregamentos estáticos que são considerados invariáveis ao longo do tempo (ou que variam tão lentamente que podem ser considerados constantes), um carregamento dinâmico pode ter sua intensidade, direção e/ou sentido alterados ao longo do tempo. As solicitações dinâmicas são capazes de variar a velocidade de cada um dos pontos de uma estrutura contínua, causando, dessa forma, uma aceleração. À essa aceleração podem ser associadas forças inerciais, que são abordadas na Dinâmica das Estruturas.

O estudo da teoria dinâmica aplicada à engenharia surgiu no século XX com Stephen Prokofyevich Timoshenko e Jacob Pieter Den Hartog. Em comparação à análise estática, uma análise dinâmica exige mais recursos computacionais e mais conhecimento por parte do projetista para que este seja capaz de interpretar os resultados obtidos de forma satisfatória (SORIANO, 2014, p. 2).

Ainda de acordo com Soriano (2014, p. 1), com o advento de concretos e aços cada vez mais resistentes, torna-se possível a utilização de elementos estruturais cada vez mais esbeltos e com maiores vãos, que são mais sujeitos a vibrações e com capacidade de dissipação de energia menor.

Na Dinâmica das Estruturas, a Análise Modal teórica permite prever os modos (ou formas modais) e as frequências naturais de vibração do sistema sujeito a uma perturbação inicial (vibração livre). Apesar de o tema ter muitas características relacionadas à Engenharia Mecânica, ao longo das últimas décadas a temática passou a ser abordada em outras áreas afins como a Engenharia Civil.

Para evitar o fenômeno físico da ressonância em estruturas, é importante conhecer a frequência da fonte excitadora e com isso projetar estruturas de tal forma que as primeiras frequências naturais estejam seguramente distantes das frequências dos carregamentos; os primeiros modos de vibração são os mais perigosos, assim

como os com harmônicos de mais baixa ordem de excitação. Considera-se que os harmônicos de mais baixa ordem correspondam àqueles de maior amplitude (ALVES FILHO, 2008, p. 182). O excesso de vibrações nas estruturas é capaz de comprometer o conforto dos usuários e, em último caso, a segurança das estruturas, podendo até mesmo levar ao colapso.

Atualmente, é altamente inviável que a análise do comportamento dinâmico de uma estrutura seja feita sem o auxílio de softwares especializados. Sem o auxílio da Mecânica Computacional, a elaboração de projetos estruturais seria uma tarefa extremamente laboriosa, o que tornaria o profissional que não faz uso de tais ferramentas pouco competitivo no mercado atual.

É neste contexto que se destaca o uso do Método dos Elementos Finitos (MEF), em inglês Finite Element Method (FEM), para a análise estrutural. O método é uma técnica poderosa utilizada para obter soluções aproximadas de equações diferenciais através de técnicas numéricas e auxílio computacional.

Segundo Logan (2008, p. 3), o Método dos Elementos Finitos surgiu durante a década de 1950 com Jon Turner. O método se apoia na análise matricial de estruturas, e foi difundido juntamente com a evolução dos computadores. Através do MEF, é possível tratar um meio contínuo partir de um meio discreto de elementos finitos com número finitos de graus de liberdade. Dessa forma, o comportamento estrutural do sistema contínuo que é descrito por equações diferenciais pode ser analisado a partir da resolução de um conjunto de equações algébricas. O processo em que um modelo matemático contínuo é substituído por um modelo com elementos finitos é chamado de processo de Discretização do Modelo Matemático Contínuo.

O MEF é fortemente associado à aplicações estruturais, mas também pode ser aplicado em diversos outros campos, como em transferência de calor, escoamento de fluídos, estudo de ondas eletromagnéticas, hidrodinâmica e até mesmo aplicações em bioengenharia.

Diante do que foi exposto, o presente trabalho tem como intuito apresentar uma rotina computacional em linguagem MATLAB capaz de efetuar a análise modal de treliças planas, por meio do MEF. As informações obtidas através da análise poderão ser utilizadas no estudo vibrações livres e também nas vibrações forçadas, sendo o segundo caso aquele em que há carregamentos variáveis em função do tempo sendo aplicados na estrutura.

2 OBJETIVOS

2.1 Objetivo geral

Elaborar um código computacional para análise das frequências naturais e modos de vibração de treliças planas, apoiado no Método dos Elementos Finitos.

2.2 Objetivos específicos

Os objetivos específicos do presente trabalho são:

- Compreender as deduções matemáticas para obtenção das matrizes de massa e de rigidez de treliças planas, essenciais na obtenção das frequências naturais e dos modos de vibração;
- Deduzir as equações matriciais que tratam os modos de vibração como função das frequências naturais através de um problema de autovalor e autovetor;
- Desenvolver e implementar um algoritmo no software MATLAB capaz de apresentar as frequências naturais, como também os modos de vibração de treliças planas, por meio de uma interface gráfica intuitiva;
- Comparar os resultados obtidos através do algoritmo apresentado com resultados obtidos a partir do software Ansys Mechanical APDL 2022 R2 versão Academic Research.

3 JUSTIFICATIVA

Nos sistemas estruturais, a análise dinâmica realizada de maneira adequada e precisa pode representar um fator crucial no sucesso de determinados empreendimentos. A falta dessa análise pode causar sérios danos econômicos e, considerando o pior caso, também à integridade da vida humana. É através desta análise que se obtém as características dinâmicas das estruturas, como frequências naturais e modos de vibração, os possíveis deslocamentos, velocidades, acelerações e esforços internos de toda estrutura.

No que se refere aos deslocamentos dos elementos estruturais, é possível afirmar que tais movimentos podem provocar um enorme desconforto aos usuários, induzindo-os a suporem, algumas vezes erroneamente, que a estrutura está prestes a ruir. Entretanto, há casos que, de fato, a vibração excessiva representa um grave perigo, como naqueles em que a ressonância tende a ampliar os deslocamentos até que haja o colapso de toda estrutura.

A Análise Modal de estruturas é particularmente importante para se evitar situações como a ocorrida no caso da ponte London Millennium Bridge. No dia de sua inauguração (em junho de 2000, na cidade de Londres), enquanto os pedestres andavam sobre a estrutura, os modos de vibração laterais eram excitados, provocando vibrações excessivas. Em determinado momento já não era mais possível caminhar, e só restava a todos se apoiarem no corrimão para evitar que caíssem. Devido ao visível risco dos usuários se machucarem, no dia 12 de junho a ponte teve que ser interditada para que o problema fosse resolvido. A solução adotada foi aumentar o amortecimento da estrutura artificialmente. A London Millennium Bridge só foi reaberta ao público no dia 22 de fevereiro de 2002 (NEWLAND, 2003).

Para todos os efeitos, é importante frisar a importância da análise dinâmica em tais circunstâncias, de tal forma que as vibrações sejam mantidas em valores mínimos, fazendo com que todos se sintam seguros e que nenhuma intervenção seja necessária.

Visto a importância das treliças e sua vasta aplicabilidade, em diversas situações, a análise dinâmica de tais elementos se torna imprescindível. A seguir, algumas situações em que devem ser consideradas a análise dinâmica na concepção estrutural de estruturas treliçadas:

- Estruturas em que há tráfego de veículos e pessoas, como pontes e passarelas;
- Estruturas sujeitas a sismos;
- Estruturas em que o efeito do vento não seja desprezível;
- Estruturas sujeitas a impactos e explosões;
- Estruturas que suportam máquinas com elementos em rotação;
- Estruturas sujeitas aos efeitos de ondas do mar;
- Estruturas offshore.

Diante do aparato tecnológico presente nos dias atuais, fica evidente a importância em se elaborar uma ferramenta computacional capaz de automatizar os cálculos envolvidos na análise estrutural proposta. Além de possibilitar a obtenção de resultados de forma ágil, economizando tempo e esforço por parte do projetista, a ferramenta também possibilita a obtenção de resultados mais seguros e confiáveis, uma vez que minimiza a possibilidade de erros nos processos matemáticos envolvidos. Dessa forma, a elaboração de um algoritmo computacional é justificada.

Por fim, é esperado que o software que elaborado auxilie estudantes e profissionais da área que pretendam realizar a análise modal de treliças planas de forma gratuita e com alta confiabilidade. Espera-se também que o desenvolvimento que será apresentado possa servir de base para aqueles que estão almejando aprofundar-se mais na análise modal de treliças planas por meio do MEF.

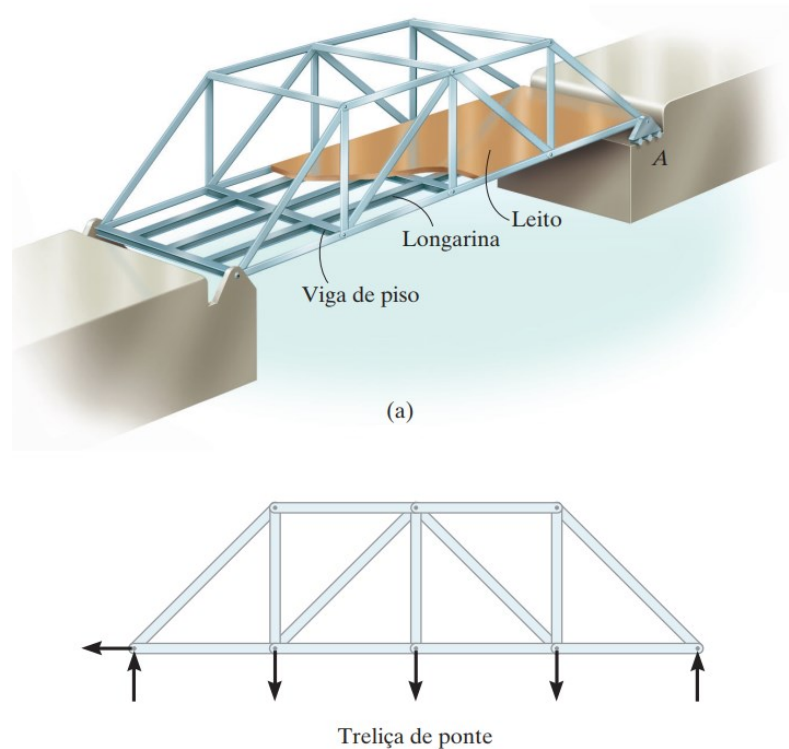
4 REFERENCIAL TEÓRICO

4.1 Treliças planas

A treliça é uma solução estrutural composta por membros esbeltos conectados entre si em suas extremidades. As treliças planas, que serão abordadas nesse trabalho, consistem em estruturas bidimensionais nas quais as barras constituintes formam triângulos em um mesmo plano, cujos carregamentos aplicados também atuam no mesmo plano da estrutura. Por simplificação, considera-se que as cargas são aplicadas nos nós da treliça e que os membros são conectados por pinos sem atrito. Dessa forma, em cada um dos nós haverá apenas forças nos eixos globais horizontal e vertical, e essas forças serão transmitidas às barras longitudinalmente de forma que só haja forças normais de tração ou compressão em cada elemento (HIBBELER, 2017, p. 239-241).

A Figura 1 ilustra o uso de duas treliças planas em paralelo como elementos estruturais em uma ponte.

Figura 1 - Treliça plana aplicada em pontes



Fonte: Hibbeler (2017, p. 240)

Tendo em vista que as forças que atuam nas barras sejam apenas axiais e que os elementos constituintes das treliças costumam ser esbeltos, particular atenção

deve ser dada às barras, sendo necessário que verifique se a flambagem em cada membro ultrapassa os limites normativos.

Na prática, as barras das treliças são unidas por conexões aparafusadas, rebitadas ou soldadas, mas por simplificação, considera-se que nos nós hajam pinos. Essa consideração implica em momento fletor nulo nos nós, que será conveniente na hora de equilibrar as forças em cada encontro de barras; assim, apenas duas equações de equilíbrio serão avaliadas em cada nó.

4.2 Análise Modal via Método dos Elementos Finitos

O primeiro pesquisador a definir as matrizes de rigidez para elementos treliçados, vigas e elementos triangulares e retangulares bidimensionais em tensão plana foi Turner, em 1956. Com a evolução dos computadores, o trabalho de Turner levou ao desenvolvimento de equações de rigidez expressas em notação matricial. O termo “elemento finito” foi introduzido por Clough, em 1960. Em 1965 Archer considerou efeitos dinâmicos no desenvolvimento da matriz consistente de massa, que é aplicado na análise de barras e vigas (LOGAN, 2007, p. 2-3).

Vale ressaltar que até 1950, os métodos matriciais presentes no MEF não eram tão úteis na resolução de problemas complexos, pois o número de equações algébricas obtidas tornava impraticável obter a solução do problema. Mas com a evolução dos computadores, tornou-se possível a resolução de milhares de equações em poucos minutos.

Mediante a Análise Modal é possível obter informações referentes às frequências naturais e também aos modos de vibração da estrutura. Para mais de um grau de liberdade, as frequências naturais e os modos de vibração são obtidos por meio da solução de um problema de autovalores e autovetores.

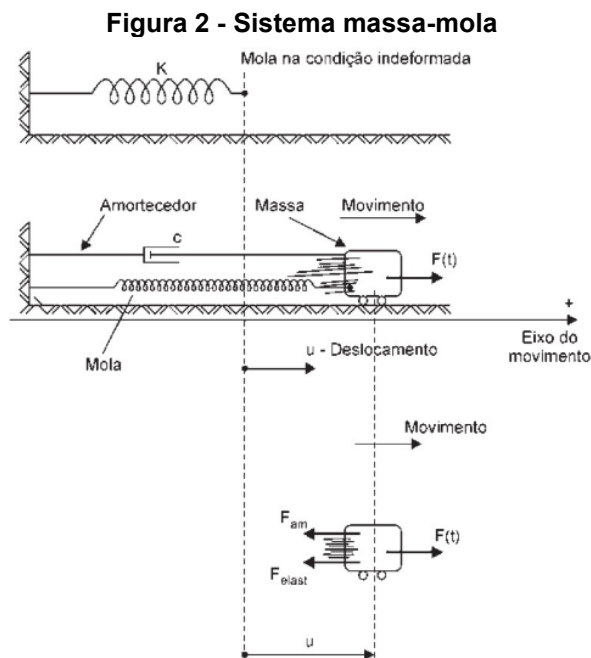
O número de graus de liberdade de um sistema mecânico refere-se ao número de componentes de deslocamentos independentes entre si exigido para localizar todas as massas constituintes do sistema. Uma estrutura real é constituída por um sistema de infinitos graus de liberdade.

Inicialmente, será demonstrado a obtenção da frequência natural em um sistema massa-mola com apenas um grau de liberdade, para que posteriormente, por analogia, a determinação da frequência natural possa ser expandida para n graus de liberdade.

4.2.1 Sistemas de um grau de liberdade

Primeiramente, será considerado o problema envolvendo um oscilador simples, que também é conhecido por sistema massa-mola. Neste caso, as deduções irão se basear em uma estrutura antes contínua e com infinitos graus de liberdade, agora com apenas um grau de liberdade, ou seja, com movimento permitido apenas em uma direção, no caso, na direção axial.

A Figura 2 ilustra um sistema massa-mola com amortecimento sob a ação de um carregamento externo $F(t)$ variante no tempo. $F(t)$ é aplicada ao sistema fazendo com que o elemento com um grau de liberdade se movimente.



Considerando o modelo linear, em decorrência da força aplicada e da deformação ocorrida na mola, surge então uma força restauradora elástica com sentido contrário à força aplicada conforme apresentado na equação (1).

$$F_{elast} = k \cdot u \quad (1)$$

em que k corresponde à rigidez da mola e u representa o deslocamento do elemento de massa na direção axial da mola.

Simultaneamente, surge também uma força de dissipação de energia proporcional à velocidade do objeto, para o caso de amortecimento viscoso, conhecida como força de amortecimento, com sentido também contrário à força aplicada:

$$F_{am} = c \cdot \dot{u} \quad (2)$$

sendo c o coeficiente de amortecimento do sistema e \dot{u} a velocidade do sistema.

Por fim, através da segunda Lei de Newton, a resultante das forças aplicadas em um elemento correspondem a uma força de inércia, que é a derivada da quantidade de movimento. Considerando a massa do sistema constante no tempo, tem-se que:

$$F_{in} = m \cdot \ddot{u} \quad (3)$$

onde m é a massa do sistema e \ddot{u} a aceleração do sistema.

Assim sendo, pelo equilíbrio dinâmico do sistema, é possível afirmar que:

$$F_{in} = F(t) - F_{elast} - F_{am} \quad (4)$$

Então,

$$F(t) = m \cdot \ddot{u} + c \cdot \dot{u} + k \cdot u \quad (5)$$

No que se refere às vibrações nas estruturas, em geral, é conveniente desconsiderar o efeito de amortecimento. Assim, deve-se tratar o oscilador simples como não amortecido. O efeito do amortecimento é mais relevante nas análises na região da ressonância.

Embora em aplicações reais sempre ocorra a dissipação da energia mecânica do sistema, que tende a diminuir as amplitudes das oscilações ao longo do tempo, aqui a dissipação será tratada como de pequena relevância. Segundo Alves Filho (2008, p. 235), na análise de estruturas sob força de impacto, o importante é a forma que a estrutura se comporta em um espaço curto de tempo, que praticamente não

depende do amortecimento. A matriz de amortecimento global e seus efeitos são relevantes em excitações de média e longa duração, no entanto, podem ser desprezíveis no caso de forças impulsivas (SORIANO, 2009, p. 49).

Quando um corpo preso a uma mola e sem presença de atrito é movido de sua posição de equilíbrio até um ponto qualquer e abandonado logo em seguida, o corpo realiza um movimento oscilatório conhecido como Movimento Harmônico Simples. Chama-se Harmônico devido as funções matemáticas utilizadas para descrever o movimento.

Como o objetivo do trabalho é obter as frequências naturais e modos de vibração, o carregamento aplicado no sistema poderá ser adotado como nulo. Assim, a equação (5) pode ser reduzida como segue:

$$m \cdot \ddot{u} + k \cdot u = 0 \quad (6)$$

Por não ter forças aplicadas e também considerando o amortecimento nulo, o movimento é denominado de vibração livre não amortecida. A vibração forçada leva em consideração o efeito de forças externas variáveis em função do tempo, tais como ação do vento, ondas, terremotos, movimentos de pessoas, etc.. Os softwares comerciais em elementos finitos desconsideram o amortecimento na Análise Modal, justamente pelo fato de pouco influenciar no resultado final (ALVES FILHO, 2008, p. 49).

Utilizando como referência a equação (6), é necessário definir a solução para a equação diferencial apresentada. Segundo Hutton (2004, p. 389), umas das possíveis soluções é do tipo:

$$u(t) = C \cdot \text{sen}(\omega \cdot t + \varphi_0) \quad (7)$$

em que,

C : amplitude do deslocamento;

ω : frequência natural;

t : tempo;

φ_0 : ângulo de fase.

Derivando-se a equação (7) duas vezes em relação ao tempo, chega-se a:

$$\ddot{u}(t) = -\omega^2 \cdot C \cdot \text{sen}(\omega \cdot t + \varphi_0) \quad (8)$$

Substituindo então as equações (7) e (8) em (6), tem-se:

$$m \cdot [-\omega^2 \cdot C \cdot \text{sen}(\omega \cdot t + \varphi_0)] + k \cdot C \cdot \text{sen}(\omega \cdot t + \varphi_0) = 0 \quad (9)$$

$$(k \cdot C - m \cdot \omega^2 \cdot C) \cdot \text{sen}(\omega \cdot t + \varphi_0) = 0 \quad (10)$$

A igualdade matemática poderá ser satisfeita em situações pontuais em que $\text{sen}(n \cdot \pi) = 0$, com n sendo um número inteiro. De maneira mais abrangente, e pensando em obter a frequência natural para o sistema massa-mola, a equação também será satisfeita se:

$$k \cdot C - m \cdot \omega^2 \cdot C = 0 \quad (11)$$

Através da equação (11) e as devidas manipulações algébricas, obtém-se a frequência natural para um sistema com um grau de liberdade:

$$\omega = \sqrt{\frac{k}{m}} \quad (12)$$

em que k é descrito em N/m , m em kg e ω em rad/s , no Sistema Internacional de Unidades (SI).

4.2.2 Sistemas de múltiplos graus de liberdade

Um dos motivos da enorme aceitação do MEF se deve a discretização do contínuo, que produz equações algébricas simultâneas que podem ser facilmente resolvidas com o auxílio de computadores. Os dados dos sistemas de equações

deverão ser armazenados por meio de matrizes, que são amplamente utilizadas no MEF.

A ideia da discretização de um sistema contínuo leva em conta que a estrutura será dividida em partes distintas, conectadas entre si nos pontos discretos, os nós. A solução aproximada obtida simula a estrutura como um conjunto de elementos com dimensões finitas, e não diferencial, como no caso contínuo. Para o caso das treliças, que têm suas barras reticuladas, as barras e conexões podem ser tratadas como casos já discretizados.

De acordo com Alves Filho (2008, p. 33), um sistema discreto com n graus de liberdade apresenta n frequências naturais, em que cada uma poderá ser associada a um modo de vibração. Antes de verificar como uma estrutura responde dinamicamente a uma série de carregamentos externos, é necessário determinar seus modos e as frequências naturais de vibração. Esse estudo é chamado de Análise Modal.

Na Análise Dinâmica por meio do MEF, o modelo discretizado da estrutura contínua estudada é caracterizada por uma grande quantidade de molas, massas e amortecedores, que descrevem trecho a trecho a rigidez, massa associada e amortecimento de cada elemento finito.

Visando definir a matriz de massa e de rigidez de treliças planas, inicialmente, considera-se que a energia inserida no sistema devido ao trabalho das forças externas seja totalmente transformada em energia interna de deformação. Assim, aplicando-se o Princípio dos Trabalhos Virtuais (PTV) em um elemento finito, tem-se:

$$\delta U^{(e)} = \delta W^{(e)} \quad (13)$$

As forças que contribuirão para a parcela do trabalho externo são as forças concentradas, forças de superfície e forças de volume. Além disso, de acordo com o princípio de D'Alembert, uma força inercial dada pelo produto da massa do sistema pela sua aceleração atua em sentido oposto ao movimento (LOGAN, 2007, p.756). Dessa forma, tem-se:

$$\delta W^{(e)} = \{\delta d\}^{*T} \cdot \{P\} + \int_A \{\delta \psi_A\}^{*T} \cdot \{T_A\} \cdot dA + \int_V \{\delta \psi\}^{*T} (\{X\} - \rho \cdot \{\ddot{y}\}) \cdot dV \quad (14)$$

onde:

$\{\delta d\}^{*T}$: vetor transposto contendo os deslocamentos nodais virtuais;

$\{\delta \psi\}^{*T}$: vetor transposto contendo as funções deslocamento virtual nas direções x, y e z;

$\{\delta \psi_A\}^{*T}$: vetor transposto contendo as funções deslocamento virtual dos pontos pertencentes à superfície do elemento nas direções em que as forças de superfície atuam;

$\{P\}$: vetor contendo as forças nodais;

$\{T_A\}$: vetor contendo as forças de superfície por unidade de área;

A : área da superfície do elemento onde as forças de superfície atuam;

$\{X\}$: vetor contendo as forças de corpo por unidade de volume;

ρ : massa específica;

$\{\ddot{\psi}\}$: vetor contendo as funções aceleração nas direções x, y e z;

V : volume do elemento.

Para a parcela do trabalho interno, considerando a energia específica de deformação, é possível chegar na seguinte expressão:

$$\delta U^{(e)} = \int_V \{\delta \varepsilon\}^{*T} \cdot \{\sigma\} \cdot dV \quad (15)$$

em que:

$\{\delta \varepsilon\}^{*T}$: vetor transposto contendo as componentes de deformação virtual;

$\{\sigma\}$: vetor contendo as componentes de tensão.

