UNIVERSIDADE FEDERAL DE JUIZ DE FORA

VALIDAÇÃO DOS CÁLCULOS E RELATÓRIOS

MAC015 - Resistência dos Materias

Júlia Zoffoli Caçador 202365520B

Robert Gonçalves Vieira de Souza 202365505B

Rubia Danielle Viol 202365515B

Link para o Google Colab: Link para o Google Colab

Juiz de Fora, 2025

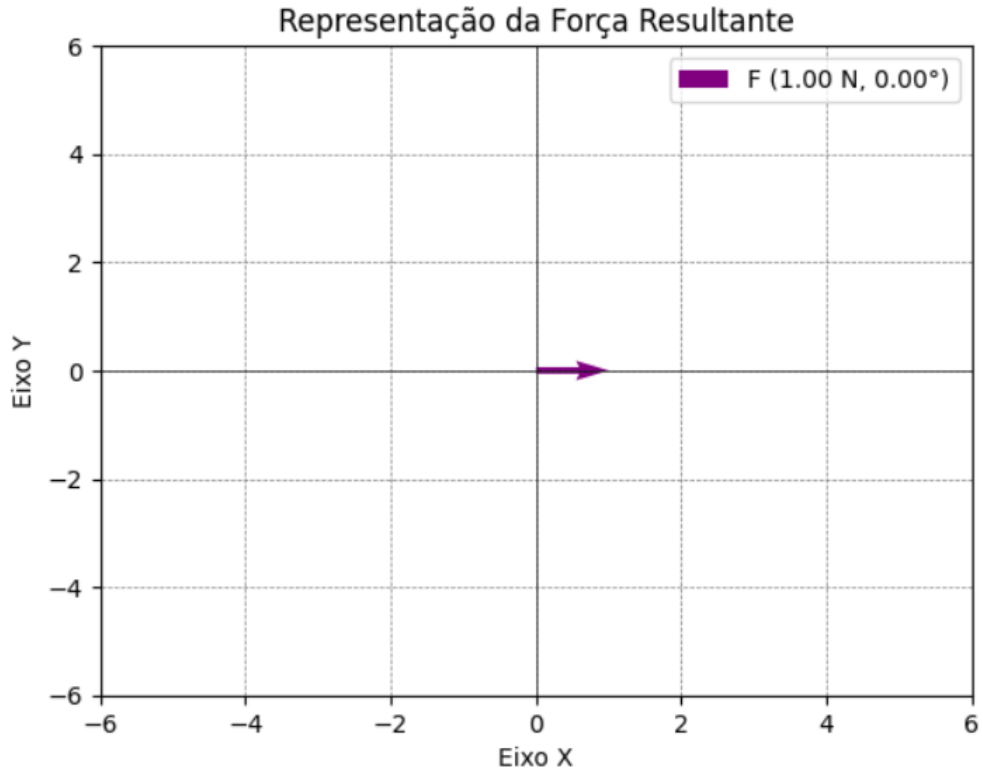
Sumário

1 Observações

Para a validação da rotina computacional gerada, foi usado o software *FTool* para comparar os resultados.

2 Questão 1

2.1 Gráfico




```
Força resultante: 1.00 N
Direção da resultante: 0.00 graus
```

Figura 1: Gráfico para representação da força resultante

2.2 Relatório

Para a execução da 1ª questão, foi desenvolvida uma rotina computacional que tem como propósito calcular e representar graficamente a força resultante de um conjunto de forças coplanares concorrentes.

Ele solicita ao usuário o número de forças atuantes e permite escolher entre dois formatos de entrada:

1. Inserção de intensidade e ângulo de cada força em relação à horizontal

2. Inserção das componentes cartesianas x e y das forças.

Para o primeiro formato, ele utiliza a função *separaComponentes* para calcular as componentes x (horizontal) e y (vertical) de cada força a partir de sua intensidade e ângulo, aplicando as funções trigonométricas seno e cosseno. No segundo formato, os valores das componentes são fornecidos diretamente.

