

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ**

**DIEGO APARECIDO DE SOUZA**

**IMPLEMENTAÇÃO DE CÓDIGO COMPUTACIONAL PARA ANÁLISE ESTÁTICA  
DE TRELIÇAS PLANAS VIA MÉTODO DOS ELEMENTOS FINITOS**

**CAMPO MOURÃO**

**2022**

**DIEGO APARECIDO DE SOUZA**

**IMPLEMENTAÇÃO DE CÓDIGO COMPUTACIONAL PARA ANÁLISE ESTÁTICA  
DE TRELIÇAS PLANAS VIA MÉTODO DOS ELEMENTOS FINITOS**

**Implementation of computational code for static analysis of plane trusses via finite  
element method**

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Engenharia Civil do Departamento Acadêmico de Construção Civil – DACOC - da Universidade Tecnológica Federal do Paraná (UTFPR).  
Orientador: Prof. Dr. Ronaldo Rigobello.  
Coorientador: Prof. Dr. Marcelo Rodrigo Carreira.

**CAMPO MOURÃO**

**2022**



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

**DIEGO APARECIDO DE SOUZA**

**IMPLEMENTAÇÃO DE CÓDIGO COMPUTACIONAL PARA ANÁLISE ESTÁTICA  
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 do Departamento Acadêmico de Construção Civil – DACOC – da Universidade Tecnológica Federal do Paraná (UTFPR).

Data de aprovação: 08/junho/2022

---

Ronaldo Rigobello  
Doutorado  
Universidade Tecnológica Federal do Paraná

---

Leandro Waidemam  
Doutorado  
Universidade Tecnológica Federal do Paraná

---

Angelo Giovanni Bonfim Corelhano  
Mestrado  
Universidade Tecnológica Federal do Paraná

**CAMPO MOURÃO**

**2022**

Dedico este trabalho à minha família, por me proporcionar  
esta conquista.

## **AGRADECIMENTOS**

A Deus, pela minha vida, e por me ajudar a ultrapassar todos os obstáculos encontrados ao longo do curso.

Aos meus pais e irmãos que me incentivaram nos momentos mais difíceis e compreenderam minha ausência enquanto me dedicava à conquista desse título.

Aos amigos e colegas que contribuíram direta ou indiretamente para a realização deste trabalho e tornaram esses anos mais felizes.

Aos professores por todo conhecimento passado e pela paciência demonstrada, me possibilitando um melhor desempenho no processo de formação acadêmica.

## RESUMO

A treliça é um dos sistemas estruturais mais usados na engenharia. Os esforços solicitantes devido às ações aplicadas a uma treliça podem ser obtidos por processos analíticos clássicos, como o método dos esforços e o método dos deslocamentos. No entanto, dependendo da quantidade de elementos e seu grau de hiperestaticidade, o cálculo dos esforços e deslocamento dessas estruturas de forma manual pode demorar demasiadamente. Com isso, é de fundamental importância que o profissional de engenharia disponha de uma ferramenta computacional, como forma de reduzir tempo e erros na etapa de análise estrutural. O objetivo do presente trabalho foi implementar e validar um código de computador para solução de treliças planas via método dos elementos finitos. O código foi escrito em linguagem Scilab considerando o contexto de material elástico linear e teoria de pequenos deslocamentos. Foram resolvidos quatro exemplos com o código desenvolvido e os resultados obtidos foram comparados com as soluções encontradas nos softwares Ftool e Risa 2D. Os resultados obtidos com o código desenvolvido foram considerados válidos, pois as diferenças obtidas nos resultados foram inferiores a 0,2% em todos os casos.

Palavras-chave: análise estrutural; treliças planas; método dos elementos finitos.

## ABSTRACT

Truss is one of the most used structural systems in Engineering. Solving for the axial forces due to the actions applied to a truss can be done by classical analytical processes, such as the force method and the displacement method. However, depending on the number of elements and their degree of hyperstaticity, these manual processes can last for hours. Thus, it is of fundamental importance that the engineering professional has a computational tool, as a way to reduce time and errors in the structural analysis. The objective of the present work was to implement a computer program for solving plane trusses using the Finite Element Method. The program was written in Scilab language considering the context of linear elastic material and small displacement theory. Four examples were solved with the code developed and the results were compared with the solutions obtained with the Ftool and Risa 2D softwares. The results obtained by the developed computer code were considered valid, since the numerical from the results were less than 0,2% in all cases.