Substituindo (14) e (15) em (13):

$$\int_V \{\delta \varepsilon\}^{*T} \cdot \{\sigma\} \cdot dV = \{\delta d\}^{*T} \cdot \{P\} + \int_A \{\delta \psi_A\}^{*T} \cdot \{T_A\} \cdot dA + \int_V \{\delta \psi\}^{*T} (\{X\} - \rho \cdot \{\ddot{\psi}\}) \cdot dV \quad (16)$$

Os deslocamentos de um ponto qualquer de um elemento podem ser representados através da função de forma ou interpolação, N , multiplicadas pelos deslocamentos nodais d que é função apenas do tempo. Assim, tem-se:

$$\{\psi_A\} = [N_A] \cdot \{d\} \quad (17)$$

$$\{\psi\} = [N] \cdot \{d\} \quad (18)$$

As deformações são relacionadas aos deslocamentos por meio de relações diferenciais. Considerando $[L]$ como a matriz que contém os operadores diferenciais, é possível escrever:

$$\{\varepsilon\} = [L] \cdot [N] \cdot \{d\} \quad (19)$$

sendo

$$[B] = [L] \cdot [N] \quad (20)$$

então,

$$\{\varepsilon\} = [B] \cdot \{d\} \quad (21)$$

As relações constitutivas entre tensão e deformação podem ser expressas como:

$$\{\sigma\} = [E] \cdot \{\varepsilon\} \quad (22)$$

Substituindo as equações (17), (18) e (21) em (16), tem-se a equação (23):

$$\int_V \{\delta d\}^{*T} \cdot [B]^T \cdot [E] \cdot [B] \cdot \{d\} \cdot dV = \{\delta d\}^{*T} \cdot \{P\} + \int_A \{\delta d\}^{*T} \cdot [N_A]^T \cdot \{T_A\} \cdot dA + \int_V \{\delta d\}^{*T} \cdot [N]^T \cdot (\{X\} - \rho \cdot [N] \cdot \{\ddot{d}\}) \cdot dV \quad (23)$$

Como os deslocamentos nodais virtuais são constantes e não nulos, a equação (23) pode ser reduzida à:

$$\int_V [B]^T \cdot [E] \cdot [B] \cdot dV \cdot \{d\} = \{P\} + \int_A [N_A]^T \cdot \{T_A\} \cdot dA + \int_V [N]^T \cdot (\{X\} - \rho \cdot [N] \cdot \{\ddot{d}\}) \cdot dV \quad (24)$$

ou ainda:

$$\left(\int_V [B]^T \cdot [E] \cdot [B] \cdot dV \right) \cdot \{d\} = \{P\} + \int_A [N_s]^T \cdot \{T_A\} \cdot dA + \int_V [N]^T \cdot \{X\} \cdot dV - \left(\int_V \rho \cdot [N]^T \cdot [N] \cdot dV \right) \cdot \{\ddot{d}\} \quad (25)$$

e na forma mais compacta,

$$[m] \cdot \{\ddot{d}\} + [k] \cdot \{d\} = \{f\} \quad (26)$$

onde $[k]$ é a matriz de rigidez do elemento, $[m]$ a matriz de massa do elemento e $\{f\}$ o vetor que contém as parcelas dos carregamentos, sendo dados por:

$$[k] = \int_V [B]^T \cdot [E] \cdot [B] \cdot dV \quad (27)$$

$$[m] = \int_V \rho \cdot [N]^T \cdot [N] \cdot dV \quad (28)$$

$$\{f\} = \{P\} + \int_A [N_s]^T \cdot \{T_A\} \cdot dA + \int_V [N]^T \cdot \{X\} \cdot dV \quad (29)$$

4.2.3 Análise modal de treliças planas

Particularizando a aplicação da equação (25) para um elemento de treliça plana, primeiramente, serão desconsideradas as parcelas dos carregamentos. Assim, considerando que não há nenhuma força externa sendo aplicada na estrutura, tem-se:

$$\left(\int_V \rho \cdot [N]^T \cdot [N] \cdot dV \right) \cdot \{\ddot{d}\} + \left(\int_V [B]^T \cdot [E] \cdot [B] \cdot dV \right) \cdot \{d\} = \{0\} \quad (30)$$

Por se tratar de um elemento estrutural que apresenta um estado de tensão uniaxial, a matriz constitutiva $[E]$ pode ser substituída pelo módulo de elasticidade longitudinal E . Considerando a massa específica do material e a área de seção transversal do elemento constantes no comprimento e, assim, $dV = A \cdot dx$, tem-se:

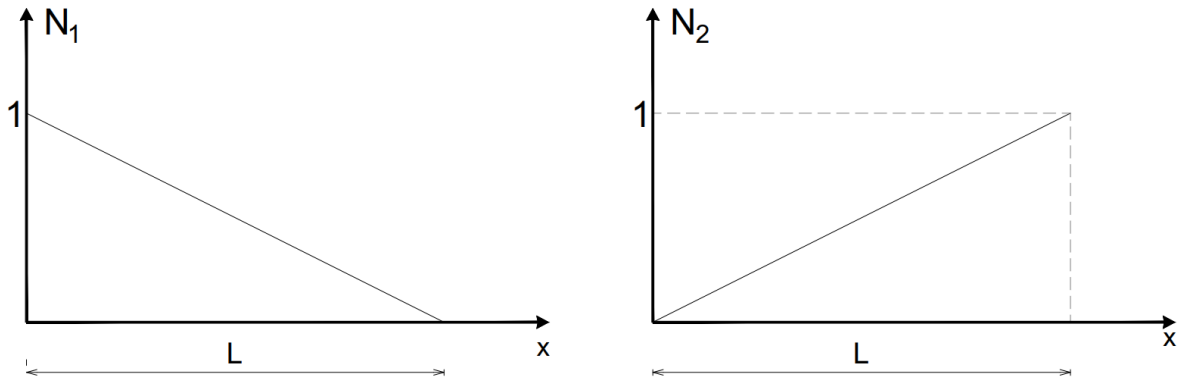
$$\rho \cdot A \cdot \left(\int_0^L [N]^T \cdot [N] \cdot dx \right) \cdot \{\ddot{d}\} + E \cdot A \cdot \left(\int_0^L [B]^T \cdot [B] \cdot dx \right) \cdot \{d\} = \{0\} \quad (31)$$

Para o caso de treliças planas, as funções de interpolação, N , podem ser escritas de acordo com a equação (32) e expressas graficamente conforme ilustra a Figura 3.

A função de interpolação N é idealizada de tal forma que em um dos nós há um valor unitário, variando até zero na outra extremidade.

$$[N] = \left[1 - \frac{x}{L} \quad \frac{x}{L} \right] \quad (32)$$

Figura 3 - Função de interpolação



Fonte: Adaptado de Logan (2007, p. 31)

Com a função de interpolação definida para este caso, é possível obter o valor de B , expresso na equação (33):

$$[B] = \begin{bmatrix} -1 & 1 \\ L & L \end{bmatrix} \quad (33)$$

Substituindo (32) e (33) em (31), tem-se:

$$\rho \cdot A \cdot \left(\int_0^L \begin{bmatrix} 1 - \frac{x}{L} \\ \frac{x}{L} \end{bmatrix} \cdot \begin{bmatrix} 1 - \frac{x}{L} & \frac{x}{L} \end{bmatrix} dx \right) \cdot \{\ddot{d}\} + E \cdot A \cdot \left(\int_0^L \begin{bmatrix} -1 \\ 1 \\ L \end{bmatrix} \cdot \begin{bmatrix} -1 & 1 \\ L & L \end{bmatrix} \cdot dx \right) \cdot \{d\} = \{0\} \quad (34)$$

Resolvendo a integral e simplificando:

$$\frac{\rho \cdot A \cdot L}{6} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \cdot \{\ddot{d}\} + \frac{E \cdot A}{L} \cdot \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \cdot \{d\} = \{0\} \quad (35)$$

ou ainda de forma mais compacta:

$$[m] \cdot \{\ddot{d}\} + [k] \cdot \{d\} = \{0\} \quad (36)$$

em que:

$[k]$: matriz de rigidez do elemento, em coordenadas locais;

$\{d\}$: vetor que contém os deslocamentos nodais, em coordenadas locais;

$[m]$: matriz de massa consistente do elemento, em coordenadas locais;

$\{\ddot{d}\}$: vetor que contém as acelerações nodais, em coordenadas locais.

sendo

$$[k] = \frac{E \cdot A}{L} \cdot \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \quad (37)$$

$$[m] = \frac{\rho \cdot A \cdot L}{6} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \quad (38)$$

A matriz de massa deduzida (e que será utilizada posteriormente) é a matriz de massa consistente. No entanto, há também a matriz de massa concentrada. Para matriz de massa concentrada, conforme apresentado na equação (39), considera-se que metade de toda a massa do elemento está concentrada em cada um dos dois nós. Matrizes de massa consistente, ao contrário das matrizes de massa concentrada, são comprovadamente eficazes em fornecer limites superiores para frequências circulares naturais. Em contrapartida, matrizes de massa concentrada são frequentemente utilizadas em elementos de barra e vigas para obter resultados razoavelmente precisos de resposta dinâmica (LOGAN, 2007, p.756).

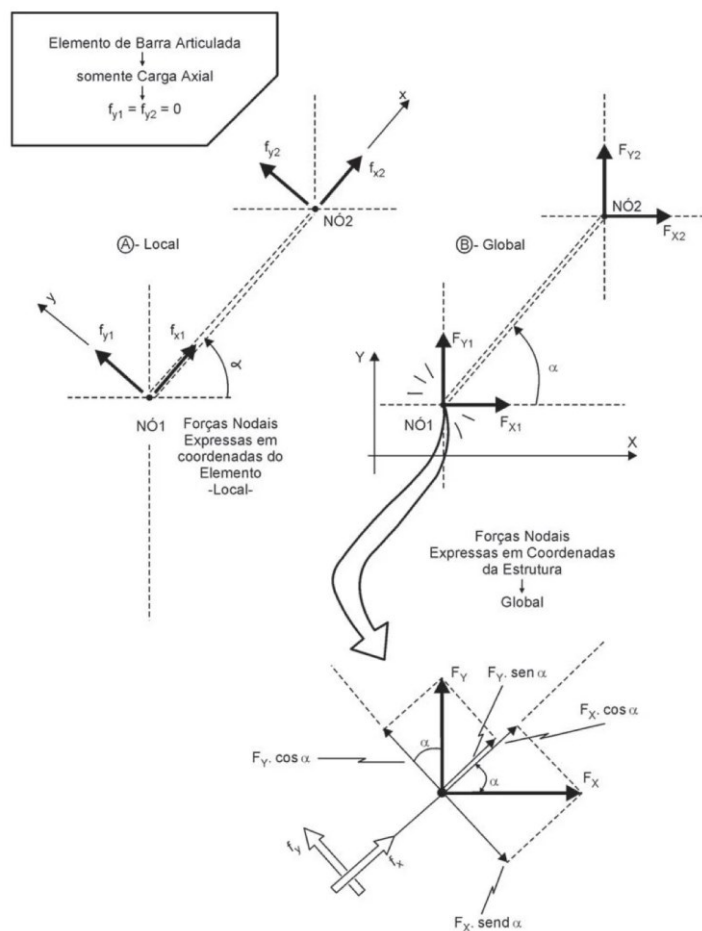
$$[m_{concentrada}] = \frac{\rho \cdot A \cdot L}{2} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (39)$$

4.2.3.1 Matriz de transformação

Na equação (35) foram utilizados como referência os eixos locais de cada barra, que podem variar para cada elemento constituinte da estrutura. Para que seja possível obter uma matriz de rigidez e de massa consistente que represente a estrutura toda, é necessário antes referenciar as equações em um sistema de coordenadas global.

Para treliças, o sistema de coordenadas local, representado por letras minúsculas (x, y), refere-se àquele em que a direção principal x é traçada por uma reta que passa pelos dois nós de uma barra; a direção secundária é perpendicular a primeira. No sistema global, representado por letras maiúsculas (X, Y), é comum chamar de eixo X aquele ao longo da direção horizontal, e de Y o que é perpendicular ao primeiro. Os sistemas local e global de coordenadas pode ser visualizado na Figura 4.

Figura 4 - Transformação de coordenadas



Fonte: Alves Filho (2013, p. 78)

Partindo-se da equação (26), particularizada para um elemento, tem-se:

$$[m] \cdot \{\ddot{d}\} + [k] \cdot \{d\} = \{f\} \quad (40)$$

que na forma explícita pode ser expressa como segue:

$$\frac{\rho \cdot A \cdot L}{6} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \cdot \begin{Bmatrix} \ddot{u}_{x1} \\ \ddot{u}_{x2} \end{Bmatrix} + \frac{E \cdot A}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \cdot \begin{Bmatrix} u_{x1} \\ u_{x2} \end{Bmatrix} = \begin{Bmatrix} f_{x1} \\ f_{x2} \end{Bmatrix} \quad (41)$$

Para o sistema de coordenadas global, as matrizes de rigidez e de massa global de cada elemento terão dimensões 4x4, uma vez que o elemento apresenta dois graus de liberdade em cada um dos nós (deslocamentos nas direções X e Y). Assim, para que se possa realizar as transformações necessárias, as matrizes de rigidez e de massa local, de dimensões 2x2, devem ser representadas por matrizes 4x4. De acordo com Logan (2007, p.79) e Hutton (2004, p. 438):

$$\frac{\rho \cdot A \cdot L}{6} \begin{bmatrix} 2 & 0 & 1 & 0 \\ 0 & 2 & 0 & 1 \\ 1 & 0 & 2 & 0 \\ 0 & 1 & 0 & 2 \end{bmatrix} \cdot \begin{Bmatrix} \ddot{u}_{x1} \\ \ddot{u}_{y1} \\ \ddot{u}_{x2} \\ \ddot{u}_{y2} \end{Bmatrix} + \frac{E \cdot A}{L} \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{Bmatrix} u_{x1} \\ u_{y1} \\ u_{x2} \\ u_{y2} \end{Bmatrix} = \begin{Bmatrix} f_{x1} \\ f_{y1} \\ f_{x2} \\ f_{y2} \end{Bmatrix} \quad (42)$$

Segundo Hutton (2004, p.434), os deslocamentos transversais são ignorados no desenvolvimento da matriz de rigidez em coordenadas globais, pois não há rigidez transversal em virtude da suposição de conexões por pinos, as quais permitem a rotação livremente. No entanto, movimento na direção transversal à barra introduz energia cinética no sistema, que não pode ser desprezado na dedução da matriz de massa em coordenadas globais.

Simplificando a equação (42):

$$[\bar{m}] \cdot \{\bar{\ddot{d}}\} + [\bar{k}] \cdot \{\bar{d}\} = \{\bar{f}\} \quad (43)$$

Projetando as forças globais em ambos os eixos locais:

$$\begin{cases} f_{x1} = F_{X1} \cdot \cos \alpha + F_{Y1} \cdot \sin \alpha \\ f_{y1} = -F_{X1} \cdot \sin \alpha + F_{Y1} \cdot \cos \alpha \\ f_{x2} = F_{X2} \cdot \cos \alpha + F_{Y2} \cdot \sin \alpha \\ f_{y2} = -F_{X2} \cdot \sin \alpha + F_{Y2} \cdot \cos \alpha \end{cases} \quad (44)$$

Considerando de maneira simplificada $c = \cos \alpha$ e $s = \sin \alpha$, é possível reescrever a equação (44) em forma matricial:

$$\begin{Bmatrix} f_{x1} \\ f_{y1} \\ f_{x2} \\ f_{y2} \end{Bmatrix} = \begin{bmatrix} c & s & 0 & 0 \\ -s & c & 0 & 0 \\ 0 & 0 & c & s \\ 0 & 0 & -s & c \end{bmatrix} \cdot \begin{Bmatrix} F_{X1} \\ F_{Y1} \\ F_{X2} \\ F_{Y2} \end{Bmatrix} \quad (45)$$

ou ainda,

$$\{\bar{f}\} = [T] \cdot \{F_G\} \quad (46)$$

em que a matriz $[T]$, de transformação, é uma matriz ortogonal. Assim sendo:

$$[T]^T = [T]^{-1} \quad (47)$$

De maneira análoga ao realizado para as forças, é possível fazer para os deslocamentos e acelerações nodais, ou seja:

$$\{\bar{d}\} = [T] \cdot \{d_G\} \quad (48)$$

$$\{\bar{\ddot{d}}\} = [T] \cdot \{\ddot{d}_G\} \quad (49)$$

Com o intuito de obter as matrizes de rigidez e de massa do elemento no sistema global, é necessário que (46) e (48) e (49) sejam substituídos na equação (43). Assim:

$$[\bar{m}] \cdot [T] \{\ddot{d}_G\} + [\bar{k}] \cdot [T] \cdot \{d_G\} = [T] \{F_G\} \quad (50)$$

$$[T]^{-1} \cdot [\bar{m}] \cdot [T] \{\ddot{d}_G\} + [T]^{-1} \cdot [\bar{k}] \cdot [T] \cdot \{d_G\} = [T]^{-1} \cdot [T] \{F_G\} \quad (51)$$

$$[T]^{-1} \cdot [\bar{m}] \cdot [T] \{\ddot{d}_G\} + [T]^{-1} \cdot [\bar{k}] \cdot [T] \cdot \{d_G\} = [I] \{F_G\} \quad (52)$$

$$[T]^{-1} \cdot [\bar{m}] \cdot [T] \{\ddot{d}_G\} + [T]^{-1} \cdot [\bar{k}] \cdot [T] \cdot \{d_G\} = \{F_G\} \quad (53)$$

ou ainda de acordo com a equação (54),

$$[m_G] \cdot \{\ddot{d}_G\} + [k_G] \cdot \{d_G\} = \{F_G\} \quad (54)$$

onde

$\{F_G\}$: vetor que contém as forças externas, em coordenadas globais;

$[k_G]$: matriz de rigidez do elemento, em coordenadas globais;

$\{d_G\}$: vetor que contém os deslocamentos nodais, em coordenadas globais.

$[m_G]$: matriz de massa do elemento, em coordenadas globais;

$\{\ddot{d}_G\}$: vetor que contém as acelerações nodais, em coordenadas globais.

sendo

$$[k_G] = \frac{E \cdot A}{L} \cdot \begin{bmatrix} c^2 & cs & -c^2 & -cs \\ cs & s^2 & -cs & -s^2 \\ -c^2 & -cs & c^2 & cs \\ -cs & -s^2 & cs & s^2 \end{bmatrix} \quad (55)$$

$$[m_G] = \frac{\rho \cdot A \cdot L}{6} \cdot \begin{bmatrix} 2 & 0 & 1 & 0 \\ 0 & 2 & 0 & 1 \\ 1 & 0 & 2 & 0 \\ 0 & 1 & 0 & 2 \end{bmatrix} \quad (56)$$

É possível perceber que as matrizes de rigidez e de massa de um elemento em coordenadas globais são simétricas. Para as matrizes globais da estrutura ocorrerá o mesmo.

4.2.3.2 Análise modal de treliças planas via MEF

De maneira similar ao que foi feito para um grau de liberdade no item 4.2.1 deste trabalho, para sistemas com múltiplos graus de liberdade é possível afirmar que:

$$[M] \cdot \{\ddot{\Delta}\} + [K] \cdot \{\Delta\} = \{0\} \quad (57)$$

onde:

$[M]$: matriz de massa do sistema;

$[K]$: matriz de rigidez do sistema;

$\{\ddot{\Delta}\}$: vetor de aceleração nodal;

$\{\Delta\}$: vetor de deslocamento nodal;

Tomando como base o que foi feito anteriormente, sabe-se que:

$$\{\Delta(t)\} = \{\Delta_0\} \cdot \text{sen}(\omega \cdot t + \varphi_0) \quad (58)$$

e conseqüentemente,

$$\{\ddot{\Delta}(t)\} = -\{\Delta_0\} \cdot \omega^2 \cdot \text{sen}(\omega \cdot t + \varphi_0) \quad (59)$$

Substituindo as equações (58) e (59) em (57):

$$-[M] \cdot \{\Delta_0\} \cdot \omega^2 \cdot \text{sen}(\omega \cdot t + \varphi_0) + [K] \cdot \{\Delta_0\} \cdot \text{sen}(\omega \cdot t + \varphi_0) = \{0\} \quad (60)$$

$$\left(-[M] \cdot \{\Delta_0\} \cdot \omega^2 + [K] \cdot \{\Delta_0\}\right) \cdot \text{sen}(\omega \cdot t + \varphi_0) = \{0\} \quad (61)$$

Para que o produto apresentado em (61) seja nulo, uma vez que $\text{sen}(\omega \cdot t + \varphi_0)$ pode assumir qualquer valor no intervalo $\{-1, 1\}$, é necessário que seja mantida a seguinte igualdade:

$$-[M] \cdot \{\Delta_0\} \cdot \omega^2 + [K] \cdot \{\Delta_0\} = \{0\} \quad (62)$$

ou

$$([K] - \omega^2 \cdot [M]) \cdot \{\Delta_0\} = \{0\} \quad (63)$$

A equação (63) sugere soluções não triviais e uma solução trivial. Para a solução trivial, é verificado que $\{\Delta_0\} = \{0\}$ é um resultado possível para o problema. Nesse caso, não há deslocamentos iniciais no sistema, e por consequência não há vibrações livres.

As soluções não triviais ocorrem para:

$$([K] - \omega^2 \cdot [M]) = \{0\} \quad (64)$$

Para isso é necessário que o determinante seja nulo, conforme apresentado na equação (65). As equações terão como incógnita ω^2 , que correspondem aos valores das frequências naturais ao quadrado dos n modos de vibração. Assim:

$$\det([K] - \omega^2 \cdot [M]) = 0 \quad (65)$$

que representa a equação de frequência do sistema. Considerando agora $\lambda = \omega^2$:

$$\det([K] - \lambda \cdot [M]) = 0 \quad (66)$$

Os valores de $\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_n$ que satisfazem a equação são os autovalores do sistema. Assim:

$$p(\lambda) = \det([K] - \lambda \cdot [M]) = 0 \quad (67)$$

É possível associar cada autovalor a um autovetor, em que este refere-se ao modo de vibração da estrutura:

$$([K] - \lambda_i \cdot [M]) \cdot \{\phi_i\} = \{0\} \quad (68)$$

onde $\{\phi_i\}$ representa os modos de vibração.

Segundo Alves Filho (2008, p. 168), é comum representar todos os autovetores em uma única matriz, que é chamada de Matriz dos Autovetores. Nesta matriz, as linhas referem-se a cada um dos GDLs da estrutura analisada, enquanto que as colunas estão relacionadas a cada um dos modos de vibração.

Os valores obtidos através dos autovetores fornecem uma indicação geométrica dos modos de vibrar. No entanto, não são indicativos da amplitude de cada vibração (HUTTON, 2004, p. 440). Para cada modo de vibrar da estrutura, o que de fato importa é que a proporção entre os valores obtidos referentes a cada GDL seja mantida; isso permite os modos de vibração sejam normalizados de inúmeras formas.

