



**Ministério da Educação**  
**Universidade Federal da Integração Latino-Americana**  
**Instituto Latino-Americano de Tecnologia, Infraestrutura e Território**  
Centro Interdisciplinar de Tecnologia e Infraestrutura  
Engenharia Civil de Infraestrutura

# **OTIMIZAÇÃO TOPOLÓGICA DE TRELIÇAS BIDIMENSIONAIS PELO MÉTODO DOS ELEMENTOS FINITOS**

TIAGO HENRIQUE GONÇALVES DA SILVA

Foz do Iguaçu, PR

Abril de 2024



**Ministério da Educação**  
**Universidade Federal da Integração Latino-Americana**  
**Instituto Latino-Americano de Tecnologia, Infraestrutura e Território**  
Centro Interdisciplinar de Tecnologia e Infraestrutura  
Engenharia Civil de Infraestrutura

# **OTIMIZAÇÃO TOPOLÓGICA DE TRELIÇAS BIDIMENSIONAIS PELO MÉTODO DOS ELEMENTOS FINITOS**

TIAGO HENRIQUE GONÇALVES DA SILVA

Trabalho de Conclusão de Curso apresentado à Banca Examinadora do Curso de Engenharia Civil de Infraestrutura da UNILA, como parte dos requisitos para obtenção do Grau de Bacharel em Engenharia Civil.

**Orientador:** Prof<sup>o</sup>. Dr<sup>o</sup>. Aref Kalilo Lima Kzam

Foz do Iguaçu, PR

Abril de 2024

TIAGO HENRIQUE GONÇALVES DA SILVA

**OTIMIZAÇÃO TOPOLÓGICA DE TRELIÇAS BIDIMENSIONAIS PELO MÉTODO  
DOS ELEMENTOS FINITOS**

Trabalho de Conclusão de Curso apresentado à Banca Examinadora do Curso de Engenharia Civil de Infraestrutura da UNILA, como parte dos requisitos para obtenção do Grau de Bacharel em Engenharia Civil.

**BANCA EXAMINADORA**

---

Orientador: Prof. Dr. Aref Kalilo Lima Kzam  
UNILA

---

Prof. Dr. Paulo Junges  
UNILA

---

Prof. Dr. Ulises Bobadilla Guadalupe  
UNILA

Foz do Iguaçu, 29 de Abril de 2024.

Dedico este trabalho aos meus pais e ao meu irmão, pela inestimável ajuda e apoio.

## **AGRADECIMENTOS**

Agradeço aos meus pais e irmão que sempre me apoiaram.

Ao professor e orientador Aref Kalilo Lima Kzam, pela oportunidade e apoio na elaboração deste trabalho.

Aos professores por me proporcionar tantos conhecimentos ao longo do curso.

A todos meus amigos que direta ou indiretamente fizeram parte da minha formação.

Duvidar de tudo ou acreditar em tudo são  
duas soluções igualmente convenientes;  
ambas dispensam a necessidade de  
reflexão.

**Henri Poincaré**

## RESUMO

A otimização estrutural visa encontrar os valores ideais para variáveis, garantindo que todas as restrições sejam respeitadas, e utiliza critérios matemáticos e mecânicos para identificar projetos ótimos de estruturas ou elementos. Sua importância reside na maximização da confiabilidade, eficiência de custos e sustentabilidade ambiental. O contínuo desenvolvimento de novas construções proporciona uma oportunidade para aplicar avanços recentes na otimização estrutural, visando sistemas estruturais mais eficientes, mantendo os objetivos de projeto. A otimização topológica é um método que determina a distribuição ideal de material em um domínio de projeto, baseando-se na análise do comportamento mecânico da estrutura, considerando aspectos como flexibilidade, frequências naturais e distribuição de tensões. O objetivo do presente trabalho é determinar a distribuição de barras em treliças bidimensionais por meio da otimização topológica e do método dos elementos finitos. Este estudo propõe a otimização topológica de treliças bidimensionais através do método dos elementos finitos, no qual foram realizadas otimizações em três treliças com diferentes configurações de barras. Para o desenvolvimento do trabalho, foi elaborado um algoritmo em Fortran para otimização topológica. Esse algoritmo foi responsável por eliminar barras e nós que atendessem ao critério de otimização. Além disso, para auxiliar o algoritmo de otimização, foi implementado um código em Fortran para determinar os esforços e deslocamentos atuantes na treliça pelo método dos elementos finitos. Os resultados mostram a obtenção de diferentes distribuições de elementos nas treliças analisadas, resultando na redução do número de barras e, conseqüentemente, no peso total da estrutura.

**Palavras-chave:** Treliças; Otimização topológica; Método dos elementos finitos.

## RESUMEN

La optimización estructural tiene como objetivo encontrar los valores ideales para variables, garantizando que todas las restricciones sean respetadas, y utiliza criterios matemáticos y mecánicos para identificar diseños óptimos de estructuras o elementos. Su importancia radica en maximizar la confiabilidad, la eficiencia de costos y la sostenibilidad ambiental. El continuo desarrollo de nuevas construcciones proporciona una oportunidad para aplicar avances recientes en la optimización estructural, buscando sistemas estructurales más eficientes, manteniendo los objetivos de diseño. La optimización topológica es un método que determina la distribución ideal de material en un dominio de diseño, basándose en el análisis del comportamiento mecánico de la estructura, considerando aspectos como flexibilidad, frecuencias naturales y distribución de tensiones. El objetivo del presente trabajo es determinar la distribución de barras en trellas bidimensionales mediante la optimización topológica y el método de los elementos finitos. Este estudio propone la optimización topológica de trellas bidimensionales a través del método de los elementos finitos, en el cual se realizaron optimizaciones en tres trellas con diferentes configuraciones de barras. Para el desarrollo del trabajo, se elaboró un algoritmo en Fortran para la optimización topológica. Este algoritmo fue responsable de eliminar barras y nodos que cumplieran con el criterio de optimización. Además, para ayudar al algoritmo de optimización, se implementó un código en Fortran para determinar los esfuerzos y desplazamientos actuantes en la trella mediante el método de los elementos finitos. Los resultados muestran la obtención de diferentes distribuciones de elementos en las trellas analizadas, lo que resulta en la reducción del número de barras y, consecuentemente, en el peso total de la estructura.

**Palabras clave:** Celosías; Optimización topológica; Método de los elementos finitos.



## ABSTRACT

Structural optimization aims to find the ideal values for variables, ensuring that all constraints are respected, and it uses mathematical and mechanical criteria to identify optimal designs of structures or elements. Its importance lies in maximizing reliability, cost efficiency, and environmental sustainability. The continuous development of new constructions provides an opportunity to apply recent advances in structural optimization, aiming for more efficient structural systems while maintaining design goals. Topological optimization is a method that determines the ideal distribution of material within a design domain, based on the analysis of the structure's mechanical behavior, considering aspects such as flexibility, natural frequencies, and stress distribution. The objective of this study is to determine the distribution of bars in two-dimensional trusses through topological optimization and the finite element method. This study proposes the topological optimization of two-dimensional trusses through the finite element method, in which optimizations were performed on three trusses with different bar configurations. For the development of the work, a Fortran algorithm was developed for topological optimization. This algorithm was responsible for eliminating bars and nodes that met the optimization criterion. Additionally, to assist the optimization algorithm, a Fortran code was implemented to determine the forces and displacements acting on the truss by the finite element method. The results show the obtaining of different element distributions in the analyzed trusses, resulting in the reduction of the number of bars and consequently in the total weight of the structure.

**Keywords:** Trusses; Topology optimization; Finite element method.

## LISTA DE FIGURAS

<b>Figura 1</b> - Três categorias de otimização estrutural. a) Otimização paramétrica, b) otimização de forma e c) otimização topológica.....	17
<b>Figura 2</b> - Treliça .....	19
<b>Figura 3</b> - Elemento de treliça.....	20
<b>Figura 4</b> - Ângulo entre as coordenadas globais e coordenadas locais.....	22
<b>Figura 5</b> - Exemplo da entrada de dados.....	27
<b>Figura 6</b> - Estrutura do exemplo da entrada dados .....	28
<b>Figura 7</b> - Fluxograma do algoritmo de otimização.....	31
<b>Figura 8</b> - Treliça de 61 barras e 26 nós.....	32
<b>Figura 9</b> - Treliça otimizada para o problema de 61 barras. ....	33
<b>Figura 10</b> - Treliça de 51 barras e 26 nós.....	33
<b>Figura 11</b> - Treliça otimizada para o problema de 51 barras. ....	34
<b>Figura 12</b> - Treliça de 303 barras e 124 nós.....	35
<b>Figura 13</b> - Treliça otimizada para o problema de 303 barras. ....	36

## LISTA DE TABELAS

<b>Tabela 1</b> - Peso inicial e final para o problema de 61 barras. ....	33
<b>Tabela 2</b> - Peso inicial e final para o problema de 51 barras. ....	34
<b>Tabela 3</b> - Peso inicial e final para o problema de 303 barras. ....	36

## SUMÁRIO

1 INTRODUÇÃO E JUSTIFICATIVA .....	13
2 OBJETIVOS .....	15
2.1 Objetivo Geral .....	15
2.2 Objetivos Específicos .....	15
3 REVISÃO BIBLIOGRÁFICA .....	16
3.1 Otimização Topológica .....	16
3.2 Método dos Elementos Finitos .....	17
3.3 Método da rigidez .....	18
3.3.1 Matriz de transformação de força e deslocamento .....	21
3.3.2 Matriz de rigidez do elemento .....	24
3.3.3 Matriz de rigidez da treliça .....	25
4 MÉTODO .....	27
4.1 Algoritmo de otimização .....	27
5 RESULTADOS .....	32
5.1 Treliça – Exemplo 01 .....	32
5.2 Treliça – Exemplo 02 .....	33
5.3 Treliça – Exemplo 03 .....	34
6 CONCLUSÃO .....	37
REFERÊNCIAS BIBLIOGRÁFICAS .....	38
APÊNDICE I – CÓDIGO DE ELEMENTOS FINITOS DE TRELIÇA .....	40

## 1 INTRODUÇÃO E JUSTIFICATIVA

A otimização estrutural busca determinar os valores mais adequados para certas variáveis, visando alcançar projetos ótimos, enquanto é garantido que todas as restrições estabelecidas sejam atendidas. A identificação desses projetos ótimos é conduzida com base em uma variedade de critérios, empregando princípios matemáticos e mecânicos para gerar estruturas completas ou elementos estruturais (COUTINHO, 2006).

A otimização de estrutural desempenha uma função importante na maximização da confiabilidade, eficiência de custos e sustentabilidade ambiental (LIU et al., 2022). Ademais, a necessidade contínua de novas construções ofereceu uma oportunidade para ampliar os avanços recentes na otimização estrutural para viabilizar sistemas estruturais mais eficientes, ao mesmo tempo que cumprem os objetivos de projeto (GOLECKI et al., 2023).