Keywords: structural analysis; plane trusses; finite element method.

## SUMÁRIO

1	<b>INTRODUÇÃO</b> .....	8
2	<b>OBJETIVOS</b> .....	10
2.1	Objetivo geral.....	10
2.2	Objetivos específicos.....	10
3	<b>JUSTIFICATIVA</b> .....	11
4	<b>DESENVOLVIMENTO</b> .....	12
4.1	Treliças.....	12
4.2	Método dos Elementos Finitos .....	16
4.3	Método da rigidez direta aplicado à análise de treliças planas .....	17
5	<b>MATERIAIS E MÉTODOS</b> .....	20
5.1	Implementação computacional.....	20
5.2	Validação do código .....	22
6	<b>RESULTADOS E DISCUSSÕES</b> .....	23
6.1	Implementação computacional.....	23
6.2	Validação dos resultados .....	23
6.2.1	Exemplo 1 .....	23
6.2.2	Exemplo 2.....	25
6.2.3	Exemplo 3.....	28
6.2.4	Exemplo 4.....	30
7	<b>CONCLUSÃO</b> .....	33
	<b>REFERÊNCIAS</b> .....	34
	<b>APÊNDICE A - Código Computacional Desenvolvido</b> .....	35



## 1 INTRODUÇÃO

Amplamente utilizadas em pontes, viadutos, torres, galpões, barracões, palcos, lajes e telhados, as treliças são soluções estruturais convenientes para superar grandes vãos com ótimo custo benefício. Elas são capazes de resistir a cargas de forma eficiente, têm peso próprio relativamente baixo e, como consequência, diminuem as cargas nas fundações.

O arquiteto italiano Andrea Palladio (1508-1580) foi o primeiro a utilizar treliças em seus projetos, os quais incluíam descrições detalhadas e desenhos de treliças de madeira bastantes semelhantes aos utilizados nos dias atuais. Após duzentos anos, foram reintroduzidas por Ulric Grubermann. Engenheiros projetistas costumam se preocupar em selecionar uma viga ou uma treliça para vencer determinada abertura. Quando não há nenhum outro fator presente a decisão é calcada no menor custo. Ao selecionar uma treliça geralmente utiliza-se menos material, no entanto, a fabricação e montagem têm custo maior se comparado as vigas. Para vãos mais curtos o custo total das vigas (custo do material, fabricação e montagem) será menor. Porém, à medida que os vãos aumentam o custo das treliças é compensado com a redução do peso total do material utilizado. Outra vantagem é que com a mesma quantidade de material uma treliça pode ter rigidez maior que uma viga com o mesmo vão (MCCORMAC, 2007).

As vigas de alma cheia de concreto armado necessitam de uma altura de aproximadamente dez por cento do seu comprimento. Já as vigas de alma cheia em aço necessitam de uma seção transversal com altura em torno de cinco por cento da do vão. Em ambos os casos pode-se determinar um limite para que tais soluções sejam econômicas frente outras possibilidades estruturais como as treliças.

Por outro lado, as vigas de concreto protendido podem vencer vãos maiores que os citados acima, em contrapartida serão elementos de difícil execução. Quando comparada às estruturas de concreto e de perfis de aço em alma cheia, há um consenso no meio técnico que a vantagem das treliças se dá para vãos acima de quinze metros

Podem ser usadas em pilares e vigas ou até mesmo em lajes. Os materiais utilizados incluem a madeira, ferro e alumínio. Geralmente dispostas em triângulos com diversos formatos diferentes.

Para o dimensionamento de uma treliça, o primeiro passo é a obtenção dos esforços solicitantes frente às ações impostas. Atualmente, a maioria dos programas de análise estrutural emprega o Método dos Elementos Finitos (MEF) para encontrar a resposta das estruturas frente ao carregamento aplicado.