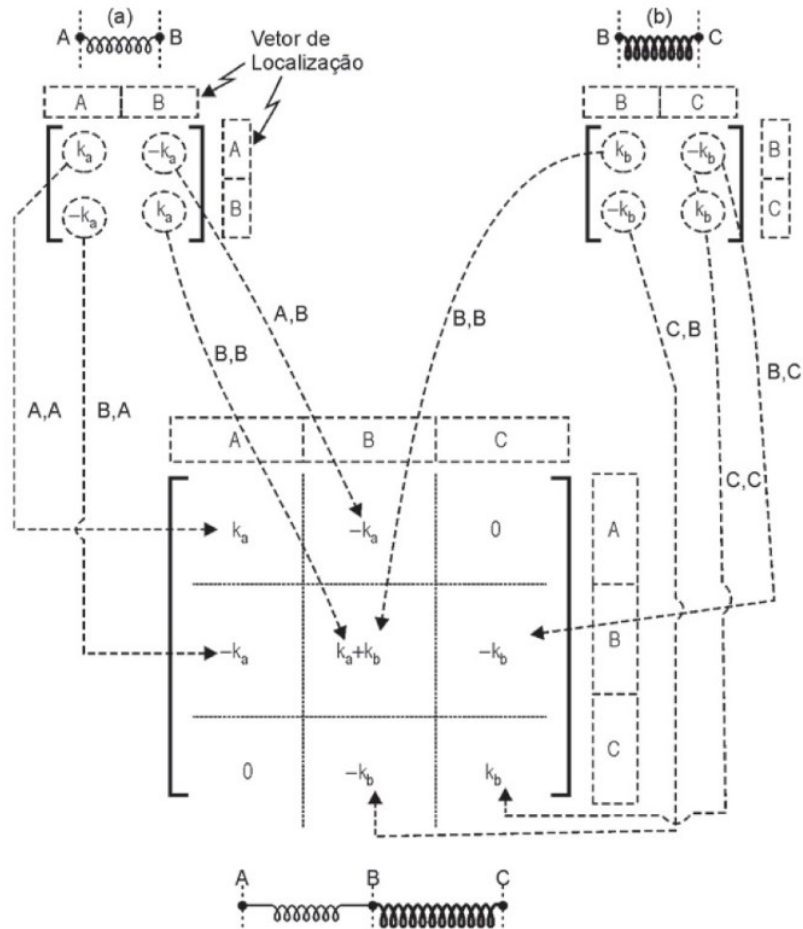
Em situações práticas, os modos superiores de vibração podem ser considerados desprezíveis devido a pouca ou nenhuma influência que esses têm na resposta dinâmica. O que de fato importa é determinar os primeiros pares “frequência e modo de vibração”.

4.2.3.3 Montagem das matrizes de massa e de rigidez globais

Para exemplificar a montagem das matrizes de massa e de rigidez globais de uma estrutura, será analisado apenas o caso da matriz de rigidez, visto que para a matriz de massa o raciocínio é o mesmo.

A Figura 5 demonstra uma matriz de rigidez de uma estrutura com 3 graus de liberdade (3 GDL), sendo 1 GDL para cada um dos nós A, B e C.

Figura 5 - Montagem da matriz de rigidez de uma estrutura com 3 GDL



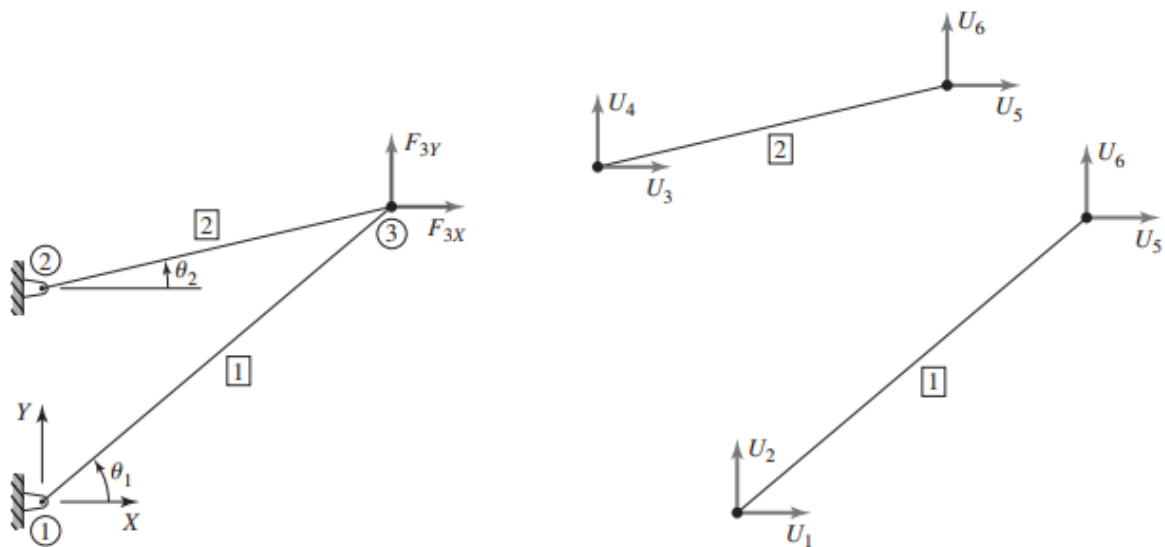
Fonte: Alves Filho (2013, p. 44)

Para o caso apresentado, percebe-se que cada linha e cada coluna da matriz de rigidez global estão associadas aos graus de liberdade de cada nó. Cada elemento da matriz elementar é posicionado na matriz estrutural de acordo com sua correspondente linha e coluna. É possível reparar que nos nós comuns a dois ou mais elementos, ocorre a soma da rigidez de cada elemento que está ligado a tal nó. Em situações em que não há interação entre os nós (como o que ocorre com os nós A e C), é assumido o valor nulo na matriz estrutural.

As condições de contorno do problema deverão ser consideradas no cálculo das frequências naturais e dos modos de vibração, afinal não há como a estrutura vibrar nas direções restringidas ao movimento. De acordo com Alves Filho (2013, p 58), a matriz de rigidez de toda a estrutura representa a estrutura sem nenhum vínculo, ou seja, hipostática. Para resolver esse impasse, é necessário que cada linha e cada coluna das matrizes de rigidez (e também de massa) da estrutura referentes a cada um dos GDLs restringidos sejam eliminadas.

Em aplicações computacionais, é comum utilizar números para descrever cada nó ao invés de letras, pois dessa forma torna-se mais prático a montagem das matrizes de massa e rigidez da estrutura analisada. Ao utilizar números para representar cada um dos nós, é possível enumerar cada um dos GDL mantendo uma lógica com o nó a que cada GDL se refere. Por meio da Figura 6, é possível verificar que cada um dos GDLs na direção vertical é um múltiplo de dois do nó a que esse grau de liberdade está associado. Para a direção horizontal, cada GDL é o dobro do valor assumido pelo nó menos uma unidade.

Figura 6 - Nós de uma treliça e os respectivos GDLs



Fonte: Hutton (2004, p. 54)

4.3 MATLAB

O MATLAB (abreviatura de MATrix LABoratory, cuja tradução literal é Laboratório de Matriz) é um software poderoso na área de computação numérica científica, amplamente utilizado no meio acadêmico, especialmente nas áreas de matemática, ciências e, especialmente, engenharias. O software também é amplamente utilizado na indústria como ferramenta de pesquisa. Sua linguagem tem como base matrizes, como o próprio nome sugere (GILAT, 2012, p. 1).

De acordo com Matsumoto (2013, p. 13), a linguagem de programação MATLAB é uma linguagem de programação de alto nível, mais simples e direta do que em linguagens como C, VB ou Fortran.

O MATLAB é muito versátil no que se refere a modelagens e simulações, cálculos matemáticos, gráficos, etc.. A razão de sua popularidade deve-se à variedade de funções predefinidas e à facilidade de utilização. O pacote padrão do MATLAB é constituído por inúmeras funções, que podem ser aplicadas em diversas áreas do conhecimento. Ademais, há ainda uma série de *toolboxes* projetadas para resolver problemas específicos.

Há diversas ferramentas no software capazes de resolverem problemas complexos em uma única etapa, o que em outras linguagens pode ser mais trabalhoso ou até mesmo implicar na compra de pacotes de softwares de terceiros para suprir a ausência de determinadas funções. As funções do MATLAB são bastante completas, pois muitas delas foram elaboradas e testadas por diversos pesquisadores, produzindo resultados consistentes para uma enorme gama de aplicações (CHAPMAN, 2016, p. 1).

4.4 Ansys

A princípio, os primeiros programas de elementos finitos eram códigos especiais desenvolvidos para resolver um tipo específico de problema (contemplavam somente um tipo de elemento e uma análise física). Era comum que diferentes grupos em uma mesma organização utilizassem diferentes programas de computador. Em muitos casos, cada grupo de engenheiros desenvolvia e utilizava seu próprio código, o que gerava preocupações sobre a compatibilidade dos resultados de diferentes programas (THOMPSON, 2017, p. 4).

Em 1965, a National Aeronautics and Space Administration (NASA) iniciou um projeto com o intuito de criar um programa de computador que pudesse ser usado por todas as suas organizações de engenharia para resolver uma variedade de problemas estruturais relacionados ao desenvolvimento da tecnologia de exploração lunar. O programa elaborado ficou conhecido como NASTRAN (THOMPSON, 2017, p. 4).

Enquanto a NASA estava focada na exploração lunar, a Westinghouse Electric Corporation estava desenvolvendo reatores nucleares para propulsão espacial e produção de energia não convencional. O Dr. John Swanson, então funcionário do Westinghouse Astronuclear Labs em Pittsburgh, acreditava que um programa integrado de elementos finitos de uso geral economizaria tempo e dinheiro ao fazer esses tipos de cálculos. Ele começou a desenvolver um programa desse tipo, chamado STASYS, para a Westinghouse em 1969 (THOMPSON, 2017, p. 4).

Em 1970, John Swanson deixou a Westinghouse e fundou a Swanson Analysis Systems, Inc., onde continuou a desenvolver um programa comercial de elementos finitos de uso geral que ele chamou de ANSYS. A versão original do ANSYS continha 40 elementos de vários tipos (molas, amortecedores, vigas, blocos, etc.), incluindo vários elementos com graus de liberdade térmicos. A Westinghouse tornou-se o primeiro cliente da ANSYS no final do ano. O programa foi rapidamente adotado por outras empresas e tornou-se o programa de elementos finitos padrão para grande parte da indústria de energia. Hoje, os produtos ANSYS são usados em todos os principais campos de engenharia, incluindo aeroespacial, automotivo, processamento químico, construção, bens de consumo, eletrônicos, energia, saúde, offshore, marítimo e indústrias de materiais (THOMPSON, 2017, p. 4).

Independentemente do software utilizado, a qualidade dos resultados dependerá da qualidade do modelo idealizado. Isso inclui a precisão das propriedades do material, a adequação dos modelos de material e assim por diante. Um programa de elementos finitos não comentará sobre a validade de quaisquer suposições feitas na configuração do modelo, desde que as leis físicas não sejam violadas. Também não garantirá que você esteja usando as leis físicas corretas para um determinado problema. O software de elementos finitos é apenas uma ferramenta. (THOMPSON, 2017, p. 3)

5 METODOLOGIA

5.1 Aspectos gerais

Para a obtenção das frequências naturais e modos de vibração de treliças planas via MEF, foram seguidas algumas etapas.

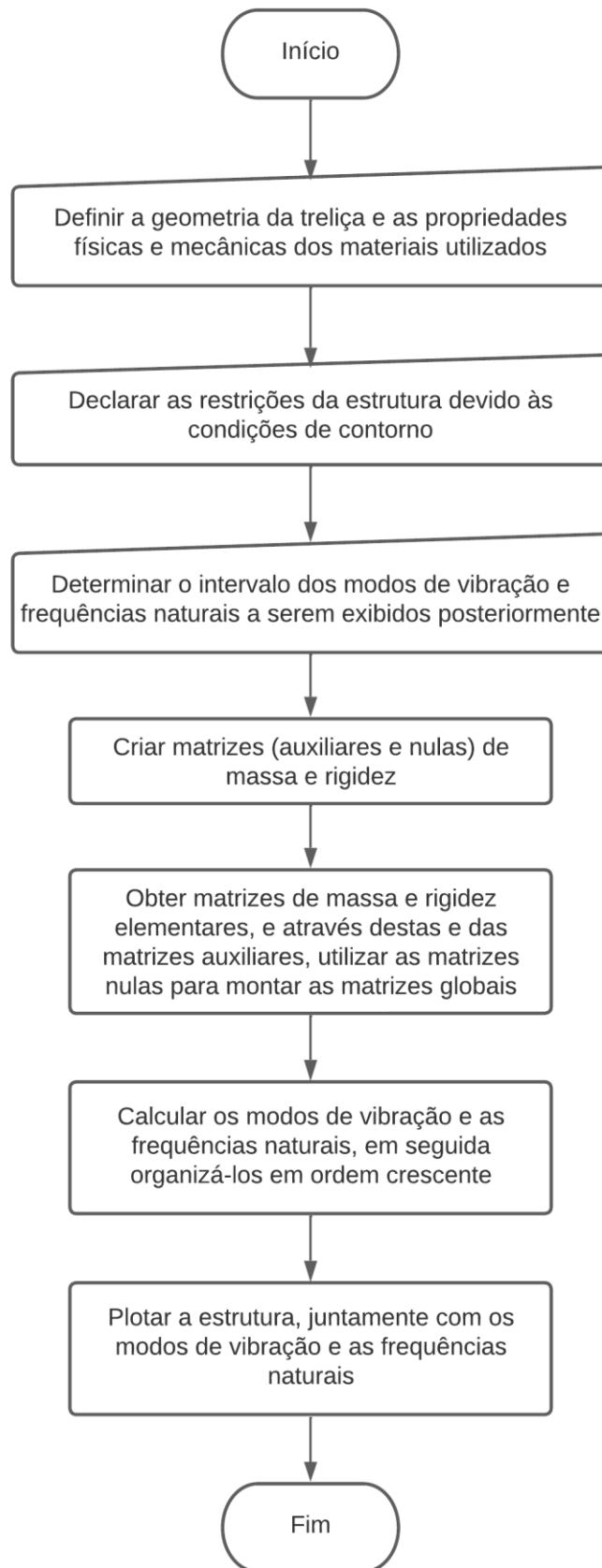
A primeira etapa baseou-se na realização de estudos teóricos acerca do tema abordado. Inicialmente, foi analisado o caso de sistemas de um grau de liberdade. Em seguida, foi deduzido o sistema de equações para o caso mais geral, que consiste em sistemas com múltiplos graus de liberdade. A partir disso, foi possível generalizar para o caso específico das treliças planas.

Para a Análise Modal das treliças planas, foram definidas as matrizes de massa e rigidez elementares, que compõem as matrizes de massa e rigidez de todo o sistema estrutural. De posse das matrizes de massa e rigidez do sistema, foi possível obter as frequências naturais e os modos de vibração, que são determinados através da resolução de um problema de autovalor e autovetor.

5.2 Aspectos computacionais

5.2.1 Algoritmo implementado

A partir das etapas descritas no item 5.1 deste trabalho, foi elaborado um código computacional no software MATLAB capaz de apresentar as frequências naturais de vibração e apresentar, graficamente, os modos de vibração de treliças planas. O algoritmo utilizado na elaboração do código pode ser visualizado no fluxograma apresentado na Figura 7, sendo o código descrito no Apêndice A deste trabalho.

Figura 7 - Fluxograma para elaboração do código

Fonte: Autoria própria (2022)

5.2.2 Desenvolvimento do código computacional

O código computacional foi elaborado a partir do algoritmo apresentado anteriormente. Para sua concepção, fez-se uso do software MATLAB R2022a

5.2.3 Utilizando o programa desenvolvido

Através da ferramenta App Designer disponível no software, é possível criar aplicativos com interface gráfica de maneira fácil e prática. Na Figura 8 é possível visualizar a interface gráfica do código computacional desenvolvido neste trabalho.

Figura 8 - Interface gráfica do código computacional

ANÁLISE MODAL DE TRELIÇAS PLANAS VIA MÉTODO DOS ELEMENTOS FINITOS

Primeiramente, como gostaria de inserir as informações nas Tabelas 1, 2 e 3? Carregar arquivo ▾

Quantidade de barras

Quantidade de nós

Tabela 1

Barras	Nó inicial	Nó final
1	1	8
2	8	10

D:\Felipe\Estudos\Enge

Tabela 2

Nós	x (m)	y (m)
1	0	
2	1	

D:\Felipe\Estudos\Ei

Restrições ao deslocamento

GDL Restringidos
1
2

As propriedades abaixo são constantes para todas as barras? Escolha ▾

Tabela 3

Barras	Área (cm ²)	Módulo de elasticidade (GPa)	Massa específica (kg/m ³)
1.0000	136.5000	14.5000	750.0000
2.0000	136.5000	14.5000	750.0000
3.0000	136.5000	14.5000	750.0000

D:\Felipe\Estudos\Engenharia Civil\Faculdade\Engenharia Civil - 10º P

O que deseja calcular? Modos de Vibração ▾

Modos de vibração a serem exibidos

Limite inf.

Limite sup.

Calcular!

Estrutura

Atualizar

Fonte: Autoria própria (2022)

Primeiramente, como é possível verificar na Figura 8, a primeira informação que é solicitada é a forma que os dados serão inseridos.

Se a opção escolhida for “carregar arquivo”, será habilitado as opções “Procurar” para carregar os dados referente às Tabelas 1, 2 e 3, que se referem às

informações geométricas da estrutura e às propriedades físicas e mecânicas dos materiais. Os arquivos a serem inseridos deverão estar no formato “xlsx”.

Caso a opção escolhida seja “Manualmente”, então as Tabelas 1 e 2 serão geradas a partir dos dados “Quantidade de Barras” e “Quantidade de nós”, com o número correto de linhas de acordo com o que foi preenchido anteriormente; neste cenário, as opções “Procurar” não serão habilitadas. Para o preenchimento da Tabela 3, será possível responder “Sim” ou “Não” para a pergunta “As propriedades abaixo são constantes para todas as barras?”, o que irá gerar uma tabela com a quantidade de linhas proporcional à resposta adotada.

Após preencher as Tabelas 1 e 2, será possível visualizar a estrutura à direita clicando no botão “Atualizar”. Essa funcionalidade permite que o usuário certifique-se que os dados foram inseridos corretamente antes de calcular os modos de vibração ou as frequências naturais (trata-se de uma ferramenta opcional).

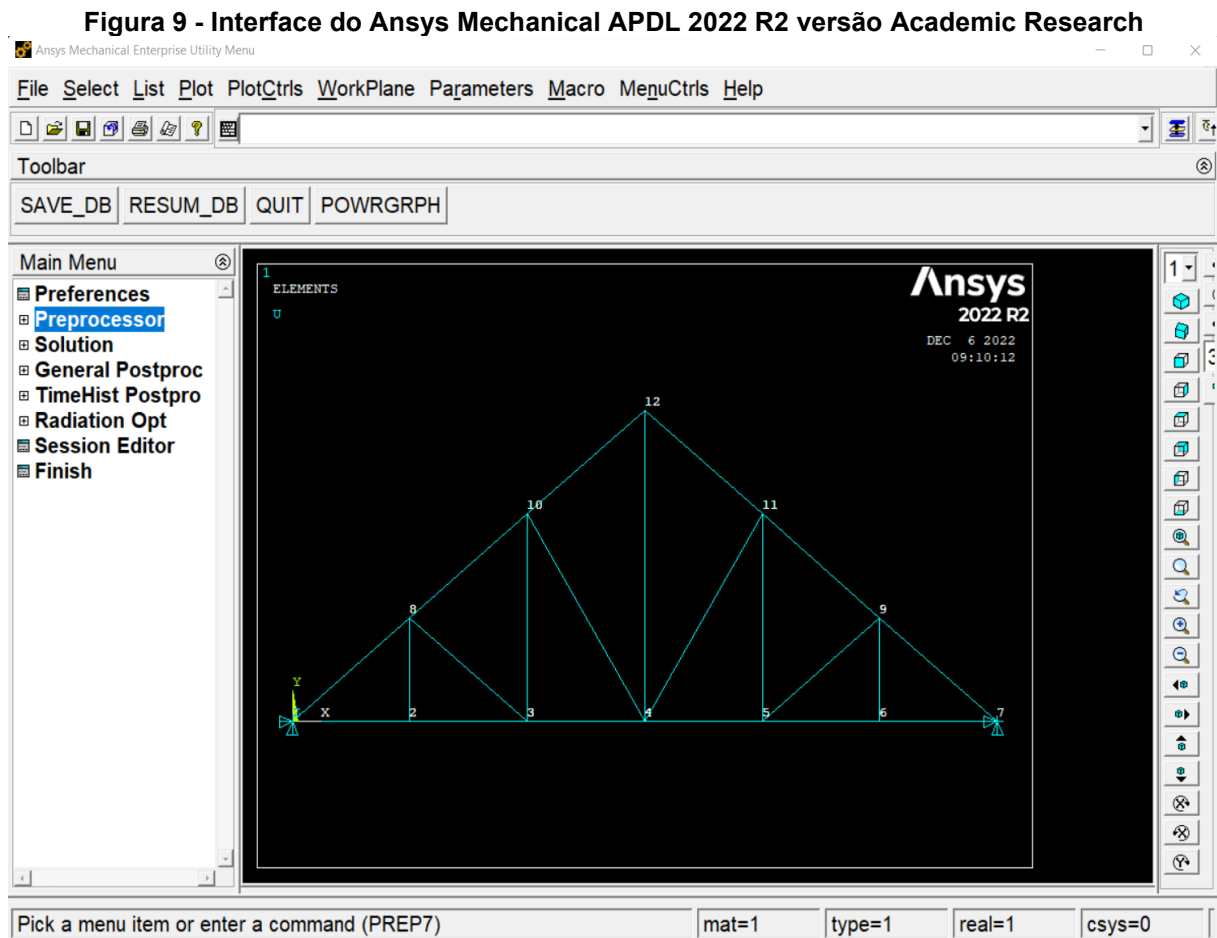
Ao escolher uma das opções para a pergunta “O que deseja calcular?”, será habilitado a opção calcular. Se a opção escolhida for “Modos de Vibração”, então os campos imediatamente abaixo poderão ser preenchidos com o intervalo dos modos de vibrar que se deseja calcular. Caso a opção escolhida seja “Frequências Naturais”, então os campos abaixo não serão habilitados.

Por fim, após certificar-se que os dados foram todos preenchidos corretamente, será possível calcular os modos de vibração e as frequências naturais correspondentes a cada modo ou somente as frequências naturais por meio do botão “Calcular!”.

Após calculados, os resultados serão exibidos em uma ou mais janelas. Caso a opção de cálculo escolhida seja “Frequências Naturais”, então será exibida uma única janela com todas as frequências naturais da estrutura organizadas em ordem crescente. Se a opção escolhida for “Modos de Vibração”, será possível visualizar todos os modos de vibração (juntamente com a estrutura indeformada) dentro do intervalo selecionado; será exibida uma janela individual para cada modo de vibrar. Na Command Window, será possível visualizar as frequências naturais tanto em Hz quanto em rad/s, como também uma tabela com os valores numéricos de todos os modos de vibração normalizados.

5.2.4 Utilizando Ansys Mechanical APDL

No presente trabalho, para validar o código computacional desenvolvido, foi utilizado o software Ansys Mechanical APDL 2022 R2 versão Academic Research. A interface utilizada, ilustrada na Figura 9, permitiu um lançamento de maneira mais simples e intuitiva.



Fonte: Autoria própria (2022)

Ao contrário do pacote Ansys Workbench que informa as unidades de medidas a serem preenchidas, a versão Mechanical APDL não define quais unidades deverão ser inseridas para cada grandeza física, deixando essa tarefa para o usuário. Dessa forma, é necessário que o usuário seja consistente em relação às unidades de medida utilizadas.