Após armazenar todas as forças em uma matriz de coordenadas, o programa utiliza a função *calculaResultante* para determinar a força resultante. Ele soma separadamente os componentes x e y de todas as forças para obter os componentes da resultante. Com esses valores, calcula a magnitude da resultante utilizando o teorema de Pitágoras

$$F = \sqrt{x^2 + y^2}$$

e determina a direção do vetor resultante em relação ao eixo horizontal por meio da função arcotangente (*atan2*), convertendo o resultado para graus.

Além disso, o código fornece uma visualização gráfica da força resultante utilizando o módulo *matplotlib*. Ele plota o vetor da resultante no plano cartesiano, com uma escala adequada para facilitar a interpretação. O gráfico inclui eixos, grades, e uma legenda indicando a intensidade e o ângulo da força resultante. Finalmente, ele imprime no terminal os valores da intensidade e direção da resultante, fornecendo uma representação quantitativa e visual do sistema de forças.

Este programa é útil para aplicações práticas, como o estudo de equilíbrio de forças em estruturas ou o cálculo de vetores resultantes em dinâmica.

3 Questão 2

3.1 Cargas Pontuais

Figura 2: Estrutura Gerada.

Figura 3: Resultado obtido através do FTool

```
====================================
RESULTADO:
====================================

Carregamento 1:
Fx: 0.00
Fy: 13.00
Posição: 2.00

Carregamento 2:
Fx: 0.00
Fy: 5.00
Posição: 9.00

====================================

Reações nos apoios:
A_x: 0.00
A_y: 6.17
B_y: 11.83

====================================
```

Figura 4: Resultado obtido através do programa desenvolvido.

3.2 Carregamentos Distribuidos

3.2.1 Carregamento Retangular

Figura 5: Estrutura Gerada.

Figura 6: Resultado obtido através do FTool

```
==================================
RESULTADO:
==================================

Carregamento 1:
Fx: 0.00
Fy: 800.00
Posição: 7.00

==================================

Reações nos apoios:
A_x: 0.00
A_y: 800.00
M: 5600.00

==================================
```

Figura 7: Resultado obtido através do programa desenvolvido.

3.2.2 Carregamento Triangular

Figura 8: Estrutura Gerada.

Figura 9: Resultado obtido através do FTool

```
====================================
RESULTADO:
====================================

Carregamento 1:
Fx: 0.00
Fy: 240.00
Posição: 7.67

Carregamento 2:
Fx: 0.00
Fy: 5.00
Posição: 2.00

====================================

Reações nos apoios:
A_x: 0.00
A_y: 39.44
B_y: 205.56

====================================
```

Figura 10: Resultado obtido através do programa desenvolvido.

3.2.3 Carregamento Trapezoidal

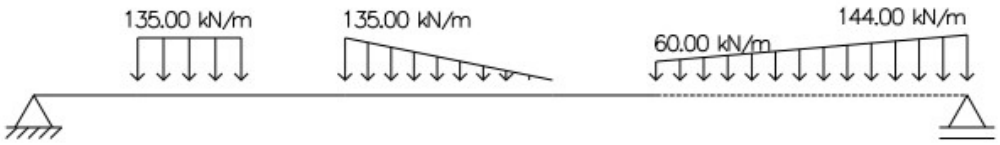


Figura 11: Estrutura Gerada.

Figura 12: Resultado obtido através do FTool

```
==================================
RESULTADO:
==================================

Carregamento 1:
Fx: 0.00
Fy: 135.00
Posição: 1.50

Carregamento 2:
Fx: 0.00
Fy: 135.00
Posição: 3.67

Carregamento 3:
Fx: 0.00
Fy: 306.00
Posição: 7.71

==================================

Reações nos apoios:
A_x: 0.00
A_y: 236.50
B_y: 339.50

==================================
```

Figura 13: Resultado obtido através do programa desenvolvido.