O presente trabalho utiliza o MEF como ferramenta para análise estrutural de casos envolvendo treliças planas. Pode ser utilizado, por exemplo, como forma de conferência dos resultados obtidos em exercícios propostos nas disciplinas de teoria das estruturas. Com auxílio do processamento de um computador tais análises podem se tornar mais rápidas e eficientes quando comparadas aos métodos clássicos de solução manual.

## **2 OBJETIVOS**

### **2.1 Objetivo geral**

Implementar e validar um código de computador para solução de treliças planas via Método dos Elementos Finitos, comparando as respostas obtidas com as fornecidas pelos programas Ftool e Risa 2D.

### **2.2 Objetivos específicos**

Para atender o objetivo geral, alguns objetivos específicos são requeridos, dentre eles:

- Pesquisar em literatura o Método dos Elementos Finitos e aplicar seus conceitos matemáticos em linguagem computacional;
- Definir o modelo a ser calculado, as propriedades dos materiais empregados, e determinar comprimentos, áreas, conectividade e condições de contorno;
- Obter as reações de apoio e os esforços internos dos elementos por meio dos programas Ftool e Risa 2D;
- Elaborar um fluxograma de cálculo de treliças planas pelo MEF;
- Validar o código proposto comparando seus resultados com os programas supracitados.

### 3 JUSTIFICATIVA

Milhões de profissionais da engenharia e cientistas no mundo inteiro utilizam o método dos elementos finitos (MEF) para prever o comportamento estrutural, mecânico, térmico, elétrico e químico de diversos sistemas. Podendo ser aplicado em problemas que abrangem desde análises de forças em uma conexão flangeada até a análise de desempenho aerodinâmico de um carro de fórmula 1.

De forma geral, esse método está cada vez mais presente na vida do engenheiro, devido a possibilidade de se executar várias simulações e testar diversas geometrias de peças para o mesmo problema e analisar qual se adequa melhor à situação. Vale ressaltar que tais análises fazem com que a etapa de projeto tenha maior grau de eficiência, proporcionando economia e desenvolvimento tecnológico das peças acabadas.

Porém, ainda que os resultados obtidos sejam de elevada qualidade, vale ressaltar que tais resultados dependem da apreciação de profissionais qualificados, pois podem ocorrer erros na etapa de inserção dos dados, acarretando na falta de correspondência entre o modelo selecionado e a estrutura analisada, dentre outros erros que podem comprometer a execução do projeto.

Mesmo que essas estruturas possam ser solucionadas manualmente por processos analíticos clássicos como o método dos esforços ou método dos deslocamentos, dependendo da quantidade de barras da estrutura e de seu grau de hiperestaticidade, o cálculo manual pode tornar-se demasiadamente extenso e propenso a erros. Assim sendo, é pertinente o engenheiro dispor de uma ferramenta computacional para a análise dessas estruturas.

Portanto, é de fundamental importância que pesquisas relacionadas a este tema sejam desenvolvidas na graduação, para que os graduados consigam utilizar tais ferramentas de forma otimizada e com menor propensão à erros.

O mercado de trabalho tem uma demanda crescente por projetistas que saibam simular utilizando tal metodologia, e o profissional que domina esse assunto poderá ser inserido mais facilmente nos setores que empregam essa tecnologia.

Do ponto de vista prático, o estudo contribui também para que o desenvolvedor seja capaz de utilizar programas relacionados ao método e conseguir avaliar os resultados com maior eficiência.

Logo, justifica-se a importância do presente trabalho em estudar o MEF e implementar um código computacional em ambiente Scilab.