A análise modal de treliças planas por meio do Ansys é possível através dos seguintes passos:

1. Deve-se adotar a opção “Structural” para disciplina individual a ser exibida (Main Menu > Preferences);
2. Como tipo de elemento, foi selecionada a opção LINK 180, em que é admitida uma barra reta, com dois nós para cada elemento, carregada axialmente em suas extremidades e com propriedades uniformes ao longo de toda a barra (Main Menu > Preprocessor > Element Type > Add/Edit/Delete > Add...);
3. As propriedades do materiais definidas foram apenas o módulo de elasticidade longitudinal e a densidade (Main Menu > Preprocessor > Material Props > Material Models), visto que são as propriedades que realmente importam na análise que será realizada;
4. Em seguida, foi criada uma seção transversal para ser utilizada posteriormente (Main Menu > Preprocessor > Sections > Link > Add);
5. Antes de inserir as barras da treliça, primeiro foram gerados todos os nós necessários (Main Menu > Preprocessor > Modeling > Create > Nodes > In active CS);
6. A partir dos nós foi possível inserir todas as barras da treliça (Main Menu > Preprocessor > Modeling > Create > Elements > Auto Numbered > Thru Nodes);
7. As condições de contorno foram definidas nesta etapa. A opção utilizada para apoio fixo foi “All DOF” (Main Menu > Preprocessor > Loads > Define Loads > Apply > Structural > Displacement > On Nodes). Caso deseje-se utilizar apoios móveis, o deslocamento de duas direções (dentre três disponíveis) deverão ser restringidos;
8. Depois de modelar a estrutura, foi necessário definir a análise modal como tipo de análise (Main Menu > Solution > Analysis Type > New Analysis);
9. Após a definição do tipo de análise, é necessário definir quais modos de vibração serão exibidos e também a frequência mínima a ser exibida. A frequência mínima a ser exibida em todas as análises do presente trabalho foram de 1 Hz (Main Menu > Solution > Analysis Type > Analysis Options);
10. Em seguida, é necessário solicitar o cálculo dos modos de vibração e das frequências naturais (Main Menu > Solution > Solve > Current LS);
11. Por fim, é possível visualizar as frequências naturais calculadas (Main Menu > General Postproc > Read Results > By Pick) ou visualizar os modos

de vibração (Main Menu > General Postproc > Plot Results > Deformed Shape);

6 ANÁLISE NUMÉRICA

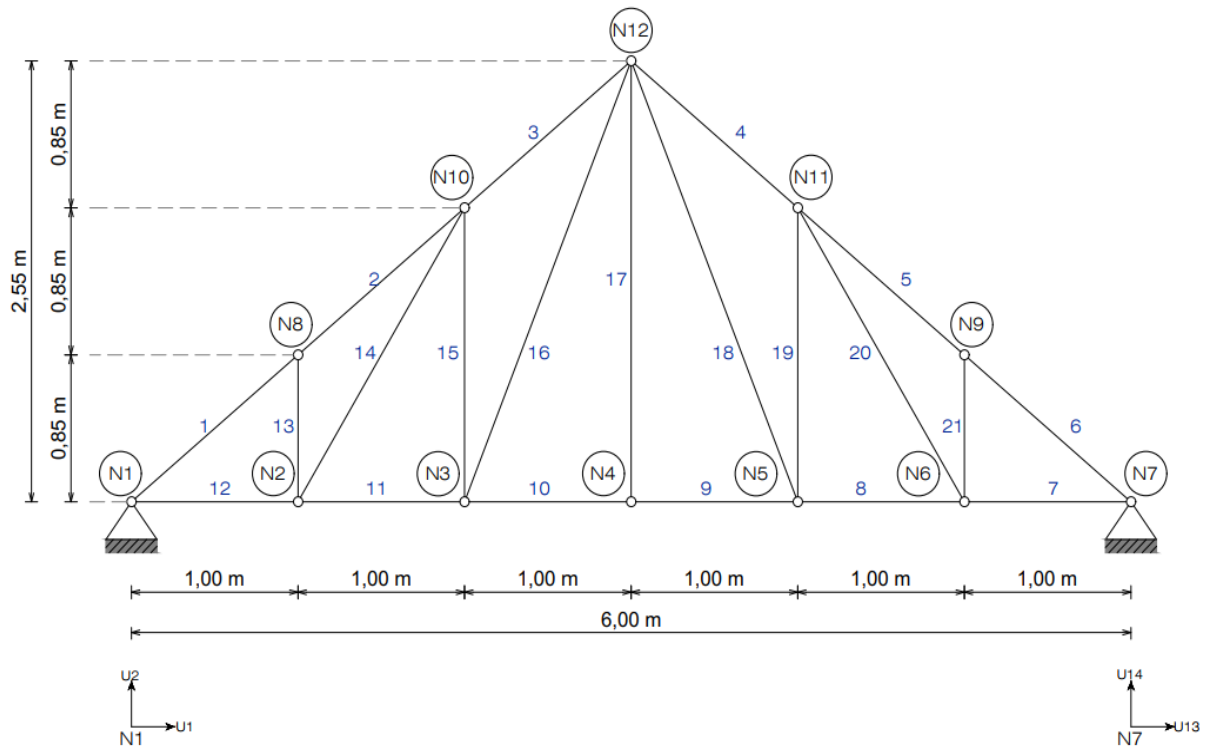
Visando validar o código computacional implementando, apresenta-se, neste item, simulações numéricas realizadas onde são apresentadas as frequências naturais e os modos de vibração de três tipos de treliças bastante comuns na Engenharia Civil, que são as treliças Howe, Pratt e Warren.

Os resultados fornecidos pelo código desenvolvido e pelo Ansys Mechanical APDL 2022 R2 versão Academic Research serão comparados para cada um dos três exemplos apresentados. Serão analisadas e comparadas todas as frequências naturais das treliças; para os modos de vibração, serão levados em consideração apenas os três primeiros.

6.1 Treliça Howe

A primeira treliça a ser analisada está representada na Figura 10 e se trata de uma treliça Howe (HIBBELER, 2015, p. 85). A estrutura conta com 12 nós, 21 elementos de barra e 24 graus de liberdade, sendo que 4 deles estão restringidos (U_1 , U_2 , U_{13} e U_{14} – deslocamentos horizontal e vertical dos nós 1 e 7, respectivamente). A área da seção transversal é invariável, sendo $A = 136,5 \text{ cm}^2$ para todas as barras; as propriedades físicas e mecânicas do material empregado em todas as barras são $E = 14,5 \text{ GPa}$ e $\rho = 750 \text{ kg/m}^3$, referentes à madeira classe D40 encontradas na ABNT NBR 7190-1:2022.

Figura 10 - Treliça Howe



Fonte: Autoria própria (2022)

Para esta estrutura, há 20 modos de vibração e 20 frequências naturais a serem analisados.

6.1.1 Frequências naturais da treliça Howe

As frequências naturais da treliça Howe podem ser observadas na Tabela 1, em que é possível verificar as frequências obtidas pelo código desenvolvido no MATLAB e também as frequências obtidas através do software Ansys Mechanical APDL 2022 R2. À direita, há uma coluna em que consta a discrepância das frequências analisadas para cada um dos modos de vibração.

Tabela 1 - Frequências naturais da treliça Howe

Modo	Frequência Natural (Hz)		Discrepância (%)
	Ansys	Código Desenvolvido	
1	56,5569	56,5569	0,00E+00
2	94,6350	94,6350	0,00E+00
3	146,9910	146,9912	1,36E-04
4	194,6670	194,6665	2,57E-04
5	218,2420	218,2419	4,58E-05
6	235,6250	235,6246	1,70E-04
7	301,0220	301,0225	1,66E-04
8	379,3030	379,3026	1,05E-04
9	496,0470	496,0471	2,02E-05
10	511,4480	511,4480	0,00E+00
11	561,3350	561,3353	5,34E-05
12	577,1080	577,1084	6,93E-05
13	662,7910	662,7909	1,51E-05
14	792,0900	792,0905	6,31E-05
15	909,8040	909,8043	3,30E-05
16	975,3880	975,3878	2,05E-05
17	1002,2400	1002,2446	4,59E-04
18	1108,1000	1108,0957	3,88E-04
19	1155,3600	1155,3589	9,52E-05
20	1158,9700	1158,9715	1,29E-04

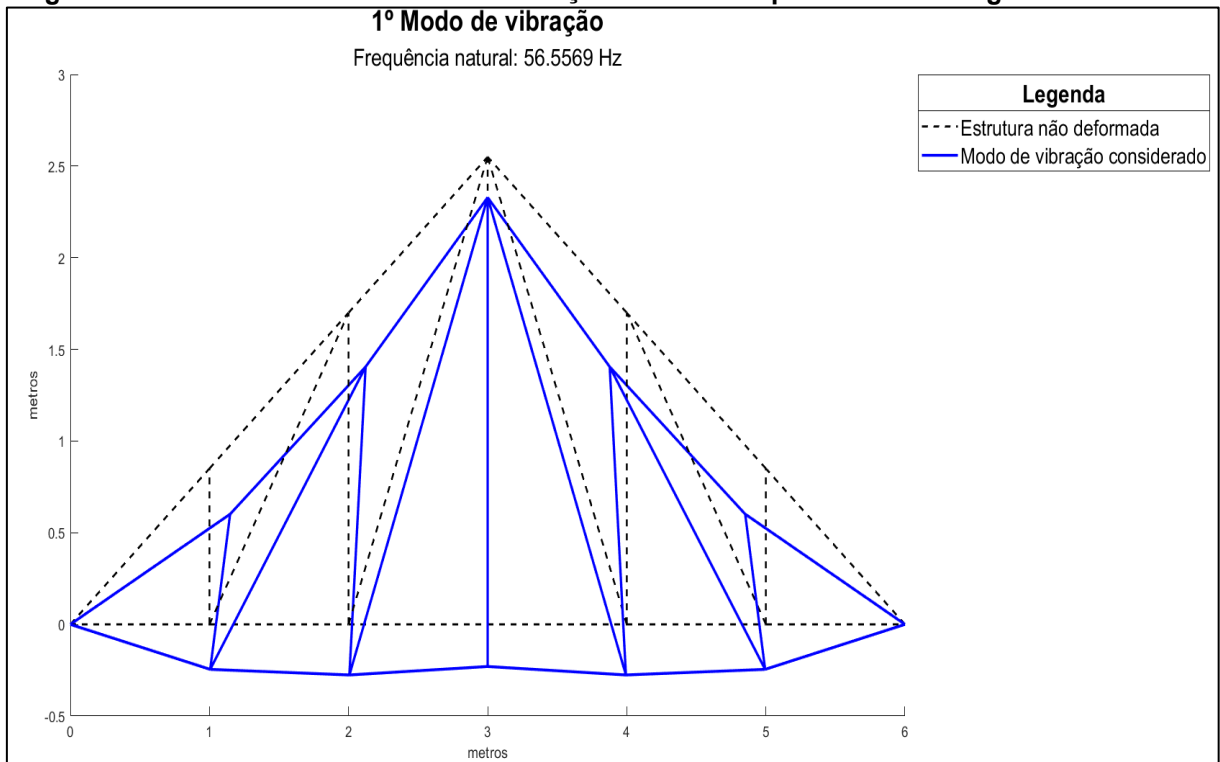
Fonte: Autoria própria (2022)

Como é possível constatar, os resultados obtidos entre os dois softwares são muito próximos, visto que a discrepância entre os valores é baixíssima. O valor médio da discrepância foi de 1,11E-04%.

6.1.2 Modos de vibração da treliça Howe

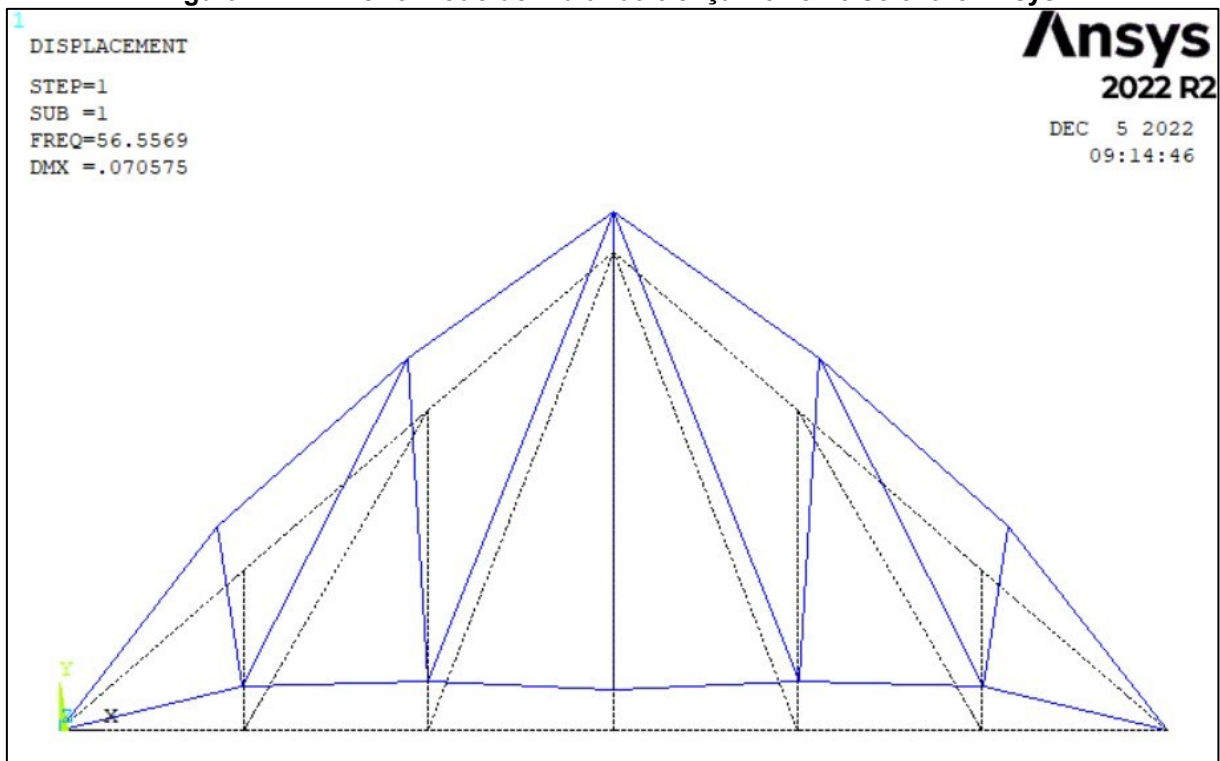
A seguir, serão apresentados os três primeiros modos de vibrar da treliça analisada através das Figuras 11, 12, 13, 14, 15 e 16. Para cada modo de vibrar, primeiramente, serão exibidos os resultados obtidos pelo código desenvolvido, e em seguida os resultados obtidos pelo software Ansys.

Figura 11 - Primeiro modo de vibrar da treliça Howe obtido por meio do código desenvolvido



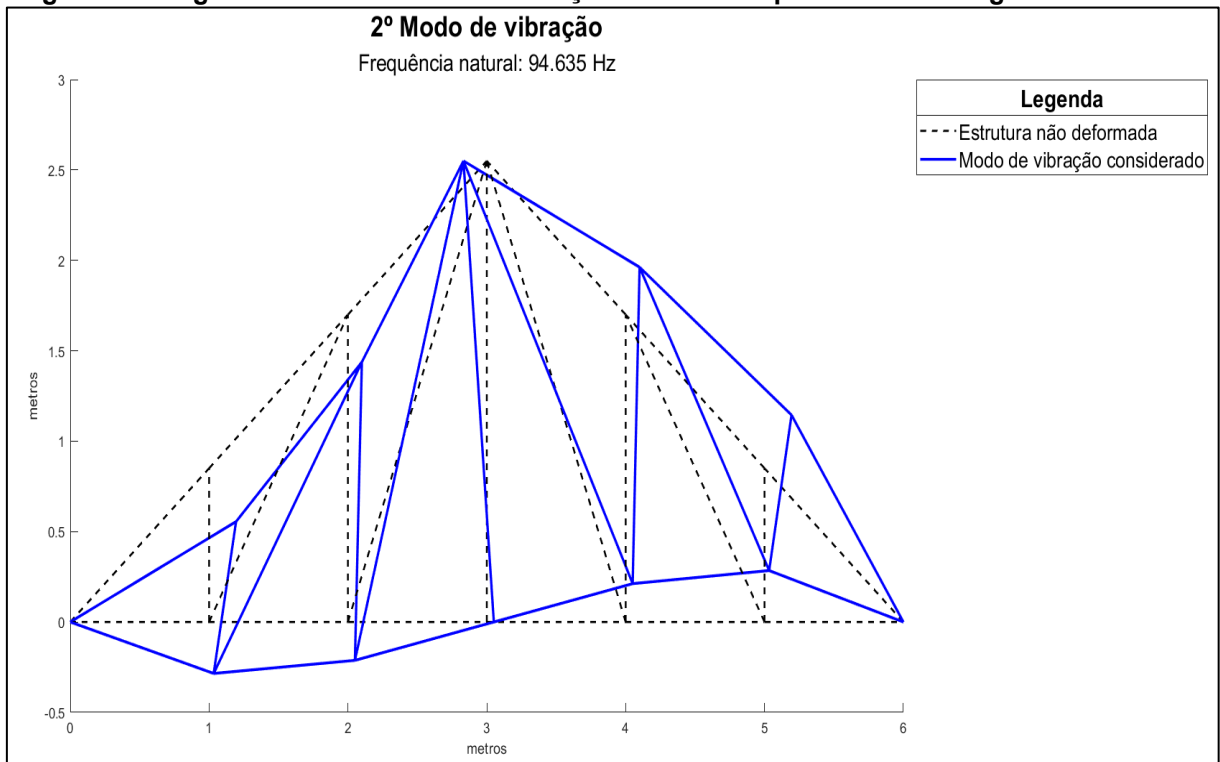
Fonte: Autoria própria (2022)

Figura 12 - Primeiro modo de vibrar da treliça Howe no software Ansys



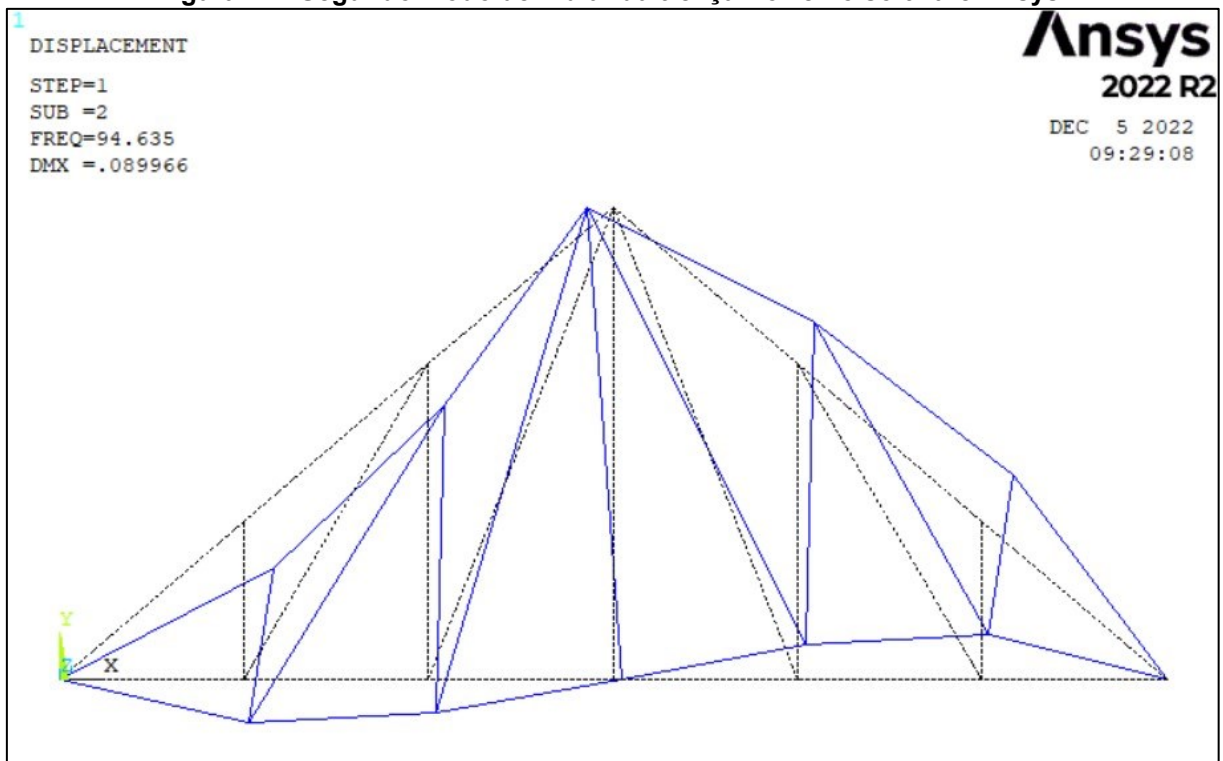
Fonte: Autoria própria (2022)

Figura 13 - Segundo modo de vibrar da treliça Howe obtido por meio do código desenvolvido



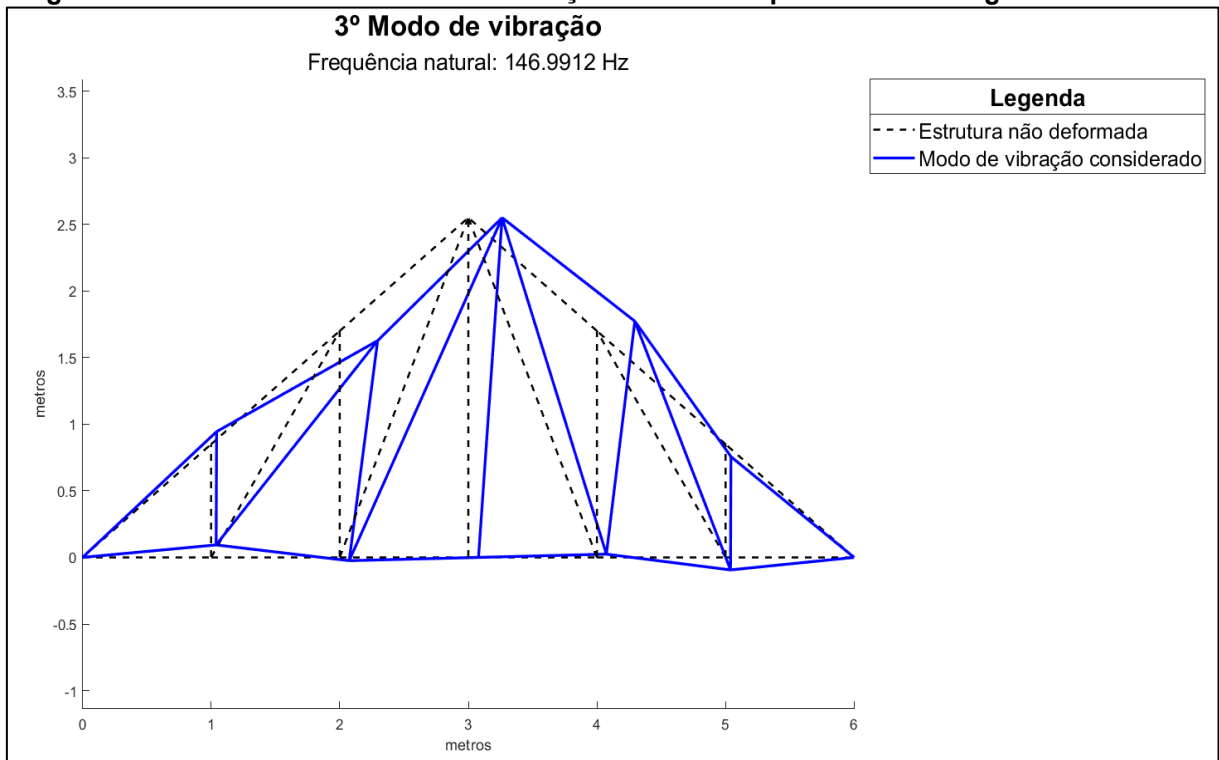
Fonte: Autoria própria (2022)