A otimização topológica é um método para determinar a distribuição ideal de material dentro de um domínio de projeto definido. Essa técnica é fundamentada na análise do comportamento mecânico da estrutura, levando em consideração parâmetros como flexibilidade, frequências naturais e distribuição de tensões (SANT'ANNA, 2002). Ao operar dentro de um espaço de projeto pré-estabelecido e considerando condições de contorno específicas, a otimização topológica busca ajustar a disposição do material de forma a minimizar uma função objetivo, enquanto atende às restrições impostas. Isso é alcançado através de algoritmos que iterativamente modifica a distribuição de elementos, removendo elementos menos solicitados ou menos eficientes em termos de desempenho estrutural, resultando em estruturas mais leves, eficientes e econômicas. As técnicas de otimização topológica são amplamente reconhecidas e altamente eficazes para maximizar o desempenho das estruturas, otimizando a disposição de materiais (ZUO; HAN; LIU, 2023).

Na análise estrutural, o Método dos Elementos Finitos (MEF) emprega a discretização do contínuo e matriz de interpolação para calcular deslocamentos nodais e resolver problemas de engenharia, representando o modelo com um número finito de incógnitas (VAZ, 2011). Dentre as abordagens do MEF encontra-se o método de rigidez, que se baseia na subdivisão da estrutura em elementos finitos discretos e na identificação de seus pontos extremos como nós. O método de rigidez é capaz de

lidar tanto com estruturas isostáticas quanto hiperestáticas, além de oferecer uma forma direta de obter deslocamentos e forças.

Este trabalho propõe a elaboração de um algoritmo em Fortran 90 para otimização topológica de treliças bidimensionais utilizando o método dos elementos finitos. A implementação do algoritmo de otimização opera por meio de processos interativos, com o propósito de remover elementos de treliça. Nesse contexto, são considerados tanto os critérios de otimização quanto os critérios de parada como diretrizes fundamentais. Este trabalho busca contribuir para o avanço contínuo da otimização topológica, promovendo o desenvolvimento de estruturas mais eficientes, econômicas e sustentáveis.

## **2 OBJETIVOS**

### **2.1 Objetivo Geral**

Determinar a distribuição de barras em treliça bidimensionais por meio da otimização topológica e o método dos elementos finitos.

### **2.2 Objetivos Específicos**

Os objetivos específicos do presente trabalho são:

- a) Implementar e validar o código do método dos elementos finitos de treliça utilizando a linguagem Fortran
- b) Implementar e validar o algoritmo computacional de otimização topológica de treliça utilizando a linguagem Fortran.
- c) Avaliar a configuração final das treliças otimizadas por meio do algoritmo de otimização topológica.

## 3 REVISÃO BIBLIOGRÁFICA

### 3.1 Otimização Topológica

Otimização é uma abordagem matemática que visa minimizar ou maximizar uma função específica, conhecida como função objetivo. Um problema típico de otimização envolve parâmetros que restringem o espaço de busca da função (PAULINO, 2019). A estrutura genérica dos problemas de otimização é definida por:

$$\text{Minimizar: } f(x)$$

$$\text{Sujeito a: } h_i(x) = a_i, \text{ com } i = 1, 2, 3 \dots, m$$

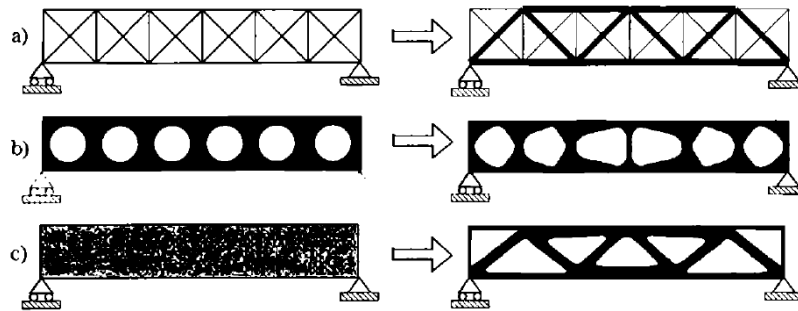
$$g_j(x) \leq b_j, \text{ com } j = 1, 2, 3 \dots, n$$

As funções  $h_i(x)$  e  $g_j(x)$  definem as restrições de igualdade e desigualdade, enquanto  $a_i$  e  $b_j$  são vetores que representam os limites dessas restrições para cada posição  $i$  e  $j$ , respectivamente.

Os problemas de otimização estrutural podem ser classificados em três diferentes tipos: otimização paramétrica, otimização de forma e otimização topológica (BENDSØE; SIGMUND, 2004). Na otimização paramétrica (**Figura 1a**), a forma da estrutura permanece constante, enquanto se otimiza a área da seção dos elementos. A otimização de forma (**Figura 1b**), ajusta a forma dos contornos da estrutura e a posição dos vazios por meio de uma função objetivo, entretanto a otimização de forma não permite modificar as propriedades topológicas da estrutura, como o número de vazios. Por fim, na otimização topológica (**Figura 1c**) a topologia da estrutura é alterada por meio da inserção de vazios no domínio.



**Figura 1** - Três categorias de otimização estrutural. a) Otimização paramétrica, b) otimização de forma e c) otimização topológica.



Fonte: Bendsøe e Sigmund (2004).

A otimização topológica (OT) consiste em determinar a melhor distribuição de material dentro de um determinado domínio de projeto (PAULINO, 2019). A OT modifica a distribuição de material no domínio baseado no comportamento mecânico da estrutura, como flexibilidade, frequências naturais, tensões, etc. (SANT'ANNA, 2002). A OT fundamenta-se em efetuar a distribuição de material dentro de um espaço de projeto previamente estabelecido, levando em consideração condições de contorno específicas, de modo a minimizar a função objetivo respeitando as restrições estabelecidas (COUTINHO, 2006).

### 3.2 Método dos Elementos Finitos

O reconhecimento internacional do Método dos Elementos Finitos (MEF) para a análise estrutural teve início nos anos cinquenta do século XX, marcado pelos trabalhos de Argyris (1960) e Turner (1956). Apesar disso, o pioneirismo do método remonta ao trabalho de Courant (1943), matemático alemão, cuja publicação sobre o problema de torção de Saint-Venant. Entretanto, na época, essa contribuição não recebeu grande atenção, possivelmente devido à relativa falta de interesse nos métodos numéricos, especialmente durante os estágios iniciais da era dos computadores. A contribuição de Zienkiewicz (1971) merece destaque, especialmente por sua obra intitulada "The Finite Element Methods for Engineering", que se tornou uma referência crucial na época para o estudo do Método dos Elementos Finitos (VAZ, 2011).

O MEF foi desenvolvido a partir da formulação em deslocamentos da análise matricial de estruturas reticuladas, impulsionada pelo avanço da computação. Embora

compartilhe semelhanças com seu precursor, como o uso de matriz de rigidez de elemento e montagem da matriz de rigidez da estrutura, o MEF se destaca por sua generalidade e suas raízes nos métodos de energia e aproximação. A abordagem matricial para a análise de estruturas reticuladas formalizou e organizou o método clássico dos deslocamentos, proporcionando uma metodologia unificada para a avaliação de uma variedade de estruturas reticuladas, incluindo treliças planas e espaciais, vigas, grelhas, pórticos planos e espaciais (VAZ, 2011).

Na área de análise estrutural, a formulação do MEF pode ser realizada por meio do Princípio da Mínima Energia Potencial Total, do Método de Resíduos Ponderados ou do Princípio dos Deslocamentos Virtuais. Essa abordagem emprega os conceitos de discretização do contínuo e de matriz de interpolação, que calcula os deslocamentos em qualquer ponto dentro do elemento com base nos deslocamentos nodais. A discretização refere-se à representação de um modelo com um número finito de incógnitas, em oposição à análise contínua da Teoria da Elasticidade, que utiliza funções contínuas e, portanto, possui um número infinito de incógnitas como solução (VAZ, 2011).

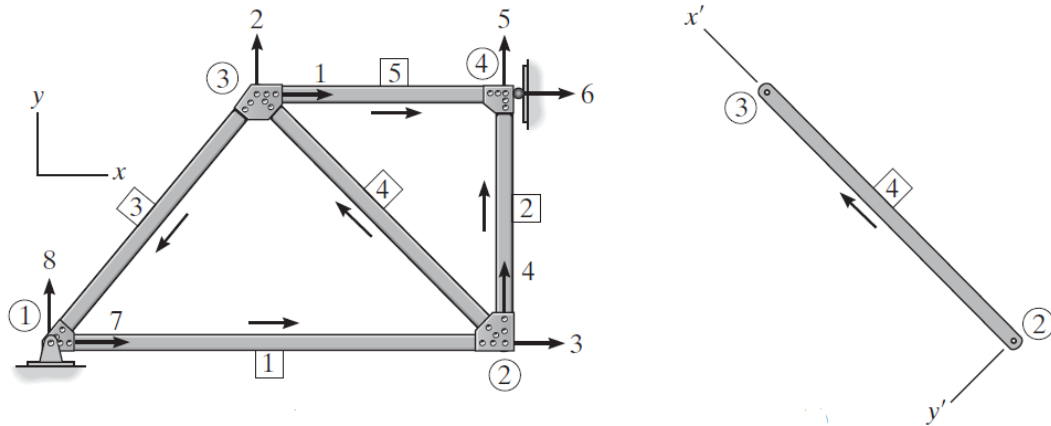
### 3.3 Método da rigidez

O Método dos Elementos Finitos (MEF) abordado neste trabalho é uma variante do Método dos Deslocamentos ou Método da Rigidez, onde os deslocamentos são selecionados como variáveis desconhecidas. Essa abordagem, comum em diversos métodos relacionados, é caracterizada pela utilização da equação de equilíbrio como equação fundamental, na qual os deslocamentos generalizados como deslocamento lineares e rotações são considerados como incógnitas (VAZ, 2011).

O método de rigidez é uma abordagem fundamental na análise de estruturas que se baseia na subdivisão da estrutura em elementos finitos discretos e na identificação de seus pontos extremos como nós. O método de rigidez é amplamente utilizado devido à sua capacidade de lidar tanto com estruturas estaticamente determinadas quanto indeterminadas, além de oferecer uma forma direta de obter deslocamentos e forças. As deduções do método da rigidez que serão apresentadas a seguir são baseadas na abordagem apresentada por Hibbeler (2013).

Para aplicar o método de rigidez, é necessário primeiro identificar os elementos ou membros da estrutura e seus nós. Cada elemento e nó é especificado por um número. A Figura 2 apresenta um exemplo uma treliça descrita por barras e nós.

**Figura 2 - Treliça**



**Fonte:** Hibbeler (2013).

As coordenadas globais definem o sentido correto das direções de carga e deslocamento. Um sistema de coordenadas global  $(x, y)$  é utilizado para especificar o sentido de cada componente de força externa e deslocamento nos nós, enquanto um sistema de coordenadas local ou do elemento  $(x', y')$  é empregado para especificar o sentido de direção dos deslocamentos e cargas internas em cada elemento.