## 4 DESENVOLVIMENTO

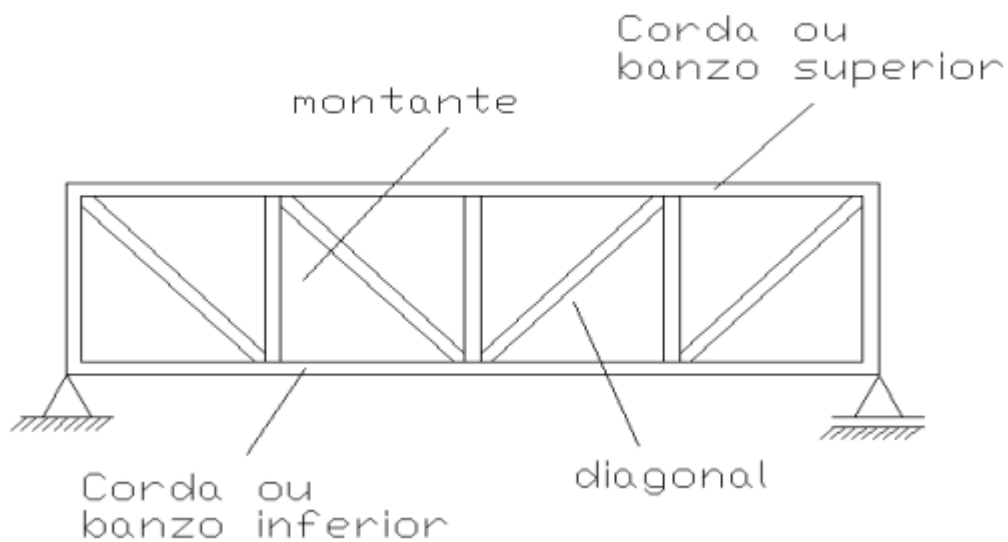
### 4.1 Treliças

Souza e Rodrigues (2008) definem as treliças como estruturas lineares compostas por barras retas. Citam como principais elementos de uma treliça:

- Corda ou banzo são as barras que limitam a treliça no topo e na base;
- Montante são as barras verticais;
- Diagonal são as barras com os eixos coincidentes a diagonal do painel;
- Painel é o trecho entre dois alinhamentos consecutivos de montantes;
- Nó é o ponto onde há o encontro e a junção das extremidades das barras.

Na figura 1 é possível observar uma treliça plana, bem como os elementos que a compõe.

**Figura 1 – Elementos de uma treliça**



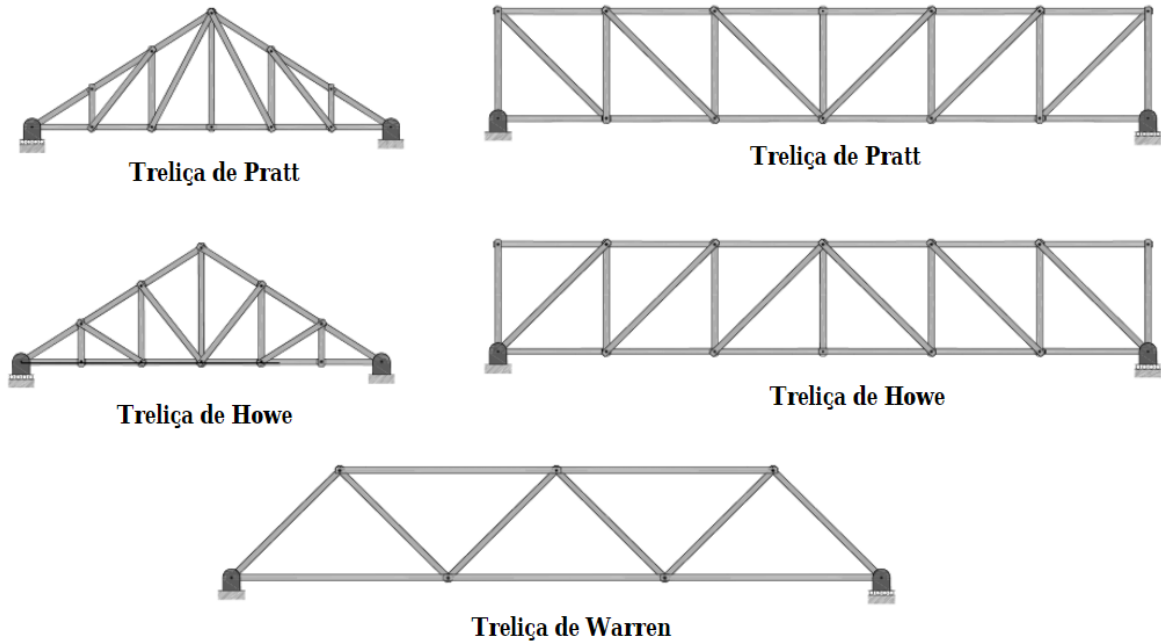
**Fonte: Souza e Rodrigues (2008, p. 29).**

As treliças planas têm a função de suportar cargas que atuam no seu próprio plano, o que garante seu tratamento como estruturas bidimensionais, ou seja, os eixos das barras estão contidos num mesmo plano. Com o carregamento aplicado apenas nos nós, as barras das treliças ficam sujeitas apenas a forças axiais. Sua principal vantagem é conseguir vencer grandes vãos e poder suportar cargas elevadas em relação ao seu peso próprio. Tornando-se uma solução prática e econômica (GOMES, 2016).