Figura 14 - Segundo modo de vibrar da treliça Howe no software Ansys



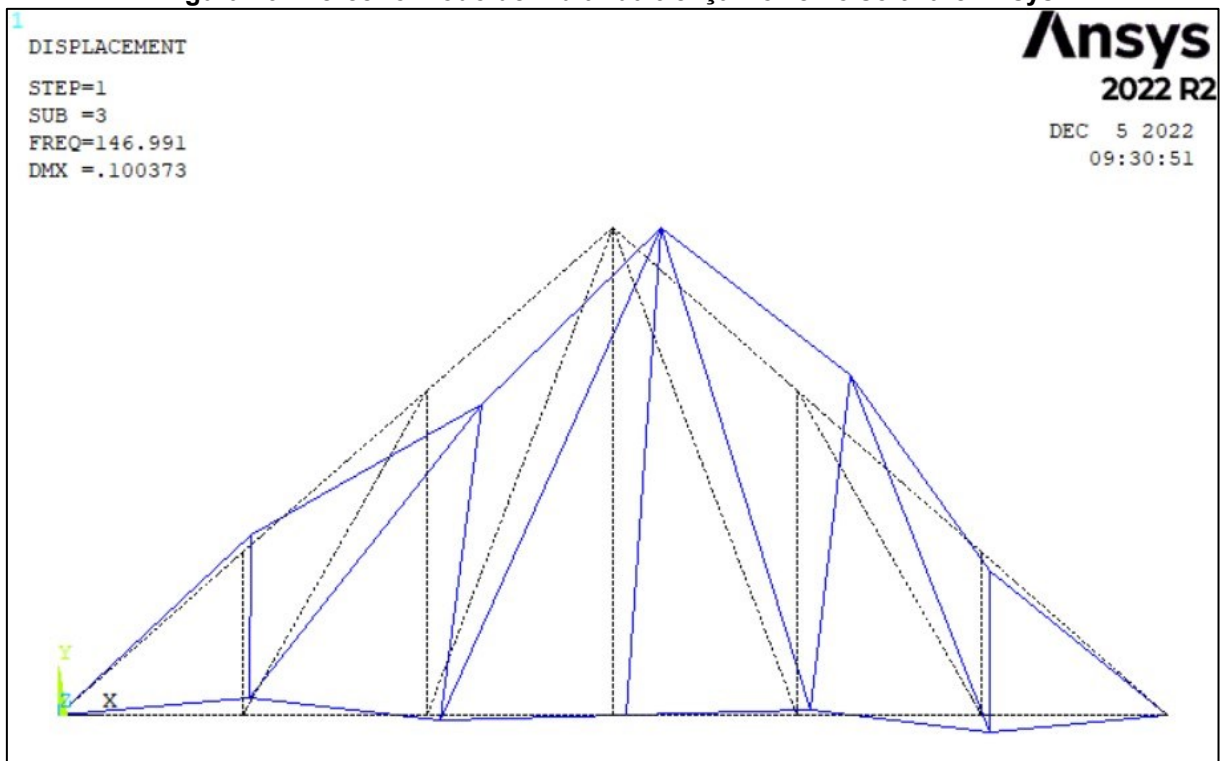
Fonte: Autoria própria (2022)

Figura 15 - Terceiro modo de vibrar da treliça Howe obtido por meio do código desenvolvido



Fonte: Autoria própria (2022)

Figura 16 - Terceiro modo de vibrar da treliça Howe no software Ansys



Fonte: Autoria própria (2022)

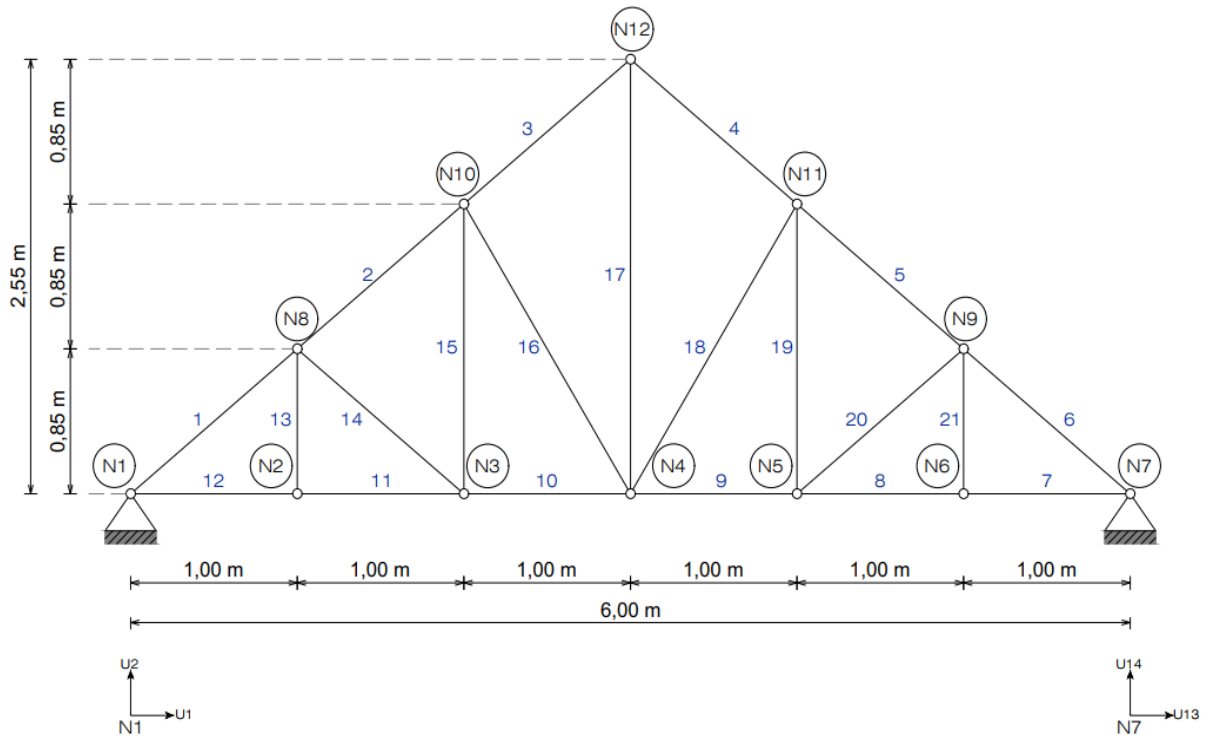
É possível reparar que, aparentemente, o primeiro modo de vibrar obtido pelos softwares divergem totalmente, mas na realidade são iguais. A estrutura tende a vibrar em torno da configuração inicial podendo assumir ambas as formas em momentos distintos. A dessemelhança nas representações gráficas se deve à diferença nos métodos utilizados para normalização dos autovetores.

Os outros modos de vibrar são praticamente idênticos, com uma pequena diferença em função da escala utilizada na plotagem dos resultados. Para o código desenvolvido, cada deslocamento foi multiplicado por uma constante que se aproximasse do que seria obtido pelo software Ansys. É possível obter resultados gráficos ainda mais próximos apenas alterando o valor constante a ser multiplicado por cada deslocamento.

6.2 Treliça Pratt

A segunda treliça a ser analisada é uma treliça Pratt (HIBBELER, 2015, p. 85), que está representada na Figura 17. A estrutura, assim como a treliça Howe analisada no item 6.1, conta com 12 nós, 21 elementos de barra e 24 graus de liberdade, sendo que 4 deles estão restringidos (U_1 , U_2 , U_{13} e U_{14} – deslocamentos horizontal e vertical dos nós 1 e 7, respectivamente). A área da seção transversal é invariável, sendo $A = 136,5 \text{ cm}^2$ para todas as barras; as propriedades físicas e mecânicas do material empregado em todas as barras são $E = 14,5 \text{ GPa}$ e $\rho = 750 \text{ kg/m}^3$, referentes à madeira classe D40 encontradas na ABNT NBR 7190-1:2022.

Figura 17 - Treliça Pratt



Fonte: Autoria própria (2022)

6.2.1 Frequências naturais da treliça Pratt

As frequências naturais da treliça Pratt podem ser visualizadas na Tabela 2.

Tabela 2 - Frequências naturais da treliça Pratt

Modo	Frequência Natural (Hz)		Discrepância (%)
	Ansys	Código Desenvolvido	
1	63,3462	63,3462	0,00E+00
2	127,0170	127,0173	2,36E-04
3	171,1380	171,1377	1,75E-04
4	271,5030	271,5033	1,10E-04
5	283,3830	283,3828	7,06E-05
6	337,2260	337,2261	2,97E-05
7	434,3760	434,3761	2,30E-05
8	472,8310	472,8309	2,11E-05
9	577,3920	577,3916	6,93E-05
10	604,0100	604,0103	4,97E-05
11	666,1180	666,1182	3,00E-05
12	674,6790	674,6793	4,45E-05
13	742,5150	742,5151	1,35E-05
14	841,3680	841,3675	5,94E-05
15	919,5010	919,5007	3,26E-05
16	927,1340	927,1341	1,08E-05
17	1144,2600	1144,2646	4,02E-04
18	1144,6600	1144,6553	4,11E-04
19	1236,9600	1236,9574	2,10E-04
20	1273,0300	1273,0349	3,85E-04

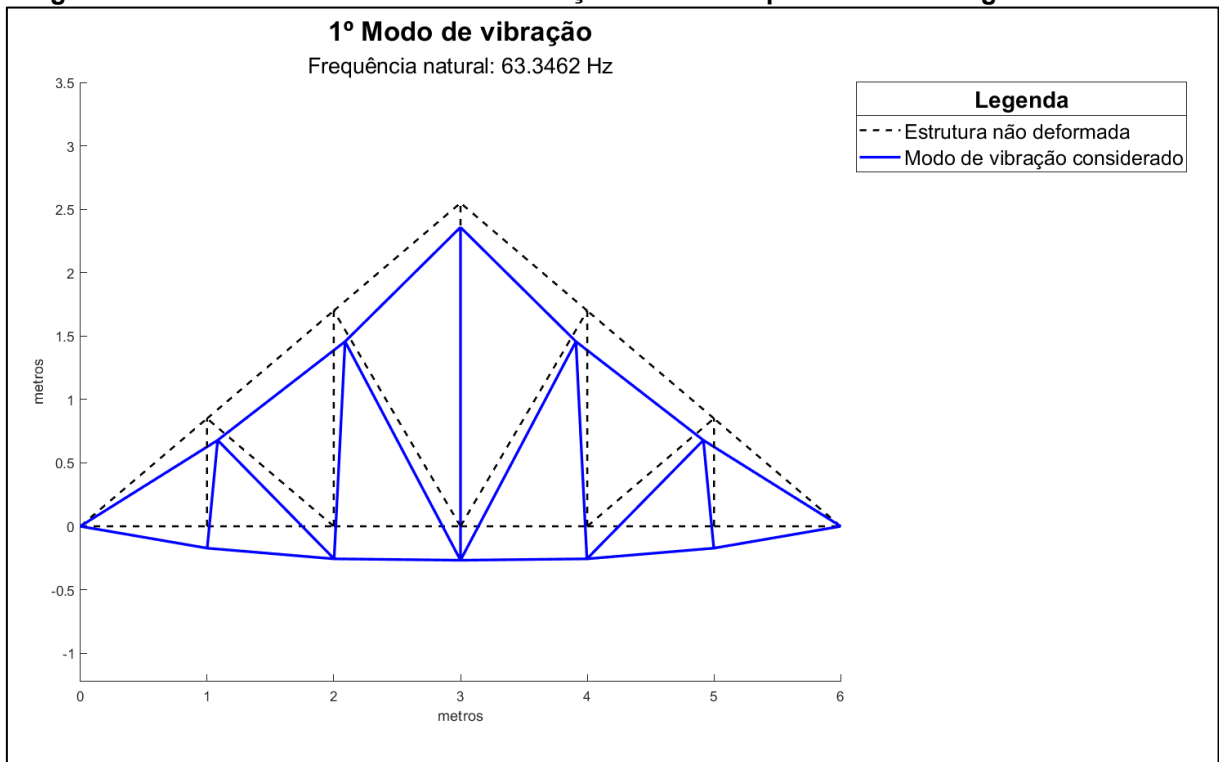
Fonte: Autoria própria (2022)

Como é possível constatar, novamente, os resultados obtidos entre os dois softwares são muito próximos, visto que a discrepância entre os valores é baixíssima. A média das discrepâncias nesse caso é $1,19E-04\%$, valor muito próximo ao obtido anteriormente.

6.2.2 Modos de vibração da treliça Pratt

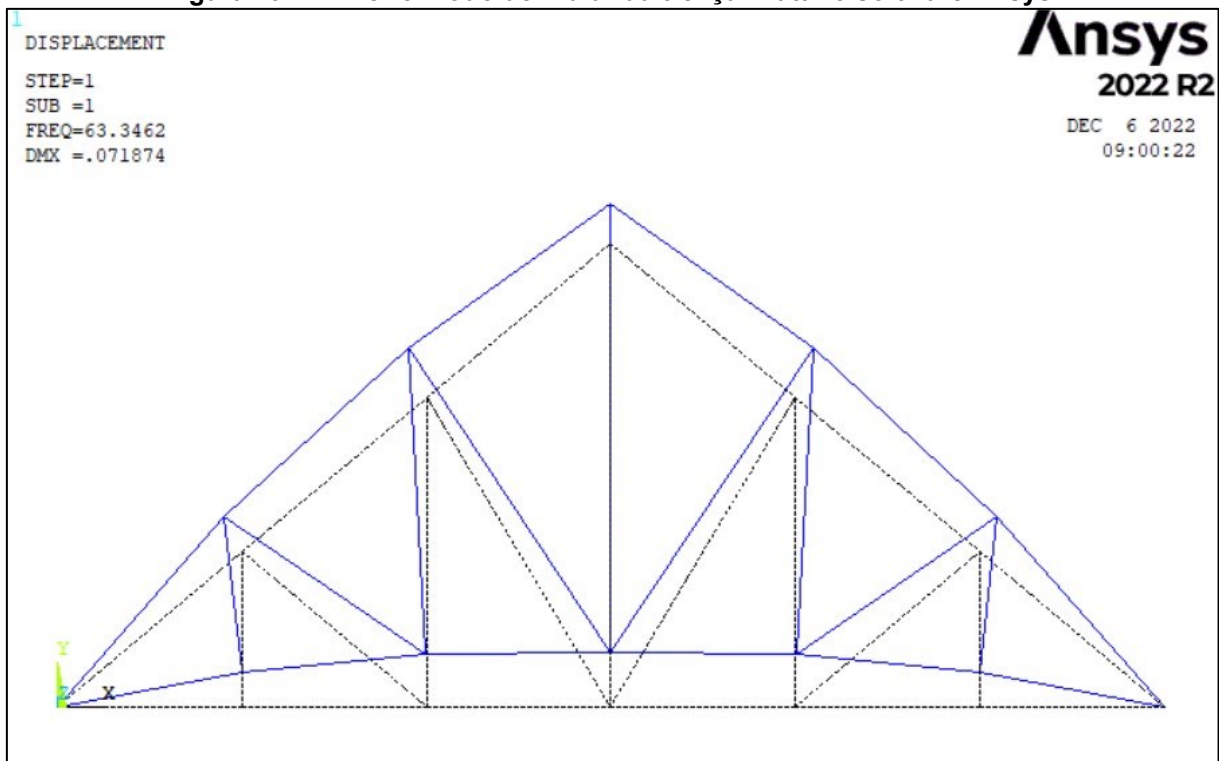
A seguir, serão apresentados os três primeiros modos de vibrar da treliça Pratt por meio das Figuras 18, 19, 20, 21, 22 e 23. Para cada modo de vibrar, primeiramente, serão exibidos os resultados obtidos pelo código desenvolvido, e em seguida os resultados obtidos pelo software Ansys.

Figura 18 - Primeiro modo de vibrar da treliça Pratt obtido por meio do código desenvolvido



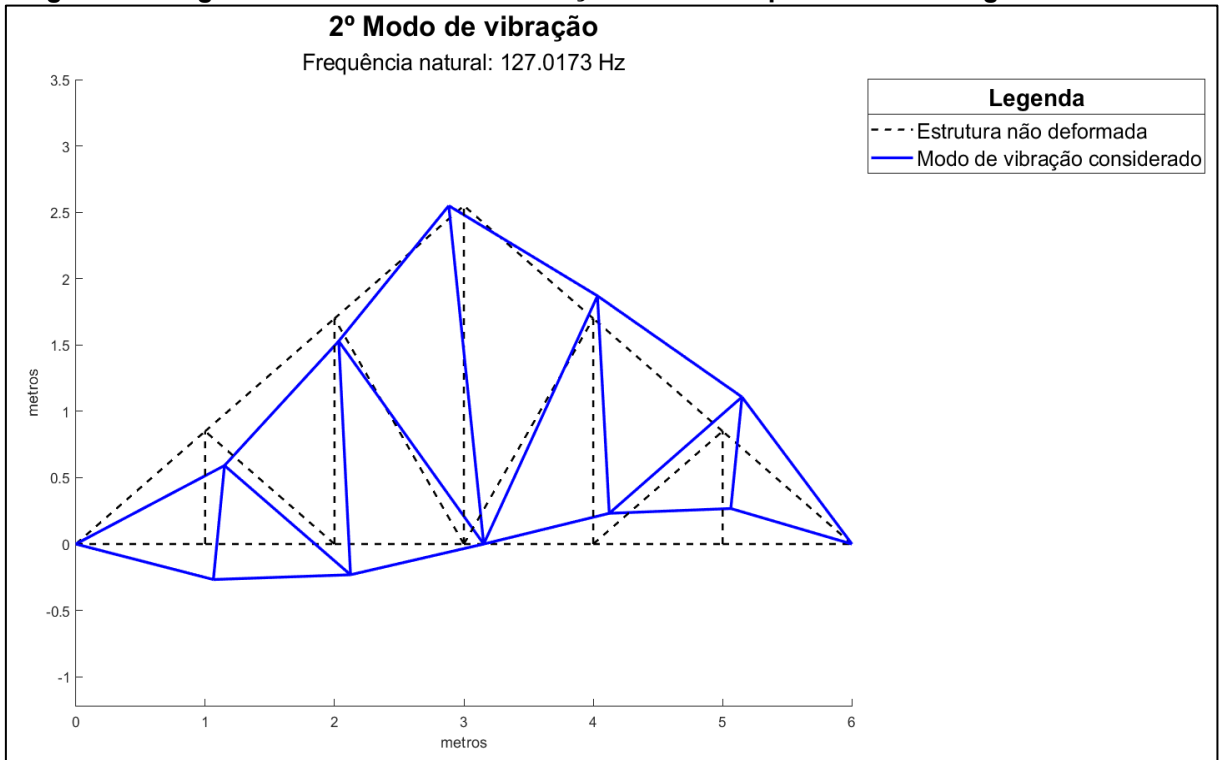
Fonte: Autoria própria (2022)

Figura 19 - Primeiro modo de vibrar da treliça Pratt no software Ansys



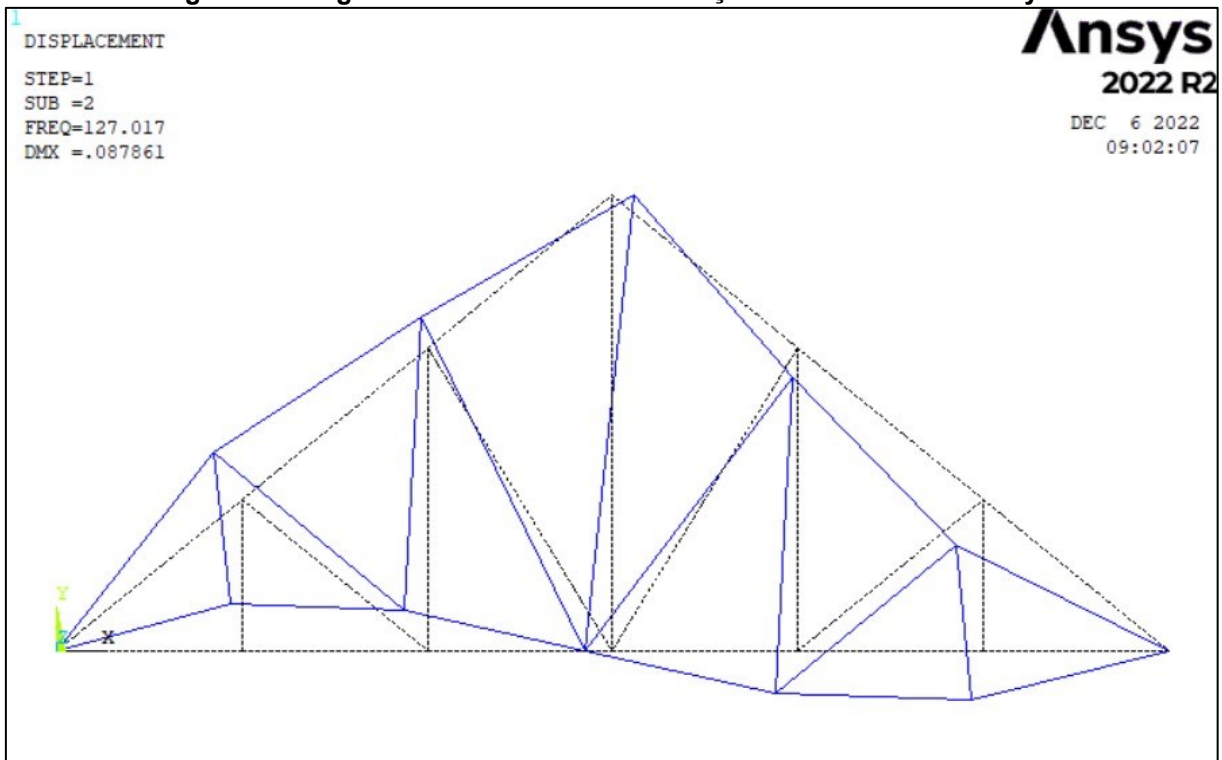
Fonte: Autoria própria (2022)

Figura 20 - Segundo modo de vibrar da treliça Pratt obtido por meio do código desenvolvido