3.3 Funcionamento e Estrutura Geral

O programa é dividido em várias partes conectadas que permitem configurar uma viga com diferentes tipos de carregamentos e apoios, verificando sua estabilidade e determinando as reações nos apoios.

3.3.1 Definições de Enums

São criados dicionários para categorizar os tipos de carregamentos, apoios e estabilidade da viga:

- Tipos de carregamento: Retangular, triangular e trapézoidal.
- Tipos de apoio: Apoio de 1º, 2º e 3º gênero.
- Tipos de estabilidade: Hipoestática, isostática, e hiperestática.

3.3.2 Classes e Funções

- CarregamentoPontual
  - Representa uma carga pontual com intensidade, posição e angulação.
  - Utiliza a função 'separaComponentes' para calcular as componentes horizontal ('Fx') e vertical ('Fy') da força.
- CarregamentoDistribuido:
  - Representa um carregamento distribuído que pode ser retangular, triangular ou trapézoidal.
  - Calcula a função $w(x)$ para o carregamento e, a partir dela, a força resultante (integral de $w(x)$) e o centroide (posição da resultante).
- Apoio: Representa os diferentes tipos de apoios e suas características, como reações verticais e horizontais.
- gerar_numIncognitas: calcula o número de incógnitas no sistema, com base no tipo e na quantidade de apoios.
- get_estabilidade: Avalia a estabilidade da viga comparando o número de equações de equilíbrio (3 para sistemas planares) com o número de incógnitas.
- calcula_reacoes: Aplica o método dos momentos e as equações de equilíbrio de forças (em x e y) para calcular as reações nos apoios.

3.4 Interação com o Usuário

O código solicita informações sobre os carregamentos e apoios de forma interativa:

3.4.1 Carregamentos Pontuais:

- Intensidade, posição e angulação.

3.4.2 Carregamentos Distribuídos:

- Tipo de carregamento (retangular, triangular ou trapézoidal).
- Valores iniciais e finais do carregamento (w_0, w_1).
- Posições inicial e final (p_0, p_f).

3.4.3 Apoios:

- Tipo de apoio (1^{o}, 2^{o} ou 3^{o} gênero).
- Posição de cada apoio.

3.5 Resultados Gerados

Após a entrada de dados:

3.5.1 Validação da Estabilidade:

- Verifica se a viga é isostática.
- Em caso de hiperestaticidade ou hipoestaticidade, o programa encerra com um erro.

3.5.2 Cálculo de Reações:

- Determina as reações nos apoios usando as equações de equilíbrio:
  - Somatório das forças em x = 0
  - Somatório das forças em y = 0
  - Somatório dos momentos = 0

3.5.3 Exibição dos Resultados

- Mostra os valores das forças horizontais Fx e verticais Fy de cada carregamento.
- Exibe as reações em cada apoio calculadas.
- É exibido um gráfico com a plotagem das reações e das forças de cada carregamento.