Outro conceito importante é a indeterminação cinemática, que se refere aos graus de liberdade não restringidos de uma estrutura. Para treliças cada nó possui dois graus de liberdade, ou seja, dois possíveis deslocamentos. Esses graus de liberdade são identificados por meio de códigos numéricos.

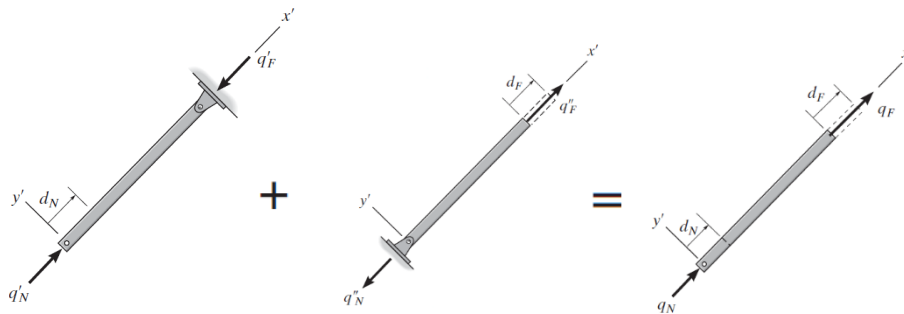
Uma vez que a estrutura é identificada e os nós são codificados, é possível determinar a matriz de rigidez da estrutura  $(K)$ . Para isso, é necessário estabelecer uma matriz de rigidez do elemento para cada elemento da estrutura e transformar essas matrizes das coordenadas locais para as coordenadas globais da estrutura. Essa transformação é realizada por meio de matrizes de transformação de força e deslocamento, permitindo que os deslocamentos desconhecidos dos nós sejam determinados para qualquer carga dada sobre a estrutura. Assim, o método de rigidez oferece uma abordagem para a análise de estruturas, permitindo calcular tanto os deslocamentos quanto as forças internas e externas.

Na Figura 3, são apresentadas as coordenadas locais ( $x'$ ,  $y'$ ) de um componente individual em uma treliça, para as quais será definida a matriz de rigidez. Os elementos desta matriz representam as interações entre carga e deslocamento. No contexto desta estrutura, a movimentação do componente ocorre exclusivamente ao longo do seu eixo (eixo  $x'$ ), devido à aplicação das cargas nessa direção. Como resultado, há dois deslocamentos independentes viáveis. Quando um deslocamento positivo  $d_N$  é imposto à extremidade próxima do componente, enquanto a extremidade distante permanece articulada, ocorrem forças resultantes nas extremidades do componente conforme as equações Eq. (1) e Eq. (2).

$$q'_N = \frac{AE}{L} d_N \quad (1)$$

$$q'_F = -\frac{AE}{L} d_N \quad (2)$$

**Figura 3** - Elemento de treliça



**Fonte:** Hibbeler (2013).

O sinal negativo de  $q'_F$  indica que força age na direção negativa do eixo  $x'$  para alcançar o equilíbrio. Da mesma forma, ocorre deslocamento positivo  $d_F$  no extremo afastado, enquanto o extremo próximo articulado, leva às seguintes forças de elemento conforme as equações Eq. (3) e Eq. (4).

$$q''_N = -\frac{AE}{L} d_F \quad (3)$$

$$q_F'' = \frac{AE}{L} d_F \quad (4)$$

Aplicando o princípio da superposição de efeitos as forças resultantes para os dois deslocamentos são escritas conforme as equações Eq. (5) e Eq. (6).

$$q_N = \frac{AE}{L} d_F - \frac{AE}{L} d_F \quad (5)$$

$$q_N = -\frac{AE}{L} d_F + \frac{AE}{L} d_F \quad (6)$$

Ambas as equações Eq. (5) e Eq. (6) descritas matricialmente resultam nas equações Eq. (7) e Eq. (8).

$$\begin{bmatrix} q_N \\ q_F \end{bmatrix} = \frac{AE}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} d_N \\ d_F \end{bmatrix} \quad (7)$$

$$\mathbf{q} = \mathbf{k}' \mathbf{d} \quad (8)$$

A equação Eq. (9) apresenta a matriz de rigidez do elemento, sendo similar para cada elemento da treliça. Os elementos que compõem a matriz de rigidez são denominados coeficientes de influência de rigidez do elemento,  $k'_{ij}$ . O coeficiente  $k'_{ij}$  descreve a força aplicada no nó  $i$  em resposta a um deslocamento unitário aplicado no nó  $j$  (HIBBELER, 2013).

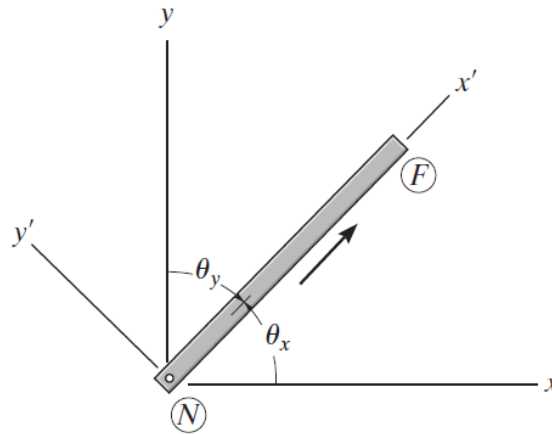
$$\mathbf{k}'' = \frac{AE}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \quad (9)$$

### 3.3.1 Matriz de transformação de força e deslocamento

Como uma treliça é composta por vários elementos, convém adotar um método para transformar as forças dos elementos  $q$  e deslocamentos  $d$  descritos por meio de coordenadas locais para coordenadas globais (HIBBELER, 2013). Os ângulos formados entre o entre os eixos globais  $x, y$  e o eixo local  $x'$  são definidos como  $\theta_x$  e

$\theta_y$ , conforme apresentado na Figura 4. Os cossenos dos ângulos entre os eixos, utilizados para definir a matriz de transformação de deslocamento e força, são apresentados nas equações Eq. (10) e Eq. (11).

**Figura 4** - Ângulo entre as coordenadas globais e coordenadas locais



Fonte: Hibbeler (2013).

$$\lambda_x = \cos \theta_x = \frac{x_F - x_N}{L} = \frac{x_F - x_N}{\sqrt{(x_F - x_N)^2 + (y_F - y_N)^2}} \quad (10)$$

$$\lambda_y = \cos \theta_y = \frac{y_F - y_N}{L} = \frac{y_F - y_N}{\sqrt{(x_F - x_N)^2 + (y_F - y_N)^2}} \quad (11)$$

Cada nó da treliça possui dois graus de liberdade representados pelos deslocamentos  $D_{Nx}$  e  $D_{Ny}$  no nó  $N$ , enquanto o nó  $F$  apresenta os deslocamentos independentes  $D_{Fx}$  e  $D_{Fy}$ . O resultado dos deslocamentos globais dos nós faz com que cada elemento apresente os deslocamentos descritos pelas equações Eq. (12) e Eq. (13).

$$d_N = D_{Nx} \cos \theta_x + D_{Ny} \cos \theta_y \quad (12)$$

$$d_F = D_{Fx} \cos \theta_x + D_{Fy} \cos \theta_y \quad (13)$$

Considerando os cossenos diretores como  $\lambda_x = \cos \theta_x$  e  $\lambda_y = \cos \theta_y$  obtém-se as equações Eq. (14) e Eq. (15).

$$d_N = D_{Nx}\lambda_x + D_{Ny}\lambda_y \quad (14)$$

$$d_F = D_{Fx}\lambda_x + D_{Fy}\lambda_y \quad (15)$$

Escrevendo as equações Eq. (14) e Eq. (15) anteriores na forma de matriz resulta nas equações Eq. (16) e Eq. (17).

$$\begin{bmatrix} d_N \\ d_F \end{bmatrix} = \begin{bmatrix} \lambda_x & \lambda_y & 0 & 0 \\ 0 & 0 & \lambda_x & \lambda_y \end{bmatrix} \begin{bmatrix} D_{Nx} \\ D_{Ny} \\ D_{Fx} \\ D_{Fy} \end{bmatrix} \quad (16)$$

$$\mathbf{d} = \mathbf{T}\mathbf{D} \quad (17)$$

A matriz  $\mathbf{T}$  apresentada na equação Eq. (18) é denominada matriz de transformação de deslocamento, responsável por converter os quatro deslocamentos globais  $\mathbf{D}$  nos dois deslocamentos locais  $\mathbf{d}$ .

$$\mathbf{T} = \begin{bmatrix} \lambda_x & \lambda_y & 0 & 0 \\ 0 & 0 & \lambda_x & \lambda_y \end{bmatrix} \quad (18)$$

Caso uma força  $q_N$  seja aplicada em nó enquanto a outra extremidade é mantida fixa, as componentes de força globais de  $q_N$  e  $q_F$  resultam nas equações Eq. (19), Eq. (20), Eq. (21) e Eq. (22). As quatro equações podem ser descritas na forma matricial conforme apresentando nas equações Eq. (23) e Eq. (24).

$$Q_{Nx} = q_N \cos \theta_x \quad (19)$$

$$Q_{Ny} = q_N \cos \theta_y \quad (20)$$

$$Q_{Fx} = q_F \cos \theta_x \quad (21)$$

$$Q_{Fy} = q_F \cos \theta_y \quad (22)$$

$$\begin{bmatrix} Q_{Nx} \\ Q_{Ny} \\ Q_{Fx} \\ Q_{Fy} \end{bmatrix} = \begin{bmatrix} \lambda_x & 0 \\ \lambda_y & 0 \\ 0 & \lambda_x \\ 0 & \lambda_y \end{bmatrix} \begin{bmatrix} q_N \\ q_F \end{bmatrix} \quad (23)$$

$$\mathbf{Q} = \mathbf{T}^T \mathbf{q} \quad (24)$$

A matriz de transformação de forças, apresentada na equação Eq. (25), converte as duas forças locais  $\mathbf{q}$  que atuam nos nós do elemento em quatro componentes de forças globais.

$$\mathbf{T}^T = \begin{bmatrix} \lambda_x & 0 \\ \lambda_y & 0 \\ 0 & \lambda_x \\ 0 & \lambda_y \end{bmatrix} \quad (25)$$

### 3.3.2 Matriz de rigidez do elemento

Reunindo as equações descritas anteriormente, é possível estabelecer a relação entre os componentes de força global  $\mathbf{Q}$  do elemento de treliça e os deslocamentos globais  $\mathbf{D}$ , conforme demonstram as equações Eq. (26), Eq. (27), Eq. (28) e Eq. (29).