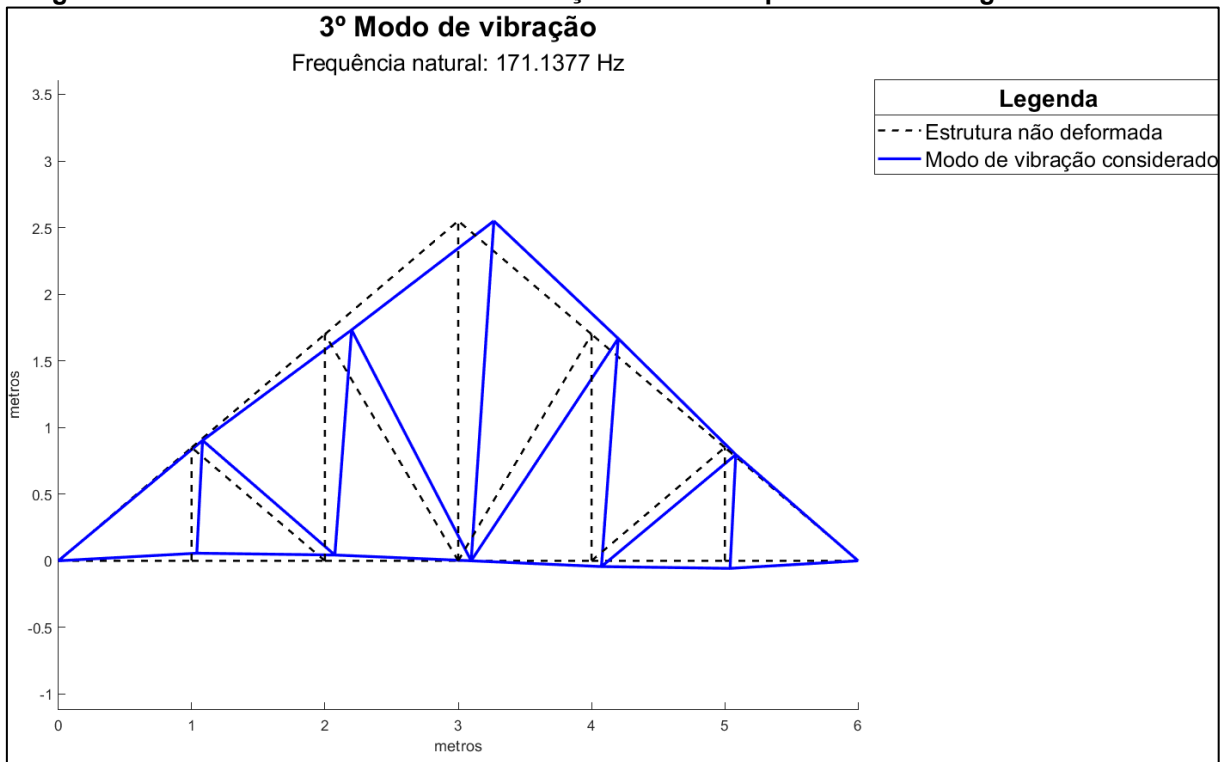
Fonte: Autoria própria (2022)

Figura 21 - Segundo modo de vibrar da treliça Pratt no software Ansys



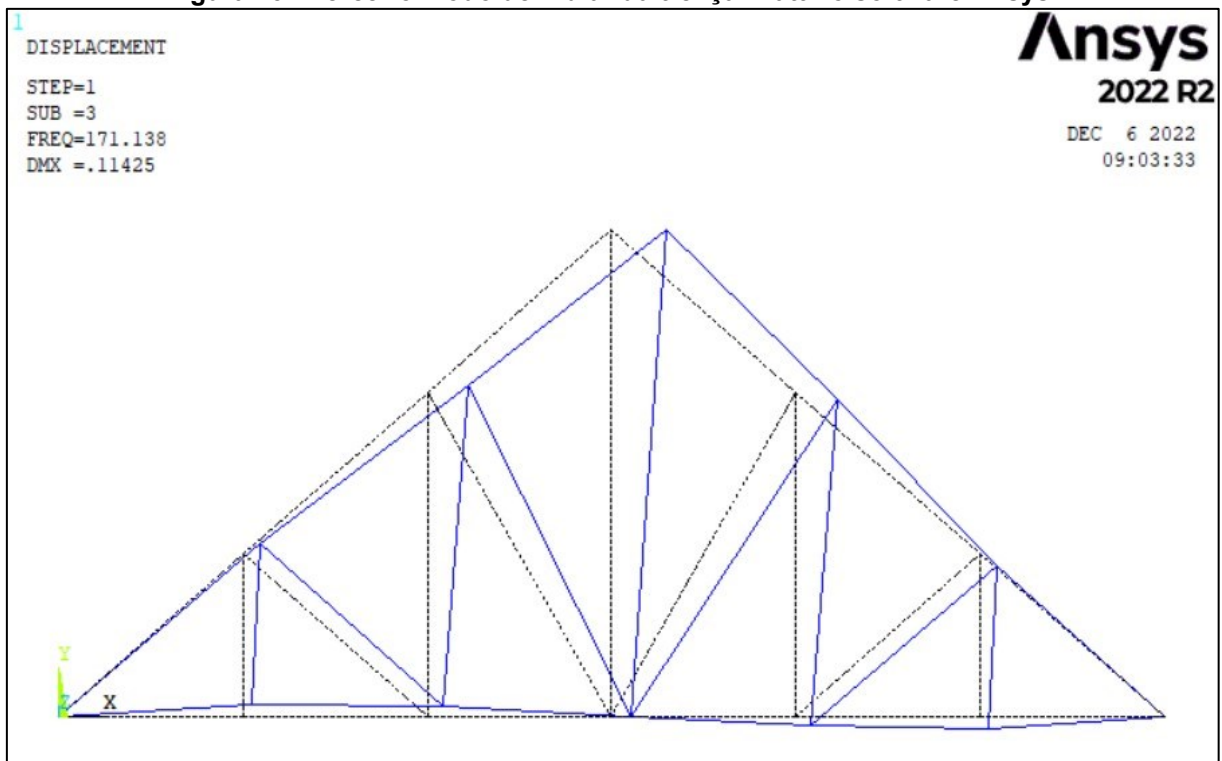
Fonte: Autoria própria (2022)

Figura 22 - Terceiro modo de vibrar da treliça Pratt obtido por meio do código desenvolvido



Fonte: Autoria própria (2022)

Figura 23 - Terceiro modo de vibrar da treliça Pratt no software Ansys



Fonte: Autoria própria (2022)

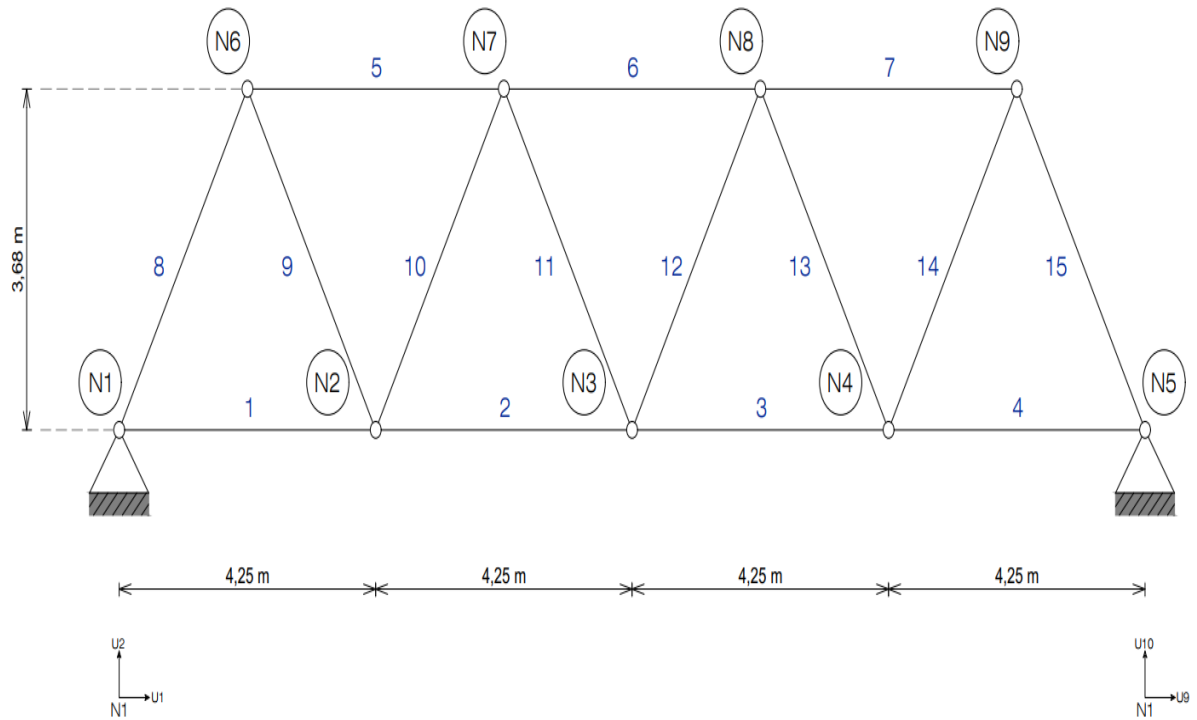
Neste exemplo, os dois primeiros modos de vibrar obtidos pelos softwares têm sentidos opostos. Vale a pena frisar que estrutura tende a vibrar em torno da configuração inicial podendo assumir ambas as formas em momentos distintos, ou seja, ambos os resultados estão corretos. A disparidade nas representações gráficas se deve à diferença nos critérios utilizados nas normalizações dos autovetores.

Ao analisar o terceiro modo de vibrar, percebe-se que ambas as imagens são praticamente iguais, com uma pequena diferença na escala usada para a plotagem dos resultados. Essa diferença visual é possível de ser verificada não pela discrepância dos resultados (por esse motivo não seria possível perceber), mas pela constante utilizada no código MATLAB ao multiplicar cada um dos valores do primeiro modo de vibrar.

6.3 Treliça Warren

Por fim, a última treliça se trata de uma treliça Warren, representada na Figura 24. A estrutura, nesse caso, conta com 9 nós, 15 elementos de barra, 18 graus de liberdade, sendo que 4 deles estão restringidos (U_1 , U_2 , U_9 e U_{10} – deslocamentos horizontal e vertical dos nós 1 e 5, respectivamente). A área da seção transversal continua sendo a mesma para todas as barras, porém agora igual a $A = 50,2 \text{ cm}^2$, e as propriedades físicas e mecânicas do material que compõe a treliça são $E = 200 \text{ GPa}$ e $\rho = 7850 \text{ kg/m}^3$, valores comuns para aço.

Figura 24 - Treliça Warren



Fonte: Autoria própria (2022)

6.3.1 Frequências naturais da treliça Warren

As frequências naturais da treliça Warren obtidas pelo MATLAB e pelo Ansys Mechanical APDL 2022 R2 podem ser conferidas na Tabela 3, juntamente com as discrepâncias nos resultados.

Tabela 3 - Frequências naturais da treliça Warren

Modo	Frequência Natural (Hz)		Discrepância (%)
	Ansys	Código Desenvolvido	
1	33,1149	33,1149	0,00E+00
2	66,9472	66,9472	0,00E+00
3	91,3299	91,3299	0,00E+00
4	148,0450	148,0454	2,70E-04
5	168,3840	168,3843	1,78E-04
6	189,5820	189,5817	1,58E-04
7	218,9000	218,9000	0,00E+00
8	256,0190	256,0194	1,56E-04
9	275,5590	275,5589	3,63E-05
10	289,3280	289,3279	3,46E-05
11	308,0390	308,0386	1,30E-04
12	316,2200	316,2201	3,16E-05
13	382,0280	382,0277	7,85E-05
14	388,2900	388,2904	1,03E-04

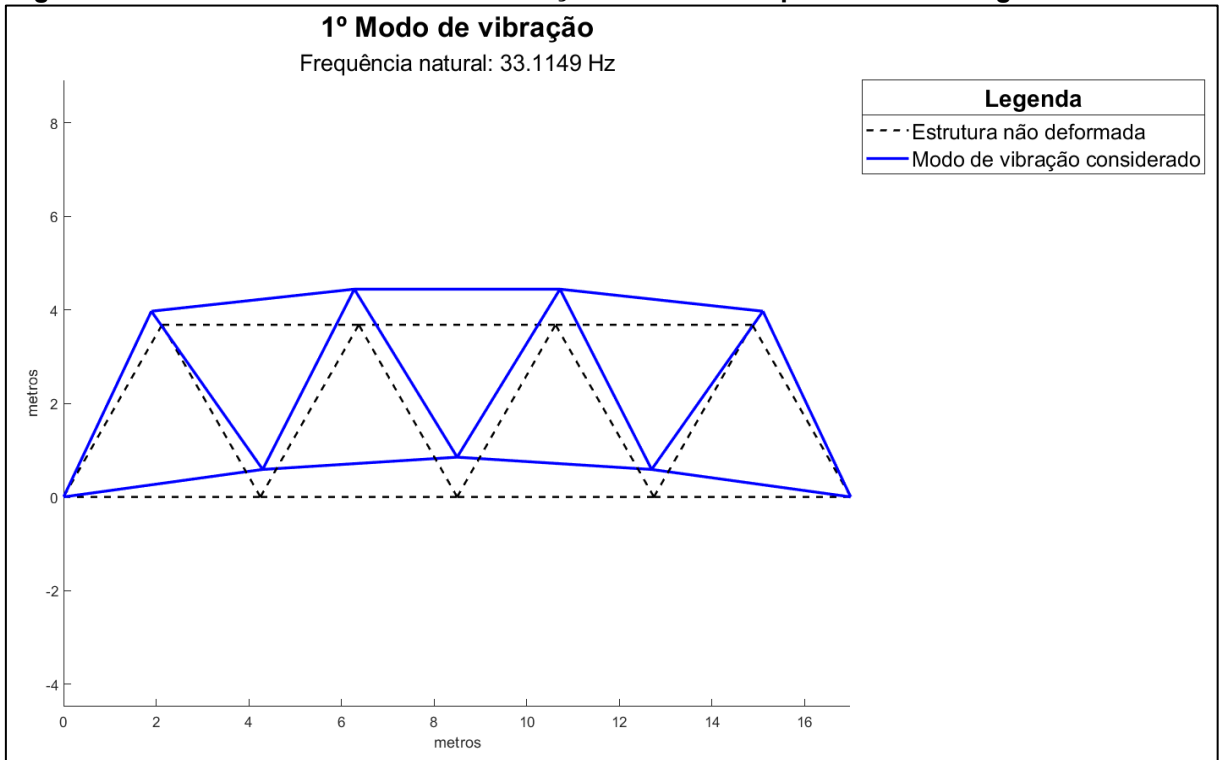
Fonte: Autoria própria (2022)

Dessa vez, a discrepância média é ainda menor, cerca de 8,41E-05%.

6.3.2 Modos de vibração da treliça Warren

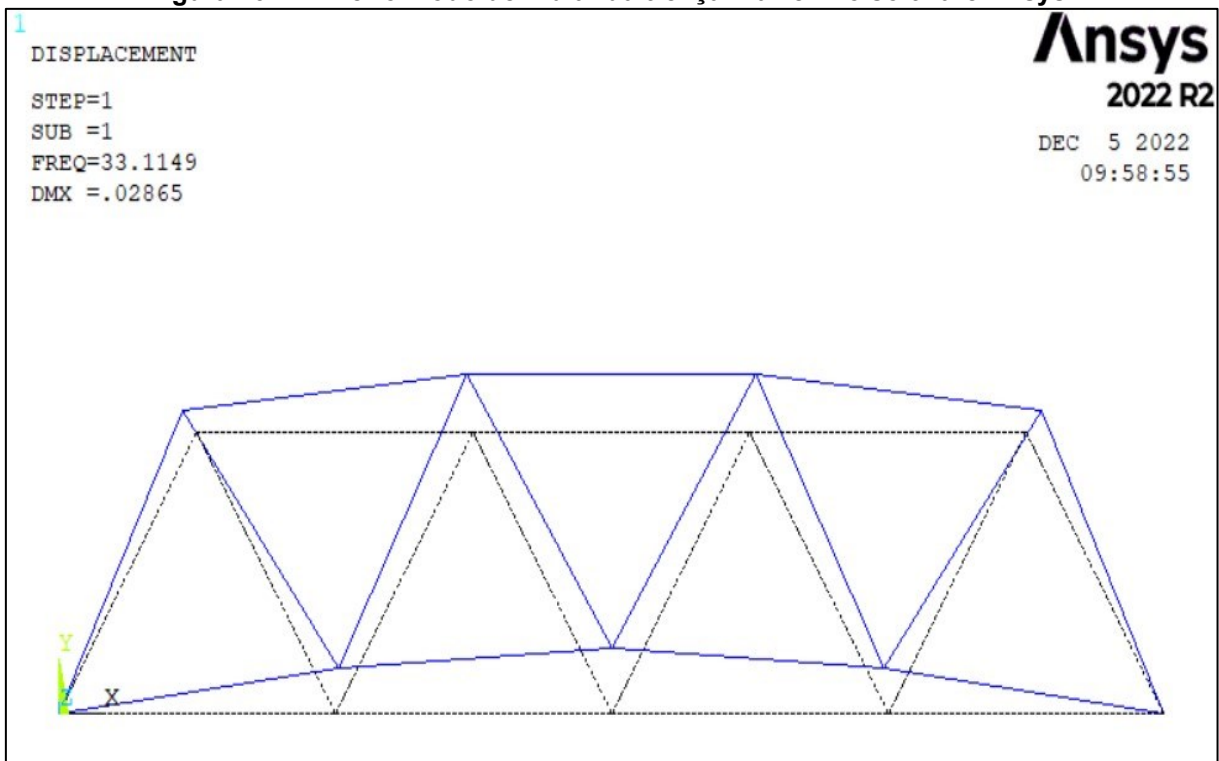
Por fim, serão apresentados os três primeiros modos de vibrar da última treliça (Warren) através das Figuras 25, 26, 27, 28, 29 e 30. Para cada modo de vibrar, primeiramente, serão exibidos os resultados obtidos pelo código desenvolvido no MATLAB, e em seguida os resultados obtidos pelo software Ansys.

Figura 25 - Primeiro modo de vibrar da treliça Warren obtido por meio do código desenvolvido



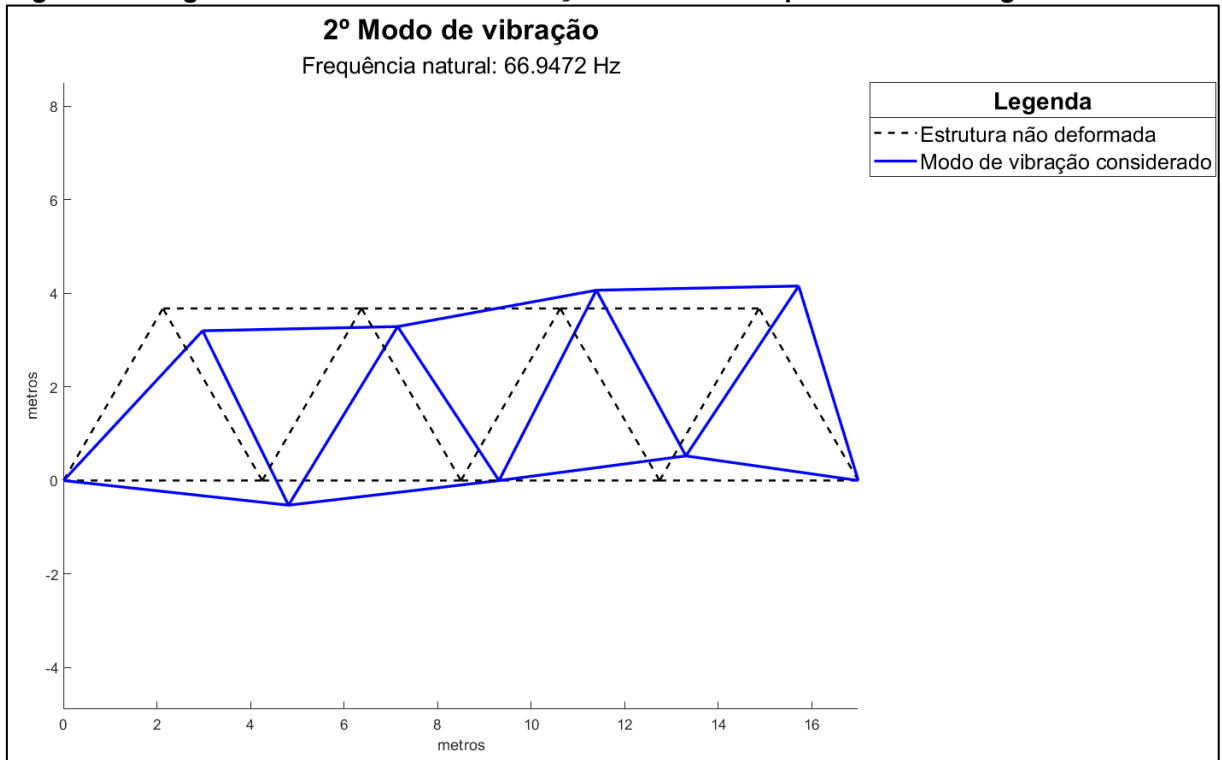
Fonte: Autoria própria (2022)

Figura 26 - Primeiro modo de vibrar da treliça Warren no software Ansys



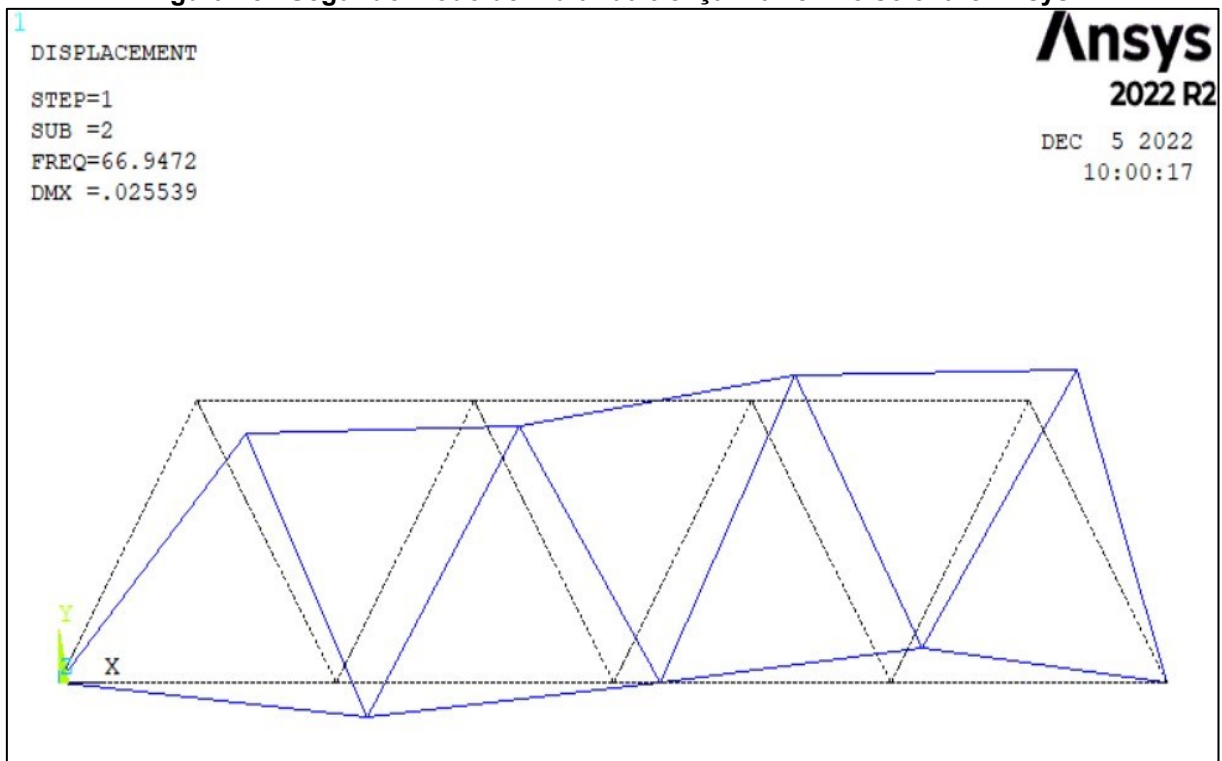
Fonte: Autoria própria (2022)

Figura 27 - Segundo modo de vibrar da treliça Warren obtido por meio do código desenvolvido