Gomes (2016) afirma que as treliças mais utilizadas são as do tipo Warren, Howe e Pratt. Talvez a treliça Warren seja o tipo mais comum para estruturas simples e contínuas, são usadas para vencer vãos entre 50 e 100 metros. A do tipo Pratt é identificada pelos elementos

diagonais, todos inclinados na direção do centro do vão, com exceção dos extremos. Já a treliça Howe é o contrário da treliça Pratt, os elementos diagonais estão na direção oposta ao centro da treliça, como mostrado na figura 2.

**Figura 2 - Elemento finito de treliça plana e coordenadas globais**



Fonte: Gomes (2016, p. 05).

Quanto sua estaticidade, podem ser hipoestáticas, isostáticas e hiperestáticas. Devendo ser analisadas por três óticas, quanto sua estaticidade global, estaticidade interior e estaticidade exterior. O grau de estaticidade global ( $h_g$ ) de uma treliça é dado na equação 1 (GOMES, 2016).

$$h_i = a + b - 2n \quad (1)$$

Onde  $a$  é o número de incógnitas de reação de apoio,  $b$  é o número de barras e  $n$  a quantidade de nós.

Se  $h_g < 0$  a treliça é globalmente hipoestática, se  $h_g = 0$  a treliça será globalmente isostática e se  $h_g > 0$  globalmente hiperestática.

O grau de estaticidade interior ( $h_i$ ) de uma treliça é dado da equação 2.

$$h_i = 3 + b - 2n = b - (2n - 3) \quad (2)$$

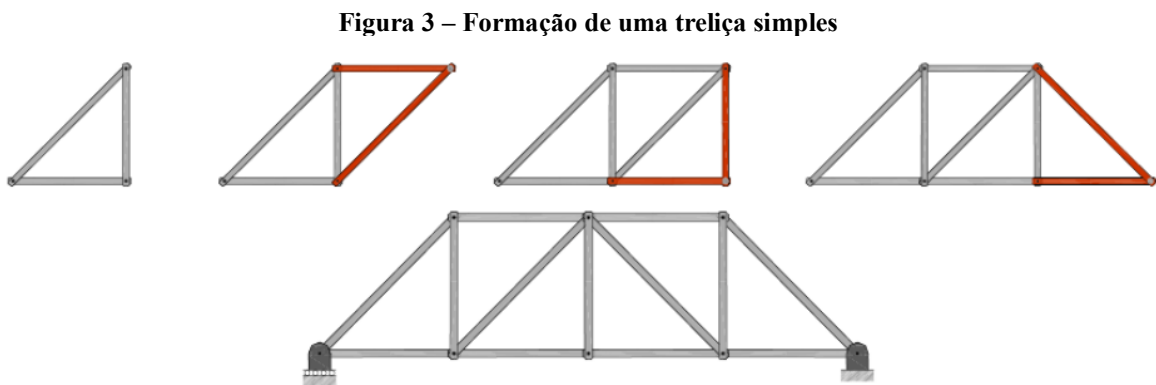
Quando  $h_i < 0$ , há deficiência de barras e a treliça é internamente hipoestática, ao passo que  $h_i = 0$  é uma condição para a estabilidade da treliça, porém a disposição das barras também deve ser considerada e se  $h_i > 0$  existem mais barras que a estrutura necessita para não colapsar.

Já o grau de estabilidade exterior ( $h_e$ ) é dado na equação 3.

$$h_e = a - 3 \quad (3)$$

Se  $h_g < 0$  a treliça é exteriormente hipoestática, se  $h_g = 0$  a treliça será exteriormente isostática e se  $h_g > 0$  exteriormente hiperestática.