$$\mathbf{q} = \mathbf{k}' \mathbf{T} \mathbf{D} \quad (26)$$

$$\mathbf{Q} = \mathbf{T}^T \mathbf{k}' \mathbf{T} \mathbf{D} \quad (27)$$

$$\mathbf{Q} = \mathbf{k} \mathbf{D} \quad (28)$$

$$\mathbf{k} = \mathbf{T}^T \mathbf{k}' \mathbf{T} \quad (29)$$

As equações Eq. (30) e Eq. (31) descrevem a matriz de rigidez  $\mathbf{k}$  do elemento em coordenadas globais.



$$\mathbf{k} = \begin{bmatrix} \lambda_x & 0 \\ \lambda_y & 0 \\ 0 & \lambda_x \\ 0 & \lambda_y \end{bmatrix} \frac{AE}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} \lambda_x & \lambda_y & 0 & 0 \\ 0 & 0 & \lambda_x & \lambda_y \end{bmatrix} \quad (30)$$

$$\mathbf{k} = \frac{AE}{L} \begin{bmatrix} \lambda_x^2 & \lambda_x \lambda_y & -\lambda_x^2 & -\lambda_x \lambda_y \\ \lambda_x \lambda_y & \lambda_y^2 & -\lambda_x \lambda_y & -\lambda_y^2 \\ -\lambda_x^2 & -\lambda_x \lambda_y & \lambda_x^2 & \lambda_x \lambda_y \\ -\lambda_x \lambda_y & -\lambda_y^2 & \lambda_x \lambda_y & \lambda_y^2 \end{bmatrix} \quad (31)$$

### 3.3.3 Matriz de rigidez da treliça

Para determinar a matriz de rigidez da estrutura é reunir as matrizes de rigidez dos elementos (HIBBELER, 2013). A matriz de rigidez da estrutura  $\mathbf{K}$  é determinada pela soma da matriz de rigidez de cada elemento da treliça conforme apresentado na equação Eq. (32), onde  $n$  corresponde ao número de elementos da treliça.

$$\mathbf{K} = \sum_{i=1}^n \mathbf{k}_i \quad (32)$$

A matriz de rigidez da treliça  $\mathbf{K}$  relaciona as componentes de força global  $\mathbf{Q}$  com o deslocamento global  $\mathbf{D}$  de acordo com a equação Eq. (33).

$$\mathbf{Q} = \mathbf{K}\mathbf{D} \quad (33)$$

Considerando que haverá cargas externas e deslocamentos conhecidos  $\mathbf{Q}_k$  e  $\mathbf{D}_k$ , bem como cargas e deslocamentos desconhecidos a equação Eq. (32) pode ser escrita conforme mostra a equação Eq. (34).

$$\begin{bmatrix} \mathbf{Q}_k \\ \mathbf{Q}_u \end{bmatrix} = \begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{21} & \mathbf{K}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{D}_u \\ \mathbf{D}_k \end{bmatrix} \quad (34)$$

Considerando os descolamentos dos apoios nulos  $\mathbf{D}_k = 0$ . Dessa forma, obtém-se a equação Eq. (35) e Eq. (36).

$$\mathbf{Q}_k = \mathbf{K}_{11}\mathbf{D}_u \quad (35)$$

$$\mathbf{Q}_u = \mathbf{K}_{21} \mathbf{D}_u \quad (36)$$

Da equação Eq. (35) pode-se obter os deslocamentos desconhecido conforme a equação Eq. (37). Enquanto a partir da equação Eq. (36), podem-se determinadas as reações de apoio.

$$\mathbf{D}_u = [\mathbf{K}_{11}]^{-1} \mathbf{Q}_k \quad (37)$$

Os esforços atuantes nos elementos de treliça descritos pela equação Eq. (26) pode ser expandido para a equação Eq. (38).

$$\begin{bmatrix} q_N \\ q_F \end{bmatrix} = \frac{AE}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} \lambda_x & \lambda_y & 0 & 0 \\ 0 & 0 & \lambda_x & \lambda_y \end{bmatrix} \begin{bmatrix} D_{Nx} \\ D_{Ny} \\ D_{Fx} \\ D_{Fy} \end{bmatrix} \quad (38)$$

Por equilíbrio  $q_N = -q_F$ , dessa forma pode-se calcular apenas uma das forças de acordo com a equação Eq. (39).

$$q_F = \frac{AE}{L} \begin{bmatrix} -\lambda_x & -\lambda_y & \lambda_x & \lambda_y \end{bmatrix} \begin{bmatrix} D_{Nx} \\ D_{Ny} \\ D_{Fx} \\ D_{Fy} \end{bmatrix} \quad (39)$$

## 4 MÉTODO

### 4.1 Algoritmo de otimização

Na primeira seção do código, são declaradas todas as variáveis necessárias para o funcionamento do programa, juntamente com o módulo que contém a sub-rotina Truss\_2D apresentada no **APÊNDICE I** para o cálculo da estrutura pelo método dos elementos finitos, por meio da sub-rotina são determinados os esforços e os deslocamentos da estrutura. Essa organização modular torna o código mais legível e gerenciável, facilitando a depuração do código.

A entrada de dados é fornecida por meio de um arquivo de texto conforme o exemplo apresentado na Figura 5 e Figura 6. Em seguida, a sub-rotina Truss\_2D é chamada para calcular os esforços iniciais na treliça, utilizando os dados de entrada fornecidos. Essa etapa é crucial para estabelecer uma base para o processo de otimização subsequente.

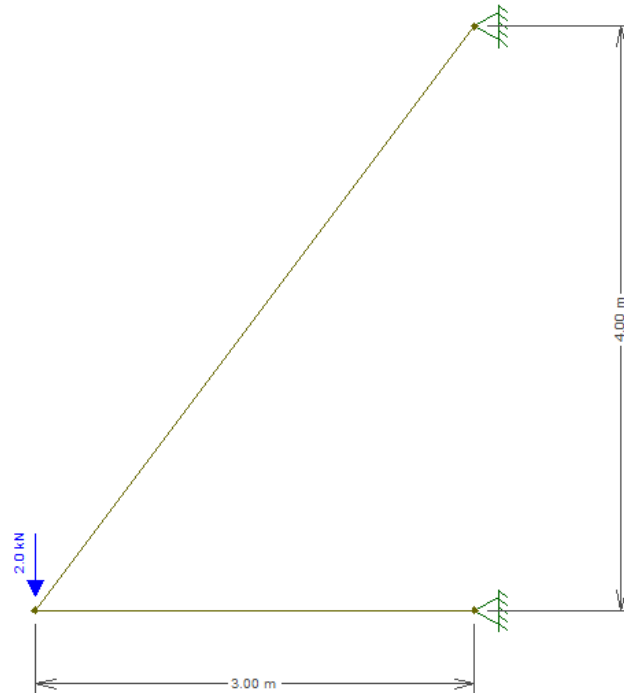
A partir desse ponto, o programa entra em um loop principal, onde são realizadas as iterações de otimização da treliça. Esse loop é essencial para permitir que o programa refine continuamente a estrutura da treliça até que alguma condição de parada seja atendida.

Durante o processo de otimização, variáveis são armazenadas para o caso de a matriz se tornar singular. Essa verificação é essencial para manter a estabilidade do algoritmo e garantir que a análise da estrutura seja precisa e confiável.

**Figura 5** - Exemplo da entrada de dados

```
numero de nós da treliça 2D (numn)
3
numero de elemento da treliça 2D (nume)
2
numero de propriedades geométricas: área da seção (Asec) e modulo de elasticidade do material (Emat)
1
numero de nós com deslocamento prescrito
2
Numero de nós com força prescrita
1
Cordenadas dos nós da treliça (idn,crdx,crdy)
1, 3.0 4.0
2, 0.0 0.0
3, 3.0 0.0
Propriedades do elemento (Idprop, Asec, Emat, Sigma_yield)
1, 1.0 1.0 1.0
Incidencia cinemática (elemento, no i, no j, proele )
1, 2 3, 1
2, 2 1, 1
Aplicação da condição de contorno em deslocamento (no, id_dx, id_dy, dx, dy) id: 0-não existe 1-existe
1, 1 1 0.0 0.0
3, 1 1 0.0 0.0
Aplicação da condição de contorno em força (no, id_fx, id_fy, fx, fy) id: 0-não existe 1-existe
2, 0 1 0.0 -2.0
```

Fonte: Autor.

**Figura 6** - Estrutura do exemplo da entrada dados

**Fonte:** Autor.

Dentro do loop de otimização, uma verificação é realizada para determinar se alguma barra da treliça apresenta tensão superior ao limite adotado. Se essa condição for atendida, o loop é encerrado, e os resultados são registrados. Em seguida as barras são ordenadas em ordem crescente de tensão absoluta. Essa ordenação facilita a identificação das barras menos solicitadas, permitindo que o algoritmo seja mais eficiente.

O critério otimização adotado para a remoção de um elemento de treliça foi a tensão mínima atuante na barra. Em geral, barras com tensões abaixo de um certo limiar são candidatas à remoção. No código, esse limiar foi definido como 20% da tensão de escoamento do material.

Antes de remover uma barra, é importante verificar se a sua remoção não compromete as condições de contorno da treliça. Condições de contorno, como deslocamentos ou forças prescritas em determinados nós, restringem a remoção de barras e nós. Durante o processo de remoção de uma barra também é necessário verificar se a remoção afeta a conexão entre os nós da treliça. Se uma barra conectada dois nós apresentar um nó que não estão conectados por outras barras, a remoção dessa barra pode desconectar o nó não conectados a outra barra. Após a

verificação dos critérios e condições, a remoção de uma barra pode ocorrer de duas maneiras: remover apenas a barra e manter ambos os nós ou remover a barra e um nó não associado a outra barra. A escolha entre essas opções depende das conexões entre os nós e barras da treliça.

Após a remoção de barras e nós, a treliça é reconstruída com base na nova configuração. Isso pode envolver a renumeração dos nós e a redefinição das conexões entre os nós e as barras restantes. O processo de remoção de uma barra termina com a elaboração de uma nova entrada de dados, desconsiderando a barra removido, bem como nó caso seja removido juntamente com a barra. Com a nova entrada de dados montada a estrutura é recalculada com a sub-rotina `Truss_2D` e os critérios de otimização de são novamente aplicados. Esse processo iterativo permite refinamentos contínuos na estrutura da treliça até que um elemento de treliça atinja a tensão máxima adotada, no caso adotou-se como tensão máxima a tensão de escoamento do material, o algoritmo também é finalizado caso não seja possível remover elementos sem comprometer o equilíbrio da estrutura.

Por fim, os resultados finais são registrados em arquivos de saída, apresentando os nós e elementos da estrutura otimizada, bem como os descolamentos e os esforços em cada nó e elementos, respectivamente. Além disso, um arquivo de saída é gerado para ser lido e visualizado no software `AcadView`.

Durante o processo de remoção de elementos, é importante verificar se a matriz de rigidez da treliça se torna singular. Se isso ocorrer, significa que a treliça perdeu estabilidade e deve ser revertida para uma configuração anterior. A detecção e tratamento da singularidade garante a validade dos resultados da otimização.