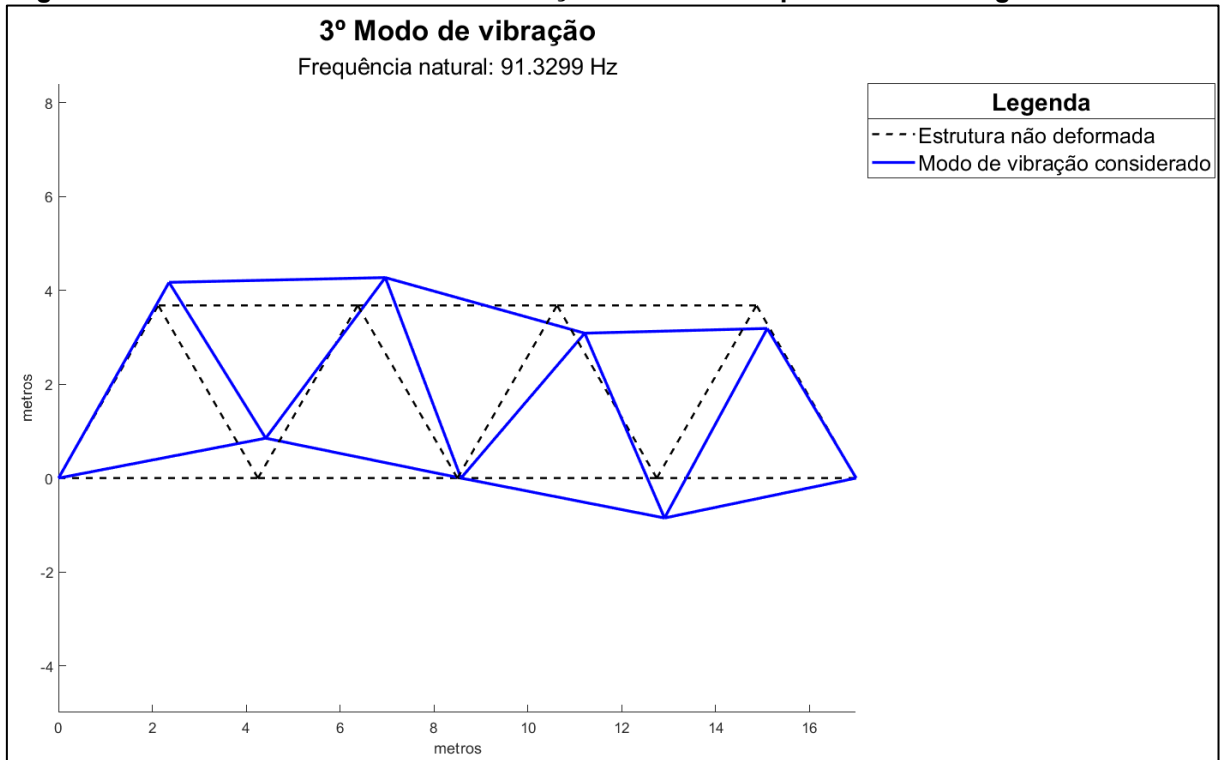
Fonte: Autoria própria (2022)

Figura 28 - Segundo modo de vibrar da treliça Warren no software Ansys



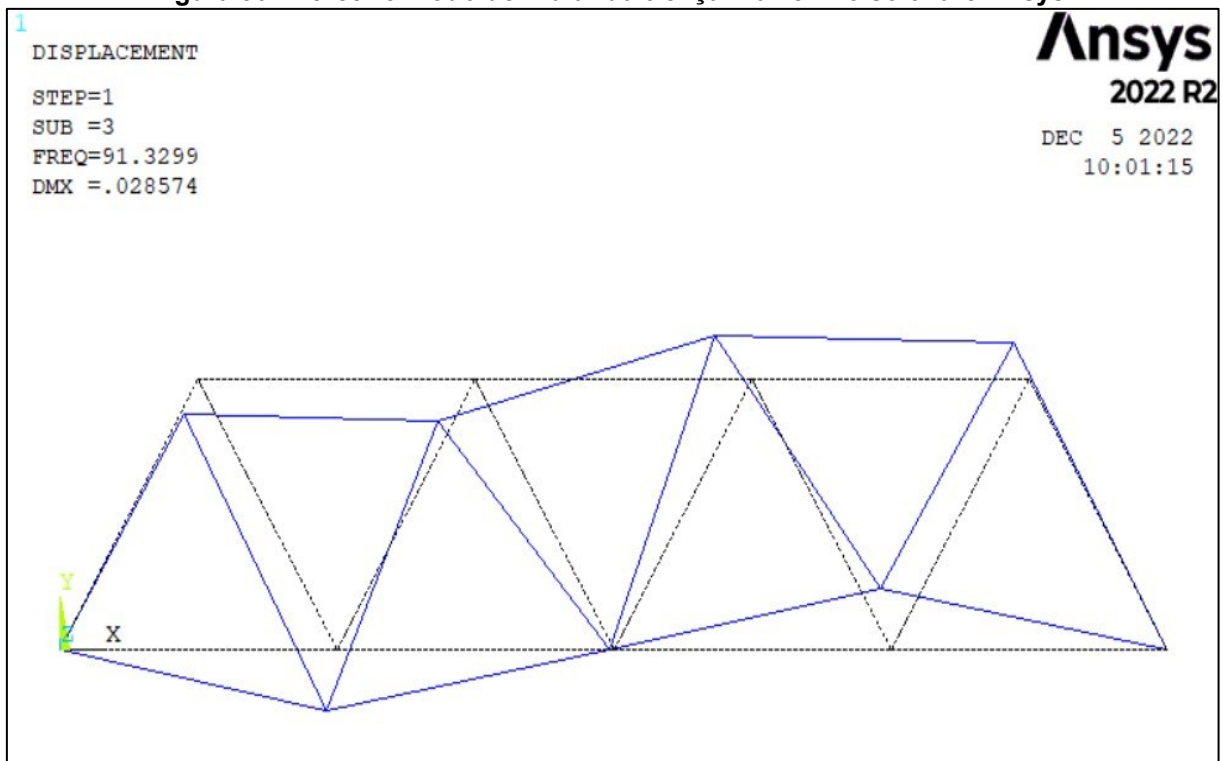
Fonte: Autoria própria (2022)

Figura 29 - Terceiro modo de vibrar da treliça Warren obtido por meio do código desenvolvido



Fonte: Autoria própria (2022)

Figura 30 - Terceiro modo de vibrar da treliça Warren no software Ansys



Fonte: Autoria própria (2022)

Neste último exemplo, assim como já ocorreu anteriormente, um dos modos de vibrar (o terceiro) obtidos têm seus nós deslocados com sentido contrário ao do outro. Novamente, é importante frisar que ambos os resultados estão corretos, e os comentários feitos para as treliças Howe e Pratt também são válidos neste caso.

Mais uma vez, os outros modos de vibrar, graficamente, são praticamente idênticos, com uma pequena diferença em cada um dos graus de liberdade. Essa diferença visual, como já comentado anteriormente, nada tem a ver com a irrelevante discrepância nos resultados.

7 CONCLUSÃO

O objetivo principal a ser alcançado durante este trabalho foi elaborar um código computacional implementado no software MATLAB, com interface gráfica intuitiva, capaz de calcular quaisquer modos de vibração e frequências naturais de treliças planas por meio do Método dos Elementos Finitos. O intuito é oferecer uma ferramenta gratuita para aqueles que estejam iniciando seus estudos nessa área.

Para que o objetivo final fosse atingido, inicialmente, foram apresentadas as deduções matemáticas necessárias para a obtenção das matrizes de massa e de rigidez de treliças planas, fundamentais para calcular tanto os modos de vibração quanto as frequências naturais de vibração.

Foi apresentado um pouco sobre os softwares MATLAB e Ansys Mechanical APDL 2022 R2, com o intuito de familiarizar o leitor com essas incríveis ferramentas disponíveis no mercado. Devido à facilidade no lançamento das treliças planas, foi utilizado a versão Mechanical APDL; no entanto, também poderia ser utilizado o pacote Ansys Workbench para a obtenção dos mesmos resultados.

A interface desenvolvida, foi elaborada pensando em ser simples, intuitiva e que permitisse a inserção de parâmetros como as propriedades físicas, mecânicas e geométricas de cada barra individualizada, os limites inferior e superior de quais modos de vibração seriam exibidos. Foi pensado também na possibilidade de pré-visualizar a estrutura antes de serem exibidos todos os resultados solicitados.

Através dos três exemplos propostos, foi possível verificar que os resultados obtidos pelo código computacional se aproximam muito daqueles obtidos pelo software Ansys. As diferenças nas frequências naturais, por exemplo, podem ser facilmente ignoradas, visto que as discrepâncias são insignificantes. Dessa forma, fica claro que o código desenvolvido foi implementado corretamente, de fato, poderá ser utilizado por aqueles que desejam iniciar ou até mesmo se aprofundar nesta temática.

REFERÊNCIAS

- ABNT – Associação Brasileira de Normas Técnicas. **ABNT NBR 7190-1 – Projeto de estruturas de madeira**. Parte 1: Critérios de dimensionamento. Rio de Janeiro: ABNT, 2022. 81 p.
- ALVES FILHO, A. **Elementos Finitos: A Base da Tecnologia CAE - Análise Dinâmica**. 2. ed. São Paulo: Érica, 2008. ISBN 9978-85-365-1971-5.
- ALVES FILHO, A. **Elementos Finitos: A Base da Tecnologia CAE**. 6. ed. São Paulo: Érica, 2013. ISBN 978-85-365-1970-8.
- BRASIL, R. M. L. R. F. **Introdução à Dinâmica das Estruturas: Para a Engenharia Civil**. 2. ed. rev. São Paulo: Blucher, 2015. ISBN 978-85-212-0910-2.
- CHAPMAN, S. J. **Programação em MATLAB para Engenheiros**. Tradução de Noveritis do Brasil. 5. ed. São Paulo: Cengage Learning, 2016. ISBN 978-85-221-2523-4.
- GILAT, A. **MATLAB com Aplicações em Engenharia**. Tradução de Rafael Silva Alípio. 4. ed. Porto Alegre: Bookman, 2012. ISBN 978-85-407-0187-8.
- HIBBELER, R. C. **Estática: Mecânica para Engenharia**. Tradução de Daniel Vieira. 14. ed. São Paulo: Pearson Education do Brasil, 2017. ISBN 978-85-430-1624-5.
- HIBBELER, R. C. **Structural Analysis**. 9. ed. Upper Saddle River: Pearson Education, 2015. ISBN 10: 0-13-394284-8.
- HUTTON, D. V. **Fundamentals of Finite Element Analysis**. 1. ed. New York: McGraw Hill, 2004. ISBN 0-07-239536-2.
- KWON, Y. W. **The Finite Element Method: Using MATLAB**. 2. ed. Boca Raton: CRC Press, 2000. ISBN 0-8493-0096-7.
- LOGAN, D. L. **A First Course in the Finite Element Method**. 4. ed. Toronto: Nelson, 2007. ISBN 0-534-55298-6.
- MATSUMOTO, É. Y. **MATLAB R2013a - Teoria e Programação: Guia Prático**. 1. ed. São Paulo: Érica, 2013. ISBN 978-85-365-1940-1.
- NEWLAND, D. E. Vibration of the London Millennium Bridge: Cause and Cure. **International Journal of Acoustics and Vibration**, v. 8, n. 1, p. 9-14, 2003.
- SORIANO, H. L. **Introdução à Dinâmica das Estruturas**. 1. ed. Rio de Janeiro: Elsevier, 2014. ISBN 978-85-352-5153-1.
- SORIANO, H. L. **Elementos Finitos: Formulação e Aplicação na Estática e Dinâmica das Estruturas**. 1. ed. Rio de Janeiro: Ciência Moderna, 2009. ISBN 978-85-739-3880-7.
- THOMPSON, M. K. **ANSYS Mechanical APDL for Finite Element Analysis**. 1. ed. Oxford: Butterworth-Heinemann, 2017. ISBN 978-0-12-812981-4.

APÊNDICE A - Código fonte do programa desenvolvido

```

classdef ModalAnalysis_App10_exported < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public)
        UIFigure          matlab.ui.Figure
        Oquecalcular      matlab.ui.control.DropDown
        AspropriedadesabaixosoconstantesparatodasbarrasLabel_2
    matlab.ui.control.Label
        Procurar3         matlab.ui.control.Button
        File3             matlab.ui.control.EditField
        Procurar2         matlab.ui.control.Button
        File2             matlab.ui.control.EditField
        Procurar1         matlab.ui.control.Button
        File1             matlab.ui.control.EditField
        Tabela3_texto     matlab.ui.control.Label
        Tabela2_texto     matlab.ui.control.Label
        Tabela1_texto     matlab.ui.control.Label
        Insercaodedados  matlab.ui.control.DropDown
        DropDownLabel     matlab.ui.control.Label
        Propriedadesconstantes  matlab.ui.control.DropDown
        AspropriedadesabaixosoconstantesparatodasbarrasLabel
    matlab.ui.control.Label
        Quantidadedebarras  matlab.ui.control.NumericEditField
        QuantidadedebarrasEditFieldLabel  matlab.ui.control.Label
        Quantidadedenos     matlab.ui.control.NumericEditField
        QuantidadedensEditFieldLabel  matlab.ui.control.Label
        Valoresrestringidos  matlab.ui.control.NumericEditField
        ValoresrestringidosvetorLabel  matlab.ui.control.Label
        Limitesup           matlab.ui.control.NumericEditField
        LimitesupEditFieldLabel  matlab.ui.control.Label
        Limiteinf           matlab.ui.control.NumericEditField
        LimiteinfEditFieldLabel  matlab.ui.control.Label
        Tabelarestricoes    matlab.ui.control.Table
        Tabela1             matlab.ui.control.Table
        Atualizar           matlab.ui.control.Button
        Calcular            matlab.ui.control.Button
        Titulo              matlab.ui.control.Label
        Tabela3             matlab.ui.control.Table
        Tabela2             matlab.ui.control.Table
        Modosdevibraoaseremexibidos  matlab.ui.control.Label
        Estrutura           matlab.ui.control.UIAxes
    end

```

```

% Callbacks that handle component events
methods (Access = private)

% Value changed function: Quantidadedebarras
function QuantidadedebarrasValueChanged(app, event)
    % Quantidade de barras
    qb = app.Quantidadedebarras.Value;

    % Restrição para carregar as tabelas
    insercaodedados = app.Insercaodedados.Value;
    switch insercaodedados
        case 'Manualmente'
            barra_no = zeros(qb,3);
            for i = 1:qb
                barra_no(i,1) = i;
            end
            app.Tabela1.Data = barra_no;
        end
    end

% Value changed function: Quantidadedenos
function QuantidadedenosValueChanged(app, event)
    % Quantidade de nós
    qn = app.Quantidadedenos.Value;

    % Restrição para carregar as tabelas
    insercaodedados = app.Insercaodedados.Value;
    switch insercaodedados
        case 'Manualmente'
            no_coord = zeros(qn,3);
            for i = 1:qn
                no_coord(i,1) = i;
            end
            app.Tabela2.Data = no_coord;
        end
    end

% Button pushed function: Atualizar
function AtualizarButtonPushed(app, event)
    % Barras e nós
    no_coord = app.Tabela2.Data;
    barra_no = app.Tabela1.Data;
    qb = size(barra_no,1);

```

```

% Plotar
cla(app.Estrutura);
hold(app.Estrutura,'on');
for i = 1:qb
    no_inicial = barra_no(i,2);
    no_final = barra_no(i,3);
    X = [no_coord(no_inicial,2) no_coord(no_final,2)];
    Y = [no_coord(no_inicial,3) no_coord(no_final,3)];
    plot(app.Estrutura,X,Y,'k-', 'LineWidth', 1);
end
end

% Value changed function: Valoresrestringidos
function ValoresrestringidosValueChanged(app, event)
    % Restrições de deslocamento
    restricoes = app.Valoresrestringidos.Value;
    U_zero = zeros(restricoes,1);

    % Tabela de restrições
    app.Tabelarestricoes.Data = U_zero; % Lembrar de depois de transpor
end

% Value changed function: Propriedadesconstantes
function PropriedadesconstantesValueChanged(app, event)
    % Propriedade das barras
    propriedade_b = app.Propriedadesconstantes.Value;
    switch propriedade_b
        case 'Sim'
            app.Tabela3.Enable = 'on';
            propriedade_matriz = zeros(1,4);
            propriedade_matriz(1,1) = 1;
        case 'Não'
            app.Tabela3.Enable = 'on';
            propriedade_matriz = zeros(app.Quantidadedebarras.Value,4);
            for i = 1:app.Quantidadedebarras.Value
                propriedade_matriz(i,1) = i;
            end
        case 'Escolha'
            app.Tabela3.Enable = 'off';
    end
    app.Tabela3.Data = propriedade_matriz;
end

% Button pushed function: Calcular

```

```

function CalcularButtonPushed(app, event)
    % Limpar e quantidade de casas decimais
    format long;
    clc;
    close all

    % Deslocamentos restringidos
    for i = 1:app.Valoresrestringidos.Value
        U_zero(1,i) = app.Tabelarestricoes.Data(i,1);
    end
    U_zero = sort(U_zero);

    % Intervalo dos modos de vibração a serem exibidos
    primeiro_mv = app.Limiteinf.Value;
    ultimo_mv = app.Limitesup.Value;

    % Barras e nós
    no_coord = app.Tabela2.Data;
    barra_no = app.Tabela1.Data;

    % Nós com restrições e sem restrições
    qn = size(app.Tabela2.Data,1);
    gdl = qn*2;
    U_ref = ones(1,gdl);
    for i = 1:size(U_zero,2)
        U_ref(1,U_zero(1,i)) = 0;
    end

    % Restrição para carregar as tabelas
    insercaodedados = app.Insercaodedados.Value;
    qb = size(app.Tabela1.Data,1);
    switch insercaodedados
        case 'Manualmente'
            matriz_propriedades = app.Tabela3.Data;
            A = zeros(1,qb);
            E = zeros(1,qb);
            rho = zeros(1,qb);
            switch app.Propriedadesconstantes.Value
                case 'Sim'
                    A = matriz_propriedades(1,2)*ones(1,qb)*10^-4;
                    E = matriz_propriedades(1,3)*ones(1,qb)*10^9;
                    rho = matriz_propriedades(1,4)*ones(1,qb);
                case 'Não'
                    for i = 1:qb

```

```

        A(1,i) = matriz_propriedades(i,2)*10^-4;
        E(1,i) = matriz_propriedades(i,3)*10^9;
        rho(1,i) = matriz_propriedades(i,4);
    end
end
case 'Carregar arquivo'
    A(1,:) = app.Tabela3.Data(:,2)*10^-4;
    E(1,:) = app.Tabela3.Data(:,3)*10^9;
    rho(1,:) = app.Tabela3.Data(:,4);
end

% Matrizes de massa e rigidez nulas
K = zeros(gdl);
M = zeros(gdl);
K_aux = zeros(gdl);
M_aux = zeros(gdl);

% Comprimento das barras, senos, cossenos e matrizes de rigidez e de
massa globais do sistema
for i = 1:qb
    no_inicial = barra_no(i,2);
    no_final = barra_no(i,3);
    x1 = no_coord(no_inicial,2);
    x2 = no_coord(no_final,2);
    y1 = no_coord(no_inicial,3);
    y2 = no_coord(no_final,3);
    L(i) = sqrt((x2 - x1)^2 + (y2 - y1)^2);
    C = (x2 - x1)/L(i);
    S = (y2 - y1)/L(i);
    K_ele = (A(i)*E(i)/L(i)*[C^2 C*S -C^2 -C*S; C*S S^2 -C*S -S^2; -C^2 -C*S
C^2 C*S; -C*S -S^2 C*S S^2]); % Matriz de rigidez elementar em coordenadas
globais
    M_ele = (rho(i)*A(i)*L(i)/6*[2 0 1 0; 0 2 0 1; 1 0 2 0; 0 1 0 2]); % Matriz de
massa elementar em coordenadas globais
    % Matrizes auxiliares
    K_aux((no_inicial*2)-1:no_inicial*2,(no_inicial*2)-1:no_inicial*2) =
K_ele(1:2,1:2);
    K_aux((no_inicial*2)-1:no_inicial*2,(no_final*2)-1:no_final*2) = K_ele(1:2,3:4);
    K_aux((no_final*2)-1:no_final*2,(no_inicial*2)-1:no_inicial*2) = K_ele(3:4,1:2);
    K_aux((no_final*2)-1:no_final*2,(no_final*2)-1:no_final*2) = K_ele(3:4,3:4);
    M_aux((no_inicial*2)-1:no_inicial*2,(no_inicial*2)-1:no_inicial*2) =
M_ele(1:2,1:2);
    M_aux((no_inicial*2)-1:no_inicial*2,(no_final*2)-1:no_final*2) =
M_ele(1:2,3:4);

```

```

M_aux((no_final*2)-1:no_final*2,(no_inicial*2)-1:no_inicial*2) =
M_ele(3:4,1:2);
M_aux((no_final*2)-1:no_final*2,(no_final*2)-1:no_final*2) = M_ele(3:4,3:4);
% Matrizes globais do sistema
K = K + K_aux;
M = M + M_aux;
% Zerar as matrizes auxiliares
K_aux = zeros(gdl);
M_aux = zeros(gdl);
end

% Zerar linhas e colunas devido às restrições
contador = 1;
while (contador <= size(U_zero,2))
    for i = 1:size(U_zero,2)
        K(U_zero(i)+(1-contador),:) = [];
        K(:,U_zero(i)+(1-contador)) = [];
        M(U_zero(i)+(1-contador),:) = [];
        M(:,U_zero(i)+(1-contador)) = [];
        contador = contador + 1;
    end
end

% Autovalor e autovetor
D = inv(M)*K;
[autovetores,autovalores] = eig(D); % Calcula os autovalores e autovetores
autovalores_vetor = diag(autovalores);
autovalores_ordenados = sort(autovalores_vetor); % Ordena os autovalores
em ordem crescente
% Ordenar os autovetores de acordo com os autovalores
valores_testados = ones(length(autovalores),1);
for i = 1:length(autovalores)
    for j = 1:length(autovalores)
        if autovalores_vetor(j) == autovalores_ordenados(i)
            if valores_testados(j,1) == 1
                phi(:,i) = autovetores(:,j);
                valores_testados(j,1) = 0;
                break
            else
                continue
            end
        end
    end
end
end
end
end
end

```

```

omega = sqrt(autovalores_ordenados); % Frequências naturais em ordem
crescente

% Normalizando os modos de vibração
for j = 1:length(phi)
    for i = 1:length(phi)
        if phi(i,j) ~= 0
            constante = 1/phi(i,j);
            phi(:,j) = constante*phi(:,j);
            break
        else
            continue
        end
    end
end

% Plotar modos de vibração
switch app.Oquecalcular.Value
    case 'Modos de Vibração'
        max_mv = max(abs(phi)); % Utilizado para deixar a "plotagem" dos
diferentes modos de vibração proporcionais entre si
        for m = primeiro_mv:ultimo_mv
            figure;
            axis equal;
            hold ON; % Permite que multiplos plots sejam desenhados no mesmo
eixo

            xlabel('metros');
            ylabel('metros');
            for i = 1:qb
                no_inicial = barra_no(i,2);
                no_final = barra_no(i,3);
                X = [no_coord(no_inicial,2) no_coord(no_final,2)];
                Y = [no_coord(no_inicial,3) no_coord(no_final,3)];
                if U_ref(1,2*(no_inicial)-1) == 1
                    contador_1 = 0;
                    for n = 1:2*no_inicial-2 % "2*no_inicial-1" já é igual a 1
                        if U_ref(1,n) == 0
                            contador_1 = contador_1 + 1; % Conta a quantidade linhas
e colunas que foram eliminadas até o nó analisado
                        else
                            continue
                        end
                    end
                end
            end
        end
    end
end