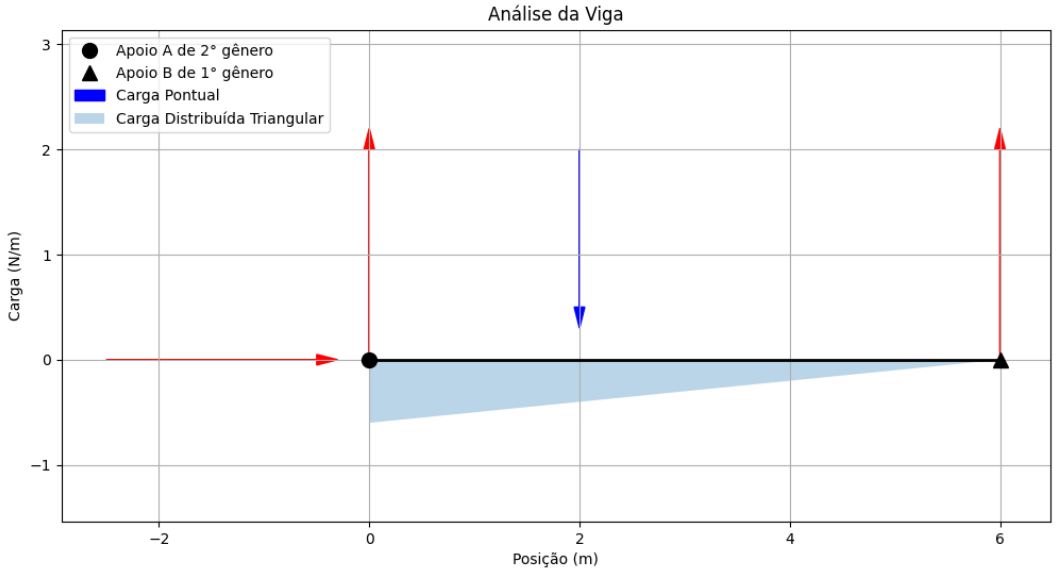


Figura 14: Gráfico gerado por um exemplo.

4 Questão 3

4.1 Aplicação em exemplos:

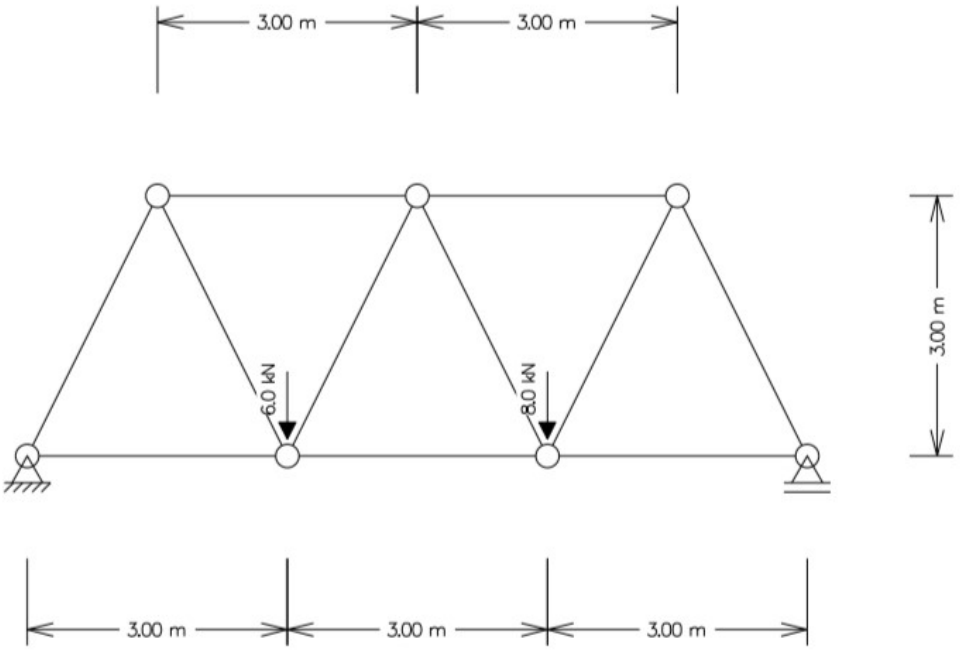


Figura 15: Estrutura Gerada.

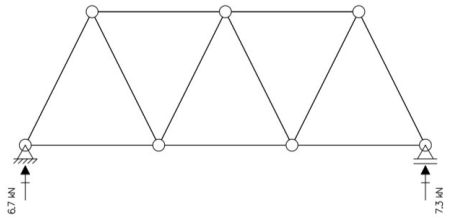


Figura 16: Resultado obtido através do FTool

```
==================================
RESULTADO:
==================================

Forças nas barras:
Barra 0: 7.45 kN (Tração)
Barra 1: 3.33 kN (Compressão)
Barra 2: 7.45 kN (Compressão)
Barra 3: 6.67 kN (Tração)
Barra 4: 0.75 kN (Tração)
Barra 5: 7.00 kN (Compressão)
Barra 6: 0.75 kN (Compressão)
Barra 7: 8.20 kN (Compressão)
Barra 8: 7.33 kN (Tração)
Barra 9: 8.20 kN (Tração)
Barra 10: 3.67 kN (Compressão)

==================================

Reações de apoio:

Nó 0:
Rx = 0.00 kN
Ry = 6.67 kN

Nó 6:
Ry = 7.33 kN
```

Figura 17: Resultado obtido através do programa desenvolvido.

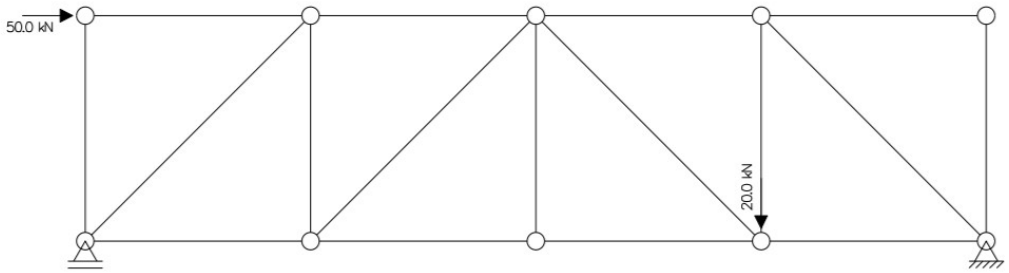


Figura 18: Estrutura Gerada.

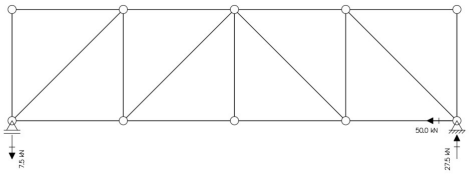


Figura 19: Resultado obtido através do FTool

```
==================================

Forças nas barras:
Barra 0: 7.50 kN (Tração)
Barra 1: 15.00 kN (Tração)
Barra 2: 15.00 kN (Tração)
Barra 3: 22.50 kN (Tração)
Barra 4: 0.00 kN (Compressão)
Barra 5: 0.00 kN (Compressão)
Barra 6: 27.50 kN (Tração)
Barra 7: 42.50 kN (Tração)
Barra 8: 50.00 kN (Tração)
Barra 9: 0.00 kN (Compressão)
Barra 10: 10.61 kN (Compressão)
Barra 11: 7.50 kN (Tração)
Barra 12: 10.61 kN (Compressão)
Barra 13: 0.00 kN (Compressão)
Barra 14: 10.61 kN (Tração)
Barra 15: 27.50 kN (Compressão)
Barra 16: 38.89 kN (Tração)

==================================

Reações de apoio:

Nó 0:
Ry = -7.50 kN

Nó 4:
Rx = -50.00 kN
Ry = 27.50 kN

==================================
```

Figura 20: Resultado obtido através do programa desenvolvido.

4.2 Funcionamento e Estrutura Geral

Este código resolve uma treliça utilizando o **Método de Equilíbrio dos Nós**, onde os nós são modelados como pontos no espaço, e as forças são calculadas com base no equilíbrio de forças. O programa possui uma estrutura modular dividida em três classes principais, que representam os componentes de uma treliça: **Nós, Barras,** e a própria **Treliça**. A resolução do sistema de equações é realizada através da montagem de um sistema linear com a biblioteca Sympy que considera as forças nos nós, as reações nos apoios e as propriedades das barras.

4.2.1 Classe Nó

A classe **No** representa um ponto na treliça, onde as forças podem ser aplicadas. Cada nó possui:

- **Coordenadas (x, y)**: Localização no plano cartesiano.
- **Forças (fx, fy)**: Forças aplicadas nas direções X e Y (opcionais).
- **Barras**: Barras conectadas ao nó.

- **Restrições de movimento**: Indica se o nó possui algum tipo de **Apoio** conectado à ele, podendo ser fixo ou móvel.

Métodos principais:

- `adicionar_barra`: Adiciona barras conectadas ao nó.
- `definir_apoio`: Define se o nó possui restrições em X ou Y (ou ambos).

4.2.2 Classe Barra

A classe **Barra** representa a conexão entre dois nós, com as propriedades de comprimento, força e direção. Cada barra tem:

- **Nós (no1, no2)**: Nós que estão conectados pela barra.
- **Comprimento**: Calculado com base na distância entre os dois nós.
- **Força**: Força interna na barra (calculada posteriormente).
- **Ângulo**: Ângulo da barra em relação ao eixo X.

Métodos principais:

- `calcular_comprimento`: Calcula o comprimento da barra usando a fórmula da distância euclidiana.
- `calcular_angulo`: Calcula o ângulo da barra em relação ao eixo X.
- `obter_cossenos_diretores`: Obtém os cossenos direcionais no eixo X e Y para aplicar no sistema de equações.

4.2.3 Classe Treliça

A classe **Treliça** gerencia a estrutura da treliça, incluindo a adição de nós, barras e apoios. Além disso, ela realiza a validação da isostaticidade e a resolução do sistema de equações. Cada treliça possui:

- **Lista de Nós**: Todos os nós da treliça.
- **Lista de Barras**: Todas as barras que conectam os nós.
- **Métodos**: Métodos para adicionar nós, barras e resolver o sistema linear de forças.

Métodos principais:

- `adicionar_no`: Adiciona um nó à treliça.

- `adicionar_barra`: Adiciona uma barra entre dois nós.
- `validar_isostatica`: Valida se a treliça é isostática (barras + numero de icógnitas = 2 * nós).
- `resolver`: Resolve o sistema de equações e determina as forças nas barras e reações nos apoios.
- `imprimir_resultados`: Exibe as forças nas barras e as reações nos apoios após a resolução.

4.3 Detalhamento do Cálculo

4.3.1 Funcionamento

O processo de cálculo segue os seguintes passos:

1. **Definição dos Nós**: O usuário insere as coordenadas dos nós e, opcionalmente, as forças externas aplicadas nos nós. Cada nó pode ser fixo ou móvel, dependendo das restrições de movimento que são definidas.
2. **Definição dos Apoios**: O programa permite definir o tipo de apoio em cada nó:
   - **Apoio de 1º gênero (móvel)**: Impede o movimento apenas no eixo Y.
   - **Apoio de 2º gênero (fixo)**: Impede o movimento nos eixos X e Y.
3. **Definição das Barras**: O usuário conecta os nós por meio das barras. Cada barra tem um comprimento, um ângulo e uma direção de força que será usada para resolver o sistema de equações.
4. **Visualização**: O sistema gera um gráfico plano cartesiano, plotando os nós, as barras e os apoios. Isso permite ao usuário visualizar a treliça e verificar se os dados foram inseridos corretamente.
5. **Exibição dos Resultados**: Após a resolução do sistema de equações, o sistema exibe as forças nas barras e as reações nos apoios. A força em cada barra pode ser classificada como **Tração** ou **Compressão**.

4.3.2 Exibição dos Resultados

Após a resolução, os resultados são apresentados:

- **Forças nas Barras**: O sistema calcula a força interna de cada barra. Se a força for positiva, significa que a barra está em **tração**; se for negativa, está em **compressão**.
- **Reações nos Apoios**: As forças de reação nos apoios são calculadas. Para cada nó com apoio, o programa exibe a reação nas direções X e Y.

4.4 Exemplo de Execução

O programa permite que o usuário insira os dados de uma treliça, conforme o exemplo abaixo:

- **Número de Nós**: O número total de nós a serem inseridos.
- **Apoios**: Definição do tipo de apoio para cada nó.
- **Barras**: Conexão entre os nós.

Após a inserção desses dados, o programa gera o gráfico da treliça e resolve o sistema linear para encontrar as forças nas barras e as reações nos apoios.