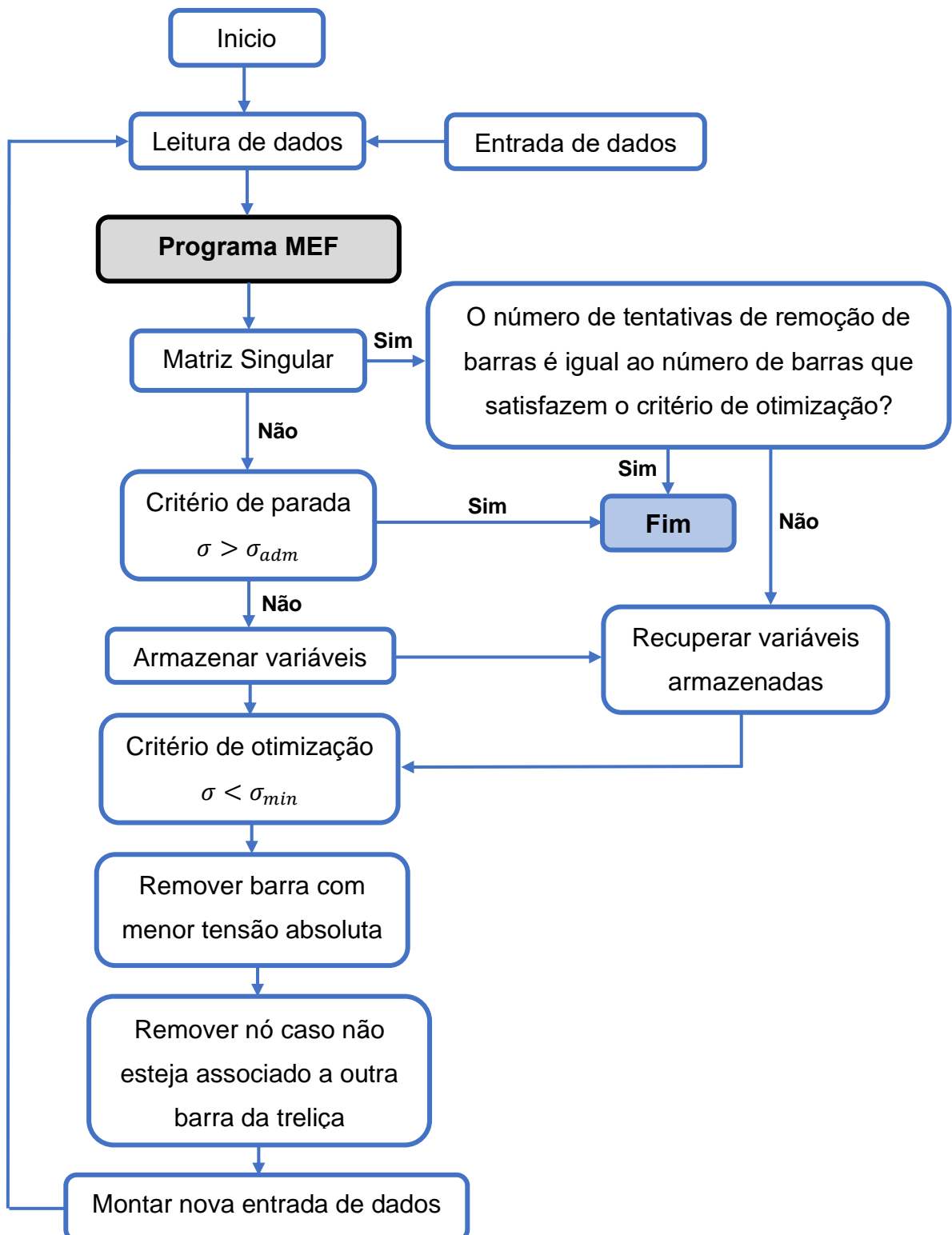
Para lidar com a singularidade, o programa precisa reverter a estrutura para um estado anterior onde a matriz de rigidez era não singular. Isso geralmente envolve desfazer as alterações que levaram à condição singular, como a remoção de barras ou nós. A reversão precisa ser feita com cuidado para evitar perdas de dados ou inconsistências na estrutura.

Após reverter a estrutura, o programa recalcula os parâmetros da treliça e retoma o processo de análise estrutural. Isso pode incluir a reintrodução de elementos removidos anteriormente para garantir a estabilidade da estrutura. A recuperação da estrutura é essencial para garantir que a análise possa continuar sem interrupções.

Uma vez que a estrutura tenha sido recuperada para um estado estável, o programa retoma a análise estrutural e continua iterando para o próximo elemento de treliça que atenda a critério de otimização até que uma solução viável seja encontrada. Esse processo pode envolver várias iterações de remoção de elementos, recuperação da estrutura e análise iterativa. A repetição da análise é necessária para garantir que a estrutura atenda aos requisitos de estabilidade.

Quando ocorrem matrizes singulares devido à hipoestaticidade da estrutura na tentativa de remoção de todos os elementos de treliças remanescentes que atendam o critério de otimização o algoritmo reverte a estrutura para um estado anterior onde a matriz de rigidez era não singular e finaliza o algoritmo. A Figura 7 apresenta um fluxograma contendo o algoritmo de otimização descrito anteriormente.

Figura 7 - Fluxograma do algoritmo de otimização



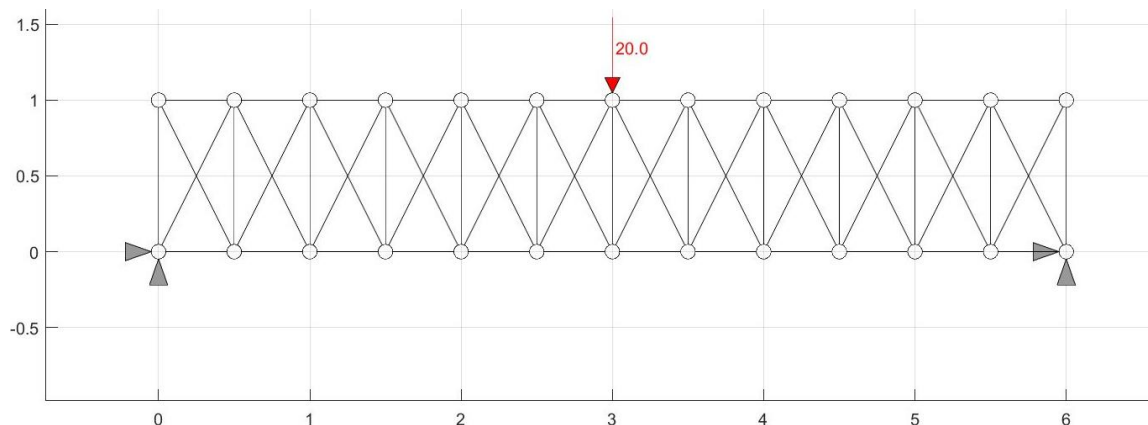
Fonte: Autor.

## 5 RESULTADOS

### 5.1 Treliça – Exemplo 01

O primeiro exemplo trata-se de uma treliça plana com banzos paralelos com 26 nós e 61 barras. A área da seção transversal de cada barra é de 1,23 cm<sup>2</sup>. O material possui módulo de elasticidade longitudinal de 200 GPa, tensão de escoamento de 250 MPa e peso específico de 7850 kgf/m<sup>3</sup>. A treliça apresenta dois apoios de segundo gênero nas extremidades em uma carga pontual de 20 kN conforme mostra a Figura 8.

**Figura 8** - Treliça de 61 barras e 26 nós.

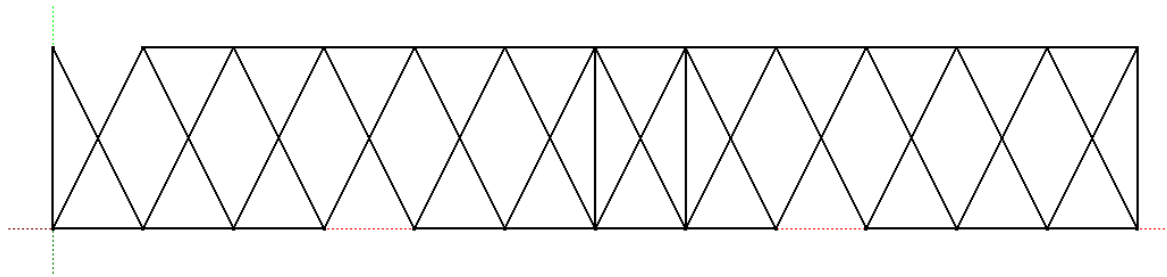


**Fonte:** Autor.

Após a execução do código de otimização ocorreu a remoção de 12 elementos de treliça, entretanto não ocorreu a remoção de nenhum nó. A Figura 9 apresenta a treliça otimizada, nota-se que algoritmo removeu principalmente os montantes da treliça, enquanto manteve todas as diagonais. A Tabela 1 apresenta o peso inicial e final da treliça otimizada.



**Figura 9** - Treliça otimizada para o problema de 61 barras.



Fonte: Autor.

**Tabela 1** - Peso inicial e final para o problema de 61 barras.

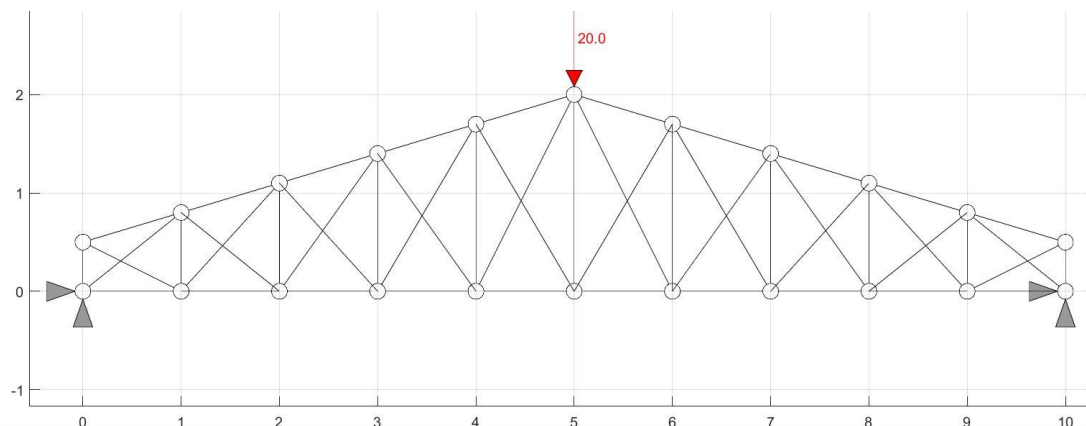
	Comprimento Total (m)	Área dos Elementos (m <sup>2</sup> )	Volume (m <sup>3</sup> )	Peso Específico (kgf/m <sup>3</sup> )	Peso Total (kgf)	Dif. (%)
Treliça Inicial	51,83	0,000123	0,006375	7850	50,05	20,3
Treliça Otimizada	41,33	0,000123	0,005084	7850	39,91	

Fonte: Autor.