```



```

X_mv(1,1) = no_coord(no_inicial,2) + mean(L)*phi(2*no_inicial -
1 - contador_1,m)/(max_mv(1,m)/0.2); % X1 do modo de vibração diferente do valor
inicial
else
X_mv(1,1) = X(1,1); % Caso haja restrição na direção analisada,
o x1 não irá se alterar
end
if U_ref(1,2*(no_final)-1) == 1
contador_2 = 0;
for n = 1:2*no_final-2 % "2*no_final-1" já é igual a 1
if U_ref(1,n) == 0
contador_2 = contador_2 + 1; % Conta a quantidade linhas
e colunas que foram eliminadas até o nó analisado
else
continue
end
end
X_mv(1,2) = no_coord(no_final,2) + mean(L)*phi(2*no_final - 1 -
contador_2,m)/(max_mv(1,m)/0.2); % X2 do modo de vibração diferente do valor
inicial
else
X_mv(1,2) = X(1,2); % Caso haja restrição na direção analisada,
o x2 não irá se alterar
end
if U_ref(1,2*no_inicial) == 1
contador_3 = 0;
for n = 1:2*no_inicial-1 % "2*no_inicial" já é igual a 1
if U_ref(1,n) == 0
contador_3 = contador_3 + 1; % Conta a quantidade linhas
e colunas que foram eliminadas até o nó analisado
else
continue
end
end
Y_mv(1,1) = no_coord(no_inicial,3) + mean(L)*phi(2*no_inicial -
contador_3,m)/(max_mv(1,m)/0.2); % Y1 do modo de vibração diferente do valor
inicial
else
Y_mv(1,1) = Y(1,1); % Caso haja restrição na direção analisada,
o y1 não irá se alterar
end
if U_ref(1,2*no_final) == 1
contador_4 = 0;
for n = 1:2*no_final-1 % "2*no_final" já é igual a 1

```

```

        if U_ref(1,n) == 0
            contador_4 = contador_4 + 1;
        else
            continue
        end
    end
    Y_mv(1,2) = no_coord(no_final,3) + mean(L)*phi(2*no_final -
contador_4,m)/(max_mv(1,m)/0.2); % Y2 do modo de vibração diferente do valor
inicial

        else
            Y_mv(1,2) = Y(1,2); % Caso haja restrição na direção analisada,
o y1 não irá se alterar
        end
        plot(X,Y, 'k--', 'LineWidth', 1.5);
        plot(X_mv,Y_mv, 'b-', 'LineWidth', 2);
        set(gcf, 'WindowState', 'maximized');
    end
    titulo = [num2str(m), 'º Modo de vibração'];
    subtitulo = ['Frequência natural: ', num2str(omega(m,1)/(pi*2)), ' Hz'];
    title(titulo, 'FontSize', 20);
    subtitle(subtitulo, 'FontSize', 16);
    legenda = legend({'Estrutura não deformada', 'Modo de vibração
considerado'}, 'Location', 'northeastoutside', 'NumColumns', 1, 'FontSize', 15);
    title(legenda, 'Legenda', 'FontSize', 17);
end
case 'Frequências Naturais'
    % Tabela com as frequências naturais
    omega_hz = omega/(pi*2);
    bar(omega_hz);
    xlabel('Modos de Vibrar', 'fontsize', 11);
    ylabel('Frequências em Hz', 'fontsize', 11);
    title('Frequências Naturais', 'fontsize', 14);
end

% Exibir frequências naturais na Command Window
clc
disp('Frequências naturais (em rad/s): ')
disp(omega)
disp('Frequências naturais (em Hz): ')
disp(omega/(pi*2))

% Exibir os modos de vibração na Command Window
% Linhas
contador = 1;

```

```

for i = 1:gdl
    if U_ref(1,i) == 1
        linhas = num2str(i);
        texto = 'U';
        num_linhas(contador,1) = join({texto,linhas});
        contador = contador + 1;
    else
        continue
    end
end
% Colunas
for i = 1:size(phi,1)
    colunas = num2str(i);
    texto = 'Modo';
    num_colunas(i,1) = join({texto,colunas});
end
% Criando a tabela
tabela =
array2table(phi,"VariableNames",num_colunas,"RowNames",num_linhas);
disp('Modos de vibração da estrutura: ')
disp(tabela)
end

% Button pushed function: Procurar1
function Procurar1ButtonPushed(app, event)
    [file,Fpath]=uigetfile('/*.xlsx');
    app.File1.Value=[Fpath file];
    ds_tabela_1 = datastore(app.File1.Value);
    tabela_1 = readall(ds_tabela_1);
    tabela_1 = table2array(tabela_1)
    app.Tabela1.Data = tabela_1;
end

% Button pushed function: Procurar2
function Procurar2ButtonPushed(app, event)
    [file,Fpath]=uigetfile('/*.xlsx');
    app.File2.Value=[Fpath file];
    ds_tabela_2 = datastore(app.File2.Value);
    tabela_2 = readall(ds_tabela_2);
    tabela_2 = table2array(tabela_2)
    app.Tabela2.Data = tabela_2;
end

% Button pushed function: Procurar3

```

```

function Procurar3ButtonPushed(app, event)
    [file,Fpath]=uigetfile('./*.xlsx');
    app.File3.Value=[Fpath file];
    ds_tabela_3 = datastore(app.File3.Value);
    tabela_3 = readall(ds_tabela_3);
    tabela_3 = table2array(tabela_3)
    app.Tabela3.Data = tabela_3;
end

% Value changed function: Insercaodedados
function InsercaodedadosValueChanged(app, event)
    insercaodedados = app.Insercaodedados.Value;
    switch insercaodedados
        case 'Manualmente'
            app.Procurar1.Enable = 'off';
            app.Procurar2.Enable = 'off';
            app.Procurar3.Enable = 'off';
            app.Quantidadedebarras.Enable = 'on';
            app.Quantidadedenos.Enable = 'on';
            app.Propriedadesconstantes.Enable = 'on';
            app.Tabelarestricoes.Enable = 'on'
            app.Atualizar.Enable = 'on'
            app.Tabela1.Enable = 'on';
            app.Tabela2.Enable = 'on';
            app.Tabela3.Enable = 'on';
            app.Valoresrestringidos.Enable = 'on'
            app.Oquecalcular.Enable = 'on'
        case 'Carregar arquivo'
            app.Procurar1.Enable = 'on';
            app.Procurar2.Enable = 'on';
            app.Procurar3.Enable = 'on';
            app.Quantidadedebarras.Enable = 'off';
            app.Quantidadedenos.Enable = 'off';
            app.Tabela1.Enable = 'on';
            app.Tabela2.Enable = 'on';
            app.Tabela3.Enable = 'on';
            app.Valoresrestringidos.Enable = 'on'
            app.Tabelarestricoes.Enable = 'on'
            app.Atualizar.Enable = 'on'
            app.Tabela1.Data = [];
            app.Tabela2.Data = [];
            app.Tabela3.Data = [];
            app.Propriedadesconstantes.Enable = 'off';
            app.Quantidadedebarras.Value = 0;
    end
end

```

```

    app.Quantidadedenos.Value = 0;
    app.Oquecalcular.Enable = 'on'
case 'Escolha'
    app.Procurar1.Enable = 'on';
    app.Procurar2.Enable = 'on';
    app.Procurar3.Enable = 'on';
    app.Quantidadedebarras.Enable = 'off';
    app.Quantidadedenos.Enable = 'off';
    app.Tabela1.Enable = 'off';
    app.Tabela2.Enable = 'off';
    app.Tabela3.Enable = 'off';
    app.Valoresrestringidos.Enable = 'off'
    app.Tabelarestricoes.Enable = 'off'
    app.Atualizar.Enable = 'off'
    app.Tabela1.Data = [];
    app.Tabela1.Data = [];
    app.Tabela2.Data = [];
    app.Tabela3.Data = [];
    app.Propriedadesconstantes.Enable = 'off'
    app.Oquecalcular.Enable = 'off'
end
end

% Value changed function: Oquecalcular
function OquecalcularValueChanged(app, event)
    switch app.Oquecalcular.Value
        case 'Frequências Naturais'
            app.Limiteinf.Enable = 'off';
            app.Limitesup.Enable = 'off';
            app.Calcular.Enable = 'on';
        case 'Modos de Vibração'
            app.Limiteinf.Enable = 'on';
            app.Limitesup.Enable = 'on';
            app.Calcular.Enable = 'on';
        case 'Escolha'
            app.Limiteinf.Enable = 'off';
            app.Limitesup.Enable = 'off';
            app.Calcular.Enable = 'off';
        end
    end
end

% Component initialization
methods (Access = private)

```

```
% Create UIFigure and components
```

```
function createComponents(app)
```

```
% Create UIFigure and hide until all components are created
```

```
app.UIFigure = uifigure('Visible', 'off');
app.UIFigure.Position = [100 100 845 686];
app.UIFigure.Name = 'MATLAB App';
```

```
% Create Estrutura
```

```
app.Estrutura = uiaxes(app.UIFigure);
title(app.Estrutura, 'Estrutura')
xlabel(app.Estrutura, 'metros')
ylabel(app.Estrutura, 'metros')
zlabel(app.Estrutura, 'Z')
app.Estrutura.Position = [546 278 294 276];
```

```
% Create Modosdevibraoaseremexibidos
```

```
app.Modosdevibraoaseremexibidos = uilabel(app.UIFigure);
app.Modosdevibraoaseremexibidos.BackgroundColor = [0.6 0.7294 0.9686];
app.Modosdevibraoaseremexibidos.HorizontalAlignment = 'center';
app.Modosdevibraoaseremexibidos.Position = [636 109 202 22];
app.Modosdevibraoaseremexibidos.Text = 'Modos de vibração a serem
exibidos';
```

```
% Create Tabela2
```

```
app.Tabela2 = uitable(app.UIFigure);
app.Tabela2.BackgroundColor = [1 1 1];
app.Tabela2.ColumnName = {'Nós'; 'x (m)'; 'y (m)'};
app.Tabela2.RowName = {};
app.Tabela2.ColumnEditable = [false true true];
app.Tabela2.Enable = 'off';
app.Tabela2.FontSize = 10;
app.Tabela2.Position = [298 401 225 80];
```

```
% Create Tabela3
```

```
app.Tabela3 = uitable(app.UIFigure);
app.Tabela3.ColumnName = {'Barras'; 'Área (cm²)'; 'Módulo de elasticidade
(GPa)'; 'Massa específica (kg/m³)'};
app.Tabela3.RowName = {};
app.Tabela3.ColumnEditable = [false true true true];
app.Tabela3.Enable = 'off';
app.Tabela3.FontSize = 10;
app.Tabela3.Position = [28 44 494 87];
```

```

% Create Titulo
app.Titulo = uilabel(app.UIFigure);
app.Titulo.BackgroundColor = [0.4118 0.5176 0.7216];
app.Titulo.HorizontalAlignment = 'center';
app.Titulo.FontName = 'Microsoft JhengHei';
app.Titulo.FontSize = 20;
app.Titulo.FontWeight = 'bold';
app.Titulo.Position = [81 611 400 61];
app.Titulo.Text = {'ANÁLISE MODAL DE TRELIÇAS PLANAS'; 'VIA
MÉTODO DOS ELEMENTOS FINITOS'};

```

```

% Create Calcular
app.Calcular = uibutton(app.UIFigure, 'push');
app.Calcular.ButtonPushedFcn = createCallbackFcn(app,
@CalcularButtonPushed, true);
app.Calcular.BackgroundColor = [0.9176 0.949 0];
app.Calcular.Enable = 'off';
app.Calcular.Position = [636 23 203 23];
app.Calcular.Text = 'Calcular!';

```

```

% Create Atualizar
app.Atualizar = uibutton(app.UIFigure, 'push');
app.Atualizar.ButtonPushedFcn = createCallbackFcn(app,
@AtualizarButtonPushed, true);
app.Atualizar.Enable = 'off';
app.Atualizar.Tooltip = {'Pré-visualizar a estrutura...'};
app.Atualizar.Position = [660 238 100 23];
app.Atualizar.Text = 'Atualizar';

```

```

% Create Tabela1
app.Tabela1 = uitable(app.UIFigure);
app.Tabela1.BackgroundColor = [1 1 1];
app.Tabela1.ColumnName = {'Barras'; 'Nó inicial'; 'Nó final'};
app.Tabela1.RowName = {};
app.Tabela1.ColumnEditable = [false true true];
app.Tabela1.Tooltip = {''};
app.Tabela1.Enable = 'off';
app.Tabela1.FontSize = 10;
app.Tabela1.Position = [38 402 243 80];

```

```

% Create Tabelarestricoes
app.Tabelarestricoes = uitable(app.UIFigure);
app.Tabelarestricoes.BackgroundColor = [1 1 1];

```

```

app.Tabelarestricoes.ColumnName = {'GDL Restringidos'};
app.Tabelarestricoes.RowName = {};
app.Tabelarestricoes.ColumnEditable = true;
app.Tabelarestricoes.Enable = 'off';
app.Tabelarestricoes.Position = [387 239 136 85];

% Create LimiteinfEditFieldLabel
app.LimiteinfEditFieldLabel = uilabel(app.UIFigure);
app.LimiteinfEditFieldLabel.FontSize = 10;
app.LimiteinfEditFieldLabel.Position = [704 79 54 22];
app.LimiteinfEditFieldLabel.Text = 'Limite inf.';

% Create Limiteinf
app.Limiteinf = uieditfield(app.UIFigure, 'numeric');
app.Limiteinf.FontSize = 10;
app.Limiteinf.Enable = 'off';
app.Limiteinf.Position = [769 79 69 22];

% Create LimitesupEditFieldLabel
app.LimitesupEditFieldLabel = uilabel(app.UIFigure);
app.LimitesupEditFieldLabel.FontSize = 10;
app.LimitesupEditFieldLabel.Position = [704 58 54 22];
app.LimitesupEditFieldLabel.Text = 'Limite sup.';

% Create Limitesup
app.Limitesup = uieditfield(app.UIFigure, 'numeric');
app.Limitesup.FontSize = 10;
app.Limitesup.Enable = 'off';
app.Limitesup.Position = [769 58 69 22];

% Create ValoresrestringidosvetorLabel
app.ValoresrestringidosvetorLabel = uilabel(app.UIFigure);
app.ValoresrestringidosvetorLabel.BackgroundColor = [0.6 0.7294 0.9686];
app.ValoresrestringidosvetorLabel.HorizontalAlignment = 'center';
app.ValoresrestringidosvetorLabel.Position = [39 302 165 22];
app.ValoresrestringidosvetorLabel.Text = 'Restrições ao deslocamento';

% Create Valoresrestringidos
app.Valoresrestringidos = uieditfield(app.UIFigure, 'numeric');
app.Valoresrestringidos.Limits = [0 Inf];
app.Valoresrestringidos.ValueChangedFcn = createCallbackFcn(app,
@ValoresrestringidosValueChanged, true);
app.Valoresrestringidos.Enable = 'off';
app.Valoresrestringidos.Position = [211 302 130 22];

```



```

% Create QuantidadeensEditFieldLabel
app.QuantidadeensEditFieldLabel = uilabel(app.UIFigure);
app.QuantidadeensEditFieldLabel.HorizontalAlignment = 'right';
app.QuantidadeensEditFieldLabel.FontSize = 10;
app.QuantidadeensEditFieldLabel.Position = [302 503 90 22];
app.QuantidadeensEditFieldLabel.Text = 'Quantidade de nós';

% Create Quantidadeenos
app.Quantidadeenos = uieditfield(app.UIFigure, 'numeric');
app.Quantidadeenos.ValueDisplayFormat = '%.0f';
app.Quantidadeenos.ValueChangedFcn = createCallbackFcn(app,
@QuantidadeenosValueChanged, true);
app.Quantidadeenos.FontSize = 10;
app.Quantidadeenos.Enable = 'off';
app.Quantidadeenos.Position = [407 503 29 22];

% Create QuantidadeebarrasEditFieldLabel
app.QuantidadeebarrasEditFieldLabel = uilabel(app.UIFigure);
app.QuantidadeebarrasEditFieldLabel.HorizontalAlignment = 'right';
app.QuantidadeebarrasEditFieldLabel.FontSize = 10;
app.QuantidadeebarrasEditFieldLabel.Position = [38 503 102 22];
app.QuantidadeebarrasEditFieldLabel.Text = 'Quantidade de barras';

% Create Quantidadeebarras
app.Quantidadeebarras = uieditfield(app.UIFigure, 'numeric');
app.Quantidadeebarras.ValueDisplayFormat = '%.0f';
app.Quantidadeebarras.ValueChangedFcn = createCallbackFcn(app,
@QuantidadeebarrasValueChanged, true);
app.Quantidadeebarras.FontSize = 10;
app.Quantidadeebarras.Enable = 'off';
app.Quantidadeebarras.Position = [155 503 29 22];

% Create AspropriedadesabaixosoconstantesparatodasasbarrasLabel
app.AspropriedadesabaixosoconstantesparatodasasbarrasLabel =
uilabel(app.UIFigure);

app.AspropriedadesabaixosoconstantesparatodasasbarrasLabel.HorizontalAlignmen
t = 'right';
app.AspropriedadesabaixosoconstantesparatodasasbarrasLabel.Position =
[20 152 340 22];
app.AspropriedadesabaixosoconstantesparatodasasbarrasLabel.Text = 'As
propriedades abaixo são constantes para todas as barras?';

```

```

% Create Propriedadesconstantes
app.Propriedadesconstantes = uidropdown(app.UIFigure);
app.Propriedadesconstantes.Items = {'Sim', 'Não', 'Escolha'};
app.Propriedadesconstantes.ValueChangedFcn = createCallbackFcn(app,
@PropriedadesconstantesValueChanged, true);
app.Propriedadesconstantes.Enable = 'off';
app.Propriedadesconstantes.Position = [375 152 89 22];
app.Propriedadesconstantes.Value = 'Escolha';

% Create DropDownLabel
app.DropDownLabel = uilabel(app.UIFigure);
app.DropDownLabel.BackgroundColor = [0.7569 0.8863 0.9686];
app.DropDownLabel.FontSize = 14;
app.DropDownLabel.FontWeight = 'bold';
app.DropDownLabel.Position = [39 542 482 43];
app.DropDownLabel.Text = {'Primeiramente, como gostaria de inserir as '
'informações nas Tabelas 1, 2 e 3?'};

% Create Insercaodedados
app.Insercaodedados = uidropdown(app.UIFigure);
app.Insercaodedados.Items = {'Escolha', 'Manualmente', 'Carregar arquivo'};
app.Insercaodedados.ValueChangedFcn = createCallbackFcn(app,
@InsercaodedadosValueChanged, true);
app.Insercaodedados.Position = [340 553 132 22];
app.Insercaodedados.Value = 'Escolha';

% Create Tabela1_texto
app.Tabela1_texto = uilabel(app.UIFigure);
app.Tabela1_texto.FontColor = [1 0 0];
app.Tabela1_texto.Position = [38 481 51 22];
app.Tabela1_texto.Text = 'Tabela 1';

% Create Tabela2_texto
app.Tabela2_texto = uilabel(app.UIFigure);
app.Tabela2_texto.FontColor = [1 0 0];
app.Tabela2_texto.Position = [300 480 51 22];
app.Tabela2_texto.Text = 'Tabela 2';

% Create Tabela3_texto
app.Tabela3_texto = uilabel(app.UIFigure);
app.Tabela3_texto.FontColor = [1 0 0];
app.Tabela3_texto.Position = [29 131 51 22];
app.Tabela3_texto.Text = 'Tabela 3';

```

```

% Create File1
app.File1 = uieditfield(app.UIFigure, 'text');
app.File1.Editable = 'off';
app.File1.Position = [39 369 135 22];

% Create Procurar1
app.Procurar1 = uibutton(app.UIFigure, 'push');
app.Procurar1.ButtonPushedFcn = createCallbackFcn(app,
@Procurar1ButtonPushed, true);
app.Procurar1.Enable = 'off';
app.Procurar1.Tooltip = {'Inserir dados no formato "xlsx"...'};
app.Procurar1.Position = [183 368 100 23];
app.Procurar1.Text = 'Procurar';

% Create File2
app.File2 = uieditfield(app.UIFigure, 'text');
app.File2.Editable = 'off';
app.File2.Position = [299 368 117 22];

% Create Procurar2
app.Procurar2 = uibutton(app.UIFigure, 'push');
app.Procurar2.ButtonPushedFcn = createCallbackFcn(app,
@Procurar2ButtonPushed, true);
app.Procurar2.Enable = 'off';
app.Procurar2.Tooltip = {'Inserir dados no formato "xlsx"...'};
app.Procurar2.Position = [424 367 100 23];
app.Procurar2.Text = 'Procurar';

% Create File3
app.File3 = uieditfield(app.UIFigure, 'text');
app.File3.Editable = 'off';
app.File3.Position = [29 13 386 22];

% Create Procurar3
app.Procurar3 = uibutton(app.UIFigure, 'push');
app.Procurar3.ButtonPushedFcn = createCallbackFcn(app,
@Procurar3ButtonPushed, true);
app.Procurar3.Enable = 'off';
app.Procurar3.Tooltip = {'Inserir dados no formato "xlsx"...'};
app.Procurar3.Position = [423 13 100 23];
app.Procurar3.Text = 'Procurar';

% Create AspropriedadesabaixoconstanteparatodasasbarrasLabel_2

```

```

        app.AspropriedadesabaixosoconstantesparatodasasbarrasLabel_2 =
        uilabel(app.UIFigure);

app.AspropriedadesabaixosoconstantesparatodasasbarrasLabel_2.BackgroundColor
= [0.7608 0.8902 0.9686];

app.AspropriedadesabaixosoconstantesparatodasasbarrasLabel_2.HorizontalAlignm
ent = 'center';
        app.AspropriedadesabaixosoconstantesparatodasasbarrasLabel_2.FontSize
= 14;

app.AspropriedadesabaixosoconstantesparatodasasbarrasLabel_2.FontWeight =
'bold';
        app.AspropriedadesabaixosoconstantesparatodasasbarrasLabel_2.Position =
[568 167 106 34];
        app.AspropriedadesabaixosoconstantesparatodasasbarrasLabel_2.Text = {'O
que deseja'; 'calcular?'};

        % Create Oquecalcular
        app.Oquecalcular = uidropdown(app.UIFigure);
        app.Oquecalcular.Items = {'Frequências Naturais', 'Modos de Vibração',
'Escolha'};
        app.Oquecalcular.ValueChangedFcn = createCallbackFcn(app,
@OquecalcularValueChanged, true);
        app.Oquecalcular.Enable = 'off';
        app.Oquecalcular.Position = [687 179 152 22];
        app.Oquecalcular.Value = 'Escolha';

        % Show the figure after all components are created
        app.UIFigure.Visible = 'on';
    end
end

% App creation and deletion
methods (Access = public)

    % Construct app
    function app = ModalAnalysis_App10_exported

        % Create UIFigure and components
        createComponents(app)

        % Register the app with App Designer
        registerApp(app, app.UIFigure)

```

```
    if nargin == 0
        clear app
    end
end

% Code that executes before app deletion
function delete(app)

    % Delete UIFigure when app is deleted
    delete(app.UIFigure)
end
end
end
```