## 5.2 Treliça – Exemplo 02

O segundo exemplo trata-se de uma treliça plana com 22 nós e 51 barras. A área da seção transversal de cada barra é de 1,23 cm<sup>2</sup>. O material possui módulo de elasticidade longitudinal de 200 GPa, tensão de escoamento de 250 MPa e peso específico de 7850 kgf/m<sup>3</sup>. A treliça apresenta dois apoios de segundo gênero nas extremidades em uma carga pontual de 20 kN conforme mostra a Figura 10.

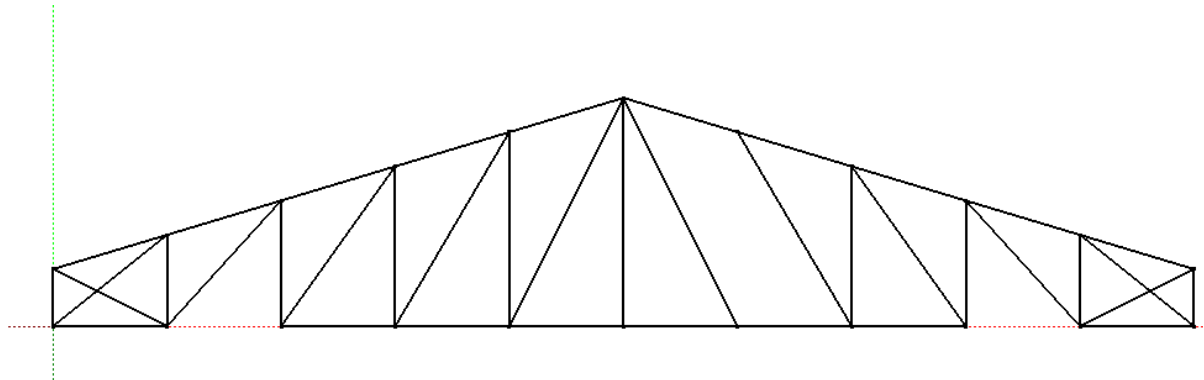
**Figura 10** - Treliça de 51 barras e 26 nós.



Fonte: Autor.

Após a execução do código de otimização, 11 elementos de treliça foram removidos, porém nenhum nó foi eliminado. A Figura 11 mostra a treliça otimizada, notando-se sua simetria em relação eixo central da estrutura. Na Tabela 2, são apresentados os pesos inicial e final da treliça otimizada.

**Figura 11** - Treliça otimizada para o problema de 51 barras.



Fonte: Autor.

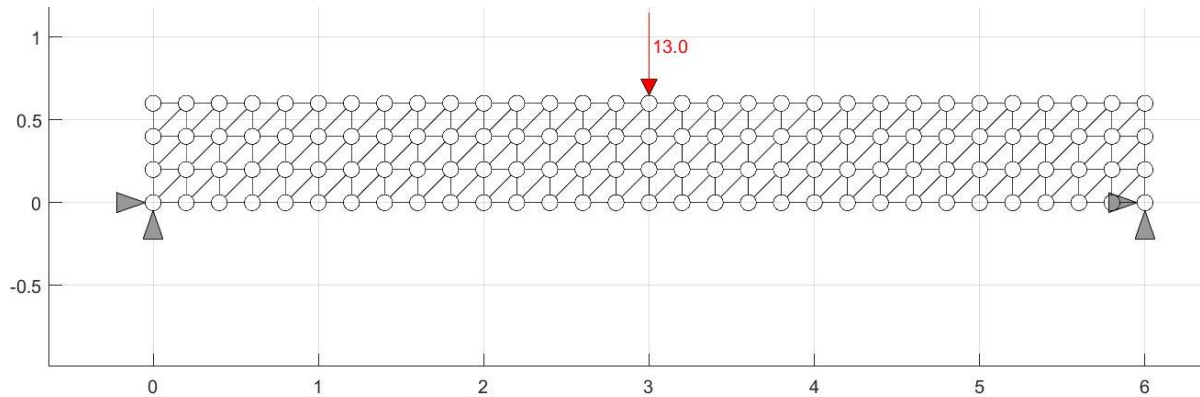
**Tabela 2** - Peso inicial e final para o problema de 51 barras.

	Comprimento Total (m)	Área dos Elementos (m <sup>2</sup> )	Volume (m <sup>3</sup> )	Peso Específico (kgf/m <sup>3</sup> )	Peso Total (kgf)	Dif. (%)
Treliça Inicial	65,99	0,000123	0,008117	7850	63,72	25,2
Treliça Otimizada	49,37	0,000123	0,006072	7850	47,67	

Fonte: Autor.

### 5.3 Treliça – Exemplo 03

O terceiro exemplo trata-se de uma treliça plana com banzos paralelo com 124 nós e 303 elementos. A área da seção transversal de cada barra é de 1,23 cm<sup>2</sup>. O material possui módulo de elasticidade longitudinal de 200 GPa, tensão de escoamento de 250 MPa e peso específico de 7850 kgf/m<sup>3</sup>. A Figura 12 mostra a treliça com uma carga pontal de 13 kN e dois apoios de segundo gênero em cada extremo.

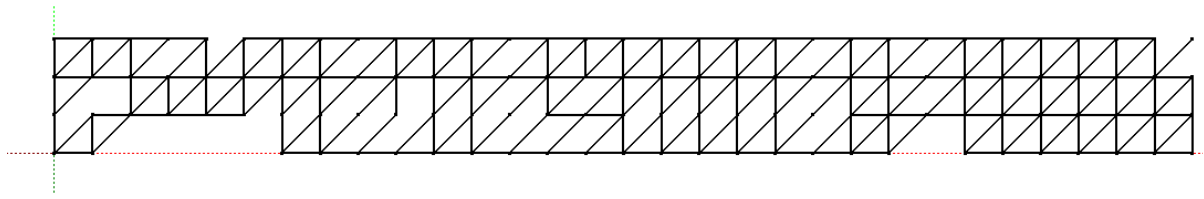
**Figura 12** - Treliça de 303 barras e 124 nós.

**Fonte:** Autor.

A Figura 13 apresenta a estrutura otimizada, ocorreram ao todo a remoção de 54 elementos de treliça e 5 nós, nota-se a estrutura apresentou uma assimetria na distribuição de elementos. Essa assimetria pode ser atribuída à lógica de execução do algoritmo de otimização. Quando um elemento é removido, a estrutura torna-se assimétrica e é recalculada, resultando em um cálculo que considera essa assimetria. Isso afeta todos os elementos removidos posteriormente, criando uma assimetria que se propaga ao longo das remoções subsequentes. Esse efeito tornou-se evidente em treliça com malha densa. A Tabela 3 apresenta o peso inicial e final da treliça otimizada.

Neste exemplo, foi realizada uma avaliação do desempenho do algoritmo em uma treliça com malha de alta densidade. Durante a execução do algoritmo, observou-se a remoção dos elementos da treliça de acordo com o critério de otimização. No entanto, foi identificada uma inconsistência em uma barra da treliça. Na Figura 13, a barra localizada na região superior direita torna a treliça instável, pois um dos nós da barra não possui conexão com outras barras ou condições de contorno de força ou deslocamento associadas a ele. Como resultado, a sub-rotina que calcula a estrutura utilizando o método dos elementos finitos não conseguiu determinar uma configuração da estrutura otimizada como todos os elementos estáveis internamente.

**Figura 13** - Treliça otimizada para o problema de 303 barras.



Fonte: Autor.

**Tabela 3** - Peso inicial e final para o problema de 303 barras.

	<b>Comprimento Total (m)</b>	<b>Área dos Elementos (m<sup>2</sup>)</b>	<b>Volume (m<sup>3</sup>)</b>	<b>Peso Específico (kgf/m<sup>3</sup>)</b>	<b>Peso Total (kgf)</b>	<b>Dif. (%)</b>
<b>Treliça Inicial</b>	68,06	0,000123	0,008371	7850	65,71	<b>16,7</b>
<b>Treliça Otimizada</b>	56,68	0,000123	0,006971	7850	54,72	

Fonte: Autor.

## 6 CONCLUSÃO

A otimização de treliças bidimensionais por meio de algoritmos computacionais representa um campo de estudo promissor para a engenharia estrutural. Neste trabalho foi desenvolvido um algoritmo de otimização topológica que se mostrou eficiente na determinação da topologia ótima das treliças, com base no critério de remoção de barras com tensão abaixo de um limite predefinido. Os resultados obtidos demonstraram a capacidade do algoritmo em reduzir o peso das estruturas otimizadas, apresentando uma redução no peso da estrutura em até 25%. Essa redução representa não apenas uma economia significativa de materiais, mas também uma melhoria no desempenho global das treliças, resultando em estruturas mais leves e eficientes.

Destaca-se que as assimetrias observadas na distribuição de elementos são resultado da lógica de execução do algoritmo de otimização. A assimetria pode ser atribuída ao recálculo da estrutura após a remoção de cada elemento, influenciando os demais elementos removidos, especialmente em treliças de malha densa. Além disso, a treliça com malha de alta densidade revelou uma inconsistência em uma barra que tornou a treliça instável internamente devido à falta de conexão de um dos nós da barra com outras barras ou condições de contorno. Isso impediu a determinação de uma configuração otimizada estável internamente pela sub-rotina de cálculo utilizando o método dos elementos finitos. A instabilidade interna pode ser relacionada à ausência de uma instrução específica que evite que um nó conectado a uma barra permaneça desconectado de outras barras, ou às condições de contorno de força e deslocamento.

Como sugestão para futuros trabalhos, recomenda-se a incorporação de outros critérios de otimização, tais como a carga crítica de flambagem e deformação máxima da estrutura. A inclusão desses critérios permitiria uma análise mais abrangente do comportamento estrutural, levando em consideração aspectos adicionais de segurança e desempenho. Além disso, outra oportunidade de pesquisa consistiria na implementação do algoritmo de otimização considerando a não-linearidade geométrica da estrutura. A inclusão desse aspecto proporcionaria uma análise mais precisa, especialmente em situações onde as deformações são significativas e apresentam uma influência maior no comportamento estrutural.

## REFERÊNCIAS BIBLIOGRÁFICAS

- ARGYRIS, J. H.; KELSEY, S. **Energy Theorems and Structural Analysis**. London: Butterworths, 1960
- BENDSØE, M. P.; SIGMUND, O. **Topology optimization: theory, methods, and applications**. 2. ed., corrected printing ed. Berlin Heidelberg: Springer, 2004.
- COURANT, R. **Variational Methods for the Solution of Problems of Equilibrium and Vibration**. Bulletin of the American Mathematical Society, 1943.
- COUTINHO, K. D. **Método de Otimização Topológica em Estruturas Tridimensionais**. Dissertação (Mestrado)—Natal: Universidade Federal do Rio Grande do Norte, 2006.
- GOLECKI, T. et al. Bridge topology optimization considering stochastic moving traffic. **Engineering Structures**, v. 292, p. 116498, out. 2023.
- HIBBELER, R. C. **Análise das estruturas**. São Paulo: Pearson Education do Brasil, 2013.
- LIU, Y. et al. Assessing the CO2 reduction target gap and sustainability for bridges in China by 2040. **Renewable and Sustainable Energy Reviews**, v. 154, p. 111811, fev. 2022.
- PAULINO, D. M. S. **Otimização topológica de estruturas planas considerando comportamento não linear geométrico**. Mestrado em Estruturas—São Carlos: Universidade de São Paulo, 11 set. 2019.
- SANT'ANNA, H. M. **Otimização Topológica De Estruturas Bidimensionais Contínuas Submetidas A Restrições De Flexibilidade E Tensão**. Porto Alegre: Universidade Federal Do Rio Grande Do Sul, 2002.
- SIGMUND, O. A 99 line topology optimization code written in Matlab. **Structural and Multidisciplinary Optimization**, v. 21, n. 2, p. 120–127, abr. 2001.
- TURNER, M. J.; CLOUGH, R. W.; MARTIN, H. C.; Topp L. J. **Stiffness and Deflection Analysis of Complex Structures**. Journal of Aeronautical Sciences, 1956.
- VAZ, L. E. **Método dos elementos finitos em análise de estruturas**. Rio de Janeiro: Elsevier, 2011.
- ZIENKIEWICZ, O. C. **The Finite Element Method in Engineering Science**. 2. ed. McGraw Hill, 1971.

ZUO, T.; HAN, H.; LIU, Z. Explicit Tunnels and Cavities Control Using SIMP and MMA in Structural Topology Optimization. **Computer-Aided Design**, v. 158, p. 103482, maio 2023.

## APÊNDICE I – CÓDIGO DE ELEMENTOS FINITOS DE TRELIÇA

```
program truss_2D
```

```
implicit none
```

```
!Bloco de declaração de variáveis
```

```
integer(4)::i,j,k,m,n,numnos,numele,numpro,numdes,numfor,rowl,coll,rowg,colg
```

```
integer(4)::idele_read,noi_read,noj_read,proele_read
```

```
integer(4),allocatable::idnos(:),idpro(:),idele(:),noi(:),noj(:),proele(:)
```

```
integer(4),allocatable::idnod(:),iddx(:),iddy(:),idgd(:)
```

```
integer(4),allocatable::idnof(:),idfx(:),idfy(:)
```

```
real(8),allocatable::crdx(:),crdy(:),Asec(:),Emat(:),desx(:),desy(:),Dg(:),forx(:),fory(:),Fg(:)
```

```
real(8),allocatable::comp(:),lbdx(:),lbdy(:),coef(:),kloc(:,,:),Kg(:,,:),Kgcc(:,,:),DI(:,,:),kTj(:,,:)
```

```
real(8),allocatable::sigma(:), sigma_yield(:)
```

```
!Arquivos de entrada e saída
```

```
open (unit=2, file='input.txt')
```

```
open (unit=6, file='output.txt')
```

```
open (unit=8, file='outputacadview.txt')
```

```
!i,j,k,m,n: contadores
```

```
!numnos: número de nos da estrutura
```

```
!numele: número de elementos da estrutura
```

```
!numpro: número de propriedade das barras de treliça
```

```
!numdes: número de nós com deslocamentos conhecidos
```

```
!numfor: número de nós com forças conhecidas
```



!ldgd: identificador dos graus de liberdade da estrutura com deslocamento conhecido

!Bloco de entrada de dados

! Abrir o arquivo de entrada

read(2,\*)!numero de nós da treliça 2D (numn)

read(2,\*)numnos

read(2,\*)!numero de elemento da treliça 2D (nume)

read(2,\*)numele

read(2,\*)!numero de propriedades geométricas: área da seção (Asec) e modulo de elasticidade do material (Emat)

read(2,\*)numpro

read(2,\*)!numero de nós com deslocamento prescrito

read(2,\*)numdes

read(2,\*)!Numero de nós com força prescrita

read(2,\*)numfor

read(2,\*)!Cordenadas dos nós da treliça (no,crdx,crdy)

allocate(idnos(numnos),crdx(numnos),crdy(numnos))

idnos=0;crdx=0.0;crdy=0.0

do i=1,numnos

    read(2,\*) idnos(i), crdx(i), crdy(i)

enddo

read(2,\*)!Propriedades do elemento (ldpro, Asec, Emat)

allocate(ldpro(numpro),Asec(numpro),Emat(numpro), sigma\_yield(numpro))

ldpro=0;Asec=0.0;Emat=0.0;sigma\_yield=0.0

do i=1,numpro

    read(2,\*)ldpro(i),Asec(i),Emat(i),sigma\_yield(i)

enddo

read(2,\*)!Incidencia cinemática (idele, no i, no j, proele )

allocate(idele(numele),noi(numele),noj(numele),proele(numele))

idele=0;noi=0;noj=0;proele=0

do i=1,numele

```
! Extrair os valores da linha
read(2,*) idele_read, noi_read, noj_read, proele_read

! Armazenar os valores nos arrays
idele(i) = idele_read
noi(i) = noi_read
noj(i) = noj_read
proele(i) = proele_read
enddo
open (unit=10, file='teste.txt')
write(10,*) idele
close(unit=10)
read(2,*)!Aplicação da condição de contorno em deslocamento (idnod, iddx,
iddy, desx, desy)

allocate(idnod(numdes),iddx(numdes),iddy(numdes),desx(numdes),desy(numdes))
idnod=0;iddx=0;iddy=0;desx=0.0;desy=0.0
do i=1,numdes
    read(2,*)idnod(i),iddx(i),iddy(i),desx(i),desy(i)
enddo
read(2,*)!Aplicação da condição de contorno em força (idnof, idfx, idfy, forx,
fory)

allocate(idnof(numfor),idfx(numfor),idfy(numfor),forx(numfor),fory(numfor))
idnof=0;idfx=0;idfy=0;forx=0.0;fory=0.0
do i=1,numfor
    read(2,*)idnof(i),idfx(i),idfy(i),forx(i),fory(i)
enddo
close(unit=2)

!Bloco de cálculo e processamento de dados

!Cálculo do comprimento dos elementos
allocate(comp(numele));comp=0.0
```

```
do i=1,numele
  comp(idele(i))=((crdx(noj(idele(i)))-crdx(noi(idele(i))))**2.0&
  +(crdy(noj(idele(i)))-crdy(noi(idele(i))))**2.0)**0.5
enddo
```

!Cálculo dos cossenos diretores dos elementos

```
allocate(lbdx(numele),lbdy(numele));lbdx=0.0;lbdy=0.0
```

```
do i=1,numele
```

```
  lbdx(idele(i))=(crdx(noj(idele(i)))-crdx(noi(idele(i))))/comp(idele(i))
```

```
  lbdy(idele(i))=(crdy(noj(idele(i)))-crdy(noi(idele(i))))/comp(idele(i))
```

```
enddo
```

!Cálculo das matriz de rigidez dos elementos

```
allocate(kloc(numele,4,4),coef(numele));kloc=0.0;coef=0.0
```

```
do i=1,numele
```

```
  coef(idele(i))=(Asec(proele(idele(i)))*Emat(proele(idele(i))))/comp(idele(i))
```

```
  kloc(i,1,1)=coef(idele(i))*lbdx(idele(i))*lbdx(idele(i))
```

```
  kloc(i,1,2)=coef(idele(i))*lbdx(idele(i))*lbdy(idele(i))
```

```
  kloc(i,1,3)=-kloc(i,1,1)
```

```
  kloc(i,1,4)=-kloc(i,1,2)
```

```
  kloc(i,2,1)=kloc(i,1,2)
```

```
  kloc(i,2,2)=coef(idele(i))*lbdy(idele(i))*lbdy(idele(i))
```

```
  kloc(i,2,3)=-kloc(i,1,2)
```

```
  kloc(i,2,4)=-kloc(i,2,2)
```

```
  kloc(i,3,1)=-kloc(i,1,1)
```

```
  kloc(i,3,2)=-kloc(i,1,2)
```

```
  kloc(i,3,3)=kloc(i,1,1)
```

```
  kloc(i,3,4)=kloc(i,1,2)
```

```
  kloc(i,4,1)=-kloc(i,1,2)
```

```
  kloc(i,4,2)=-kloc(i,2,2)
```

```
  kloc(i,4,3)=kloc(i,1,2)
```

```
  kloc(i,4,4)=kloc(i,2,2)
```

```
enddo
```

!Cálculo da incidência cinemática

```
allocate(Kg(2*numnos,2*numnos));Kg=0.0
```

```
allocate(Kgcc(2*numnos,2*numnos));Kgcc=0.0
```

```
do i=1,numele
```

```
  do j=1,2 !número de nós do elemento
```

```
    do k=1,2 !número de graus de liberdade
```

```
      rowl=2*(j-1)+k
```

```
      if (j==1) rowg=2*(noi(idele(i))-1)+k
```

```
      if (j==2) rowg=2*(noj(idele(i))-1)+k
```

```
      do m=1,2
```

```
        do n=1,2
```

```
          coll=2*(m-1)+n
```

```
          if (m==1) colg=2*(noi(idele(i))-1)+n
```

```
          if (m==2) colg=2*(noj(idele(i))-1)+n
```

```
          Kg(rowg,colg)=Kg(rowg,colg)+kloc(i,rowl,coll)
```

```
          Kgcc(rowg,colg)=Kgcc(rowg,colg)+kloc(i,rowl,coll)
```

```
        enddo
```

```
      enddo
```

```
    enddo
```

```
  enddo
```

```
enddo
```

!Aplicação das condições de contorno em deslocamentos conhecidos

```
allocate(idgd(2*numnos));idgd=0
```

```
do i=1,numdes
```

```
  if(iddx(i)==1) idgd(2*idnod(i)-1)=1
```

```
  if(iddy(i)==1) idgd(2*idnod(i)-0)=1
```

```
enddo
```

!Técnica da diagonal unitária, zerando linhas e colunas correspondentes

```
do i=1,2*numnos
```

```
  if(idgd(i)==1) then
```

$Kg_{cc}(:,i)=0.0$  !zerando as linhas da matriz de rigidez com o deslocamento do grau de liberdade conhecido

$Kg_{cc}(i,:)=0.0$  !zerando as colunas da matriz de rigidez com o deslocamento do grau de liberdade conhecido

$Kg_{cc}(i,i)=1.0$  !inserindo a unidade na diagonal da matriz de rigidez com o deslocamento do grau de liberdade conhecido

endif

enddo

!Determinação de um vetor deslocamento global

allocate(Dg(2\*numnos));Dg=0.0

do i=1,numnos

do j=1,numdes

if(idnos(i)==idnod(j))then

do k=1,2

if(k==1.and.iddx(j)==1) Dg(2\*(idnos(i)-1)+k)=desx(j)

if(k==2.and.iddy(j)==1) Dg(2\*(idnos(i)-1)+k)=desy(j)

enddo

endif

enddo

enddo

!Bloco de aplicação das condições de contorno em força conhecidos

allocate(Fg(2\*numnos));Fg=0.0

do i=1,numnos

do j=1,numfor

if(idnos(i)==idnof(j))then

do k=1,2

if(k==1.and.idfx(j)==1) Fg(2\*(idnos(i)-1)+k)=forx(j)

if(k==2.and.idfy(j)==1) Fg(2\*(idnos(i)-1)+k)=fory(j)

enddo

endif

enddo

```
enddo
```

!Cálculo do sistema de equações por meio da rotina Gauss-Jordan  
(numerical recipes)

```
call gaussj (Kgcc,2*numnos,2*numnos,Fg,1,1)
```

```
Dg=Fg
```

!Cálculo das reações de apoio

```
Fg=0.0
```

```
do i=1,2*numnos
```

```
do j=1,2*numnos
```

```
Fg(i)=Fg(i)+Kg(i,j)*Dg(j)
```

```
end do
```

```
end do
```

!Passagem dos deslocamentos globais para locais

```
allocate(Dl(numele,4));Dl=0.0
```

```
do i=1,numele
```

```
do j=1,2
```

```
do k=1,2
```

```
rowl=2*(j-1)+k
```

```
if (j==1) rowg=2*(noi(idele(i))-1)+k
```

```
if (j==2) rowg=2*(noj(idele(i))-1)+k
```

```
Dl(i,rowl)=Dg(rowg)
```

```
end do
```

```
end do
```

```
end do
```

!Cálculo do produto entre a matriz de rigidez local e a matriz de  
transformação pro no final

```
allocate(kTj(numele,4));kTj=0.0
```

```
Do i=1,numele
```

```
kTj(i,1)=-
```

```
lbdx(idele(i))*(Asec(proele(idele(i)))*Emat(proele(idele(i))))/comp(idele(i))
```

```
      kTj(i,2)=-  
lbdy(idele(i))*(Asec(proele(idele(i)))*Emat(proele(idele(i))))/comp(idele(i))  
  
kTj(i,3)=lbdx(idele(i))*(Asec(proele(idele(i)))*Emat(proele(idele(i))))/comp(idele(i))  
  
kTj(i,4)=lbdy(idele(i))*(Asec(proele(idele(i)))*Emat(proele(idele(i))))/comp(idele(i))  
    end do  
    !Cálculo dos esforços nos elementos de treliça  
    allocate(sigma(numele)); sigma=0.0  
    do i=1,numele  
        do j=1,4  
            sigma(idele(i))=sigma(idele(i))+kTj(i,j)*DI(i,j)  
        end do  
    end do  
  
    !Bloco de saída de dados  
    write(6,*)"numero de nós da treliça 2D:"  
    write(6,*)numnos  
    write(6,*)"numero de elemento da treliça 2D:"  
    write(6,*)numele  
    write(6,*)"numero de propriedades geométricas e de materiais"  
    write(6,*)numpro  
    write(6,*)"numero de nós com deslocamento prescrito"  
    write(6,*)numdes  
    write(6,*)"Numero de nós com força prescrita"  
    write(6,*)numfor  
    write(6,*)  
    write(6,*)"Cordenadas dos nós da treliça (no,coordenada x,coordenada y)"  
    do i=1,numnos  
        write(6,*)idnos(i),crdx(idnos(i)),crdy(idnos(i))  
    enddo  
    write(6,*)  
    write(6,*)"Propriedades do elemento (ldpro, Asec, Emat)"
```

```
do i=1,numpro
    write(6,*)ldpro(i),Asec(i),Emat(i),sigma_yield(i)
enddo
write(6,*)
write(6,*)"Incidencia cinemática (idele, no i, no j, proele )"
do i=1,numele
    write(6,*)idele(i),noi(idele(i)),noj(idele(i)),proele(idele(i))
enddo
write(6,*)
write(6,*)"Aplicação da condição de contorno em deslocamento (idnod, iddx,
iddy, desx, desy)"
do i=1,numdes
    write(6,*)idnod(i),iddx(i),iddy(i),desx(i),desy(i)
enddo
write(6,*)
write(6,*)"Aplicação da condição de contorno em força (idnof, idfx, idfy, forx,
fory)"
do i=1,numfor
    write(6,*)idnof(i),idfx(i),idfy(i),forx(i),fory(i)
enddo
write(6,*)
write(6,*)
write(6,*)"comprimento dos elementos de treliça"
do i=1,numele
    write(6,*)comp(idele(i))
enddo
write(6,*)
write(6,*)"cossenos diretores dos elementos de treliça"
write(6,*)"lambda x, lambda y"
do i=1,numele
    write(6,*)lbdx(idele(i)),lbdy(idele(i))
enddo
write(6,*)
```



```
write(6,*)"matriz de rigidez do elemento de treliça"  
write(6,*)  
do i=1,numele  
  write(6,*)"elemento:",idele(i)  
  do j=1,4  
    write(6,*)(kloc(i,j,k), k=1,4)  
  enddo  
  write(6,*)  
enddo  
write(6,*)  
write(6,*)"matriz de rigidez da estrutura"  
write(6,*)  
do i=1,2*numnos  
  write(6,*)(Kg(i,j), j=1,2*numnos)  
enddo  
write(6,*)  
write(6,*)"matriz de flexibilidade da estrutura com condições de contorno"  
write(6,*)  
do i=1,2*numnos  
  write(6,*)(Kgcc(i,j), j=1,2*numnos)  
enddo  
write(6,*)  
write(6,*)"Deslocamento dos nós"  
write(6,*)  
do i=1,2*numnos  
  write(6,*)i,Dg(i)  
enddo  
write(6,*)  
write(6,*)"Forças externas e reações de apoio"  
write(6,*)  
do i=1,2*numnos  
  write(6,*)i,Fg(i)  
enddo
```

```
write(6,*)
write(6,*)"Tensões nas barras da treliça "
write(6,*)
do i=1,numele
    write(6,*)idele(i),sigma(idele(i))
enddo

!Bloco do arquivo de saída de dados para o acadview
write(8,("Acadview"))
write(8,*)
write(8,("Number of nodes, number of elements, number of lists"))
write(8,("#"))
write(8,*)numnos,numele,2
write(8,*)
write(8,("x,y,z,dx,dy,dz"))
write(8,("#"))
do i=1,numnos
    write(8,(6(f10.5)))crdx(idnos(i)),crdy(idnos(i)),0.0,0.0,0.0,0.0
end do
write(8,*)
write(8,("Truss element, degree of approximation, nodes of element"))
write(8,("#"))
do i=1,numele
    write(8,(10(i5)))1,1,noi(idele(i)),noj(idele(i))
end do

!Acadiview - List of displacements
write(8,*)
write(8,("List of displacements"))
write(8,*)
write(8,("#"))
write(8,("Deslocamento em x"))
do i=1,numnos
```

```

        write(8,'(4(es20.10))')Dg(2*idnos(i)-1),Dg(2*idnos(i)-
0),0.0,Dg(2*idnos(i)-1)
        end do
        write (8,*)
        write(8,'("#"')
        write(8,'("Deslocamento em y"')
        do i=1,numnos
            write(8,'(4(es20.10))')Dg(2*idnos(i)-1),Dg(2*idnos(i)-
0),0.0,Dg(2*idnos(i)-0)
        end do
        close(unit=6)
        close(unit=8)
    end program truss_2D
    
```

SUBROUTINE gaussj(a,n,np,b,m,mp)

INTEGER(4) m,mp,n,np,NMAX

REAL(8) a(np,np),b(np,mp)

PARAMETER (NMAX=20000)

!Linear equation solution by Gauss-Jordan elimination, equation (2.1.1) above.  $a(1:n,1:n)$  is an input matrix stored in an array of physical dimensions  $np$  by  $np$ .  $b(1:n,1:m)$  is an input matrix containing the right-hand side vectors, stored in an array of physical dimensions  $np$  by  $mp$ .

!On output,  $a(1:n,1:n)$  is replaced by its matrix inverse, and  $b(1:n,1:m)$  is replaced by the corresponding set of solution vectors.

!Parameter: NMAX is the largest anticipated value of  $n$ .

INTEGER i,icol,irow,j,k,l,ll,indx(NMAX),indxr(NMAX),ipiv(NMAX) !The integer arrays ipiv, indxr, and indx are used for bookkeeping on the pivoting.

REAL big,dum,pivinv

do j=1,n

ipiv(j)=0

end do

do i=1,n !This is the main loop over the columns to be reduced.

big=0.0

```
do j=1,n          !This is the outer loop of the search for a pivot element.
if(ipiv(j).ne.1)then
do k=1,n
if (ipiv(k).eq.0) then
if (abs(a(j,k)).ge.big)then
big=abs(a(j,k))
irow=j
icol=k
endif
endif
enddo
endif
enddo
ipiv(icol)=ipiv(icol)+1    !We now have the pivot element, so we interchange
rows, if needed, to put the pivot element on the diagonal. The columns are not
physically interchanged, only relabele
if (irow.ne.icol) then
do l=1,n
dum=a(irow,l)
a(irow,l)=a(icol,l)
a(icol,l)=dum
enddo
do l=1,m
dum=b(irow,l)
b(irow,l)=b(icol,l)
b(icol,l)=dum
enddo
endif
indxr(i)=irow !We are now ready to divide the pivot row by the pivot element,
located at i row and i col.
indxc(i)=icol
if (a(icol,icol).eq.0.) write(*,*) 'singular matrix in gaussj'
pivinv=1./a(icol,icol)
```

```
a(icol,icol)=1.  
do l=1,n  
a(icol,l)=a(icol,l)*pivinv  
enddo  
do l=1,m  
b(icol,l)=b(icol,l)*pivinv  
enddo  
do ll=1,n          !Next, we reduce the rows...  
if(ll.ne.icol)then !except for the pivot one.  
dum=a(ll,icol)  
a(ll,icol)=0.  
do l=1,n  
a(ll,l)=a(ll,l)-a(icol,l)*dum  
enddo  
do l=1,m  
b(ll,l)=b(ll,l)-b(icol,l)*dum  
enddo  
endif  
enddo
```

enddo ! This is the end of the main loop over columns of the reduction.

do l=n,1,-1 !It only remains to unscramble the solution in view of the column interchanges. We do this by interchanging pairs of columns in the reverse order that the permutation was built up.

```
if(indxr(l).ne.indxc(l))then  
do k=1,n  
dum=a(k,indxr(l))  
a(k,indxr(l))=a(k,indxc(l))  
a(k,indxc(l))=dum  
enddo  
endif  
enddo  
return  
END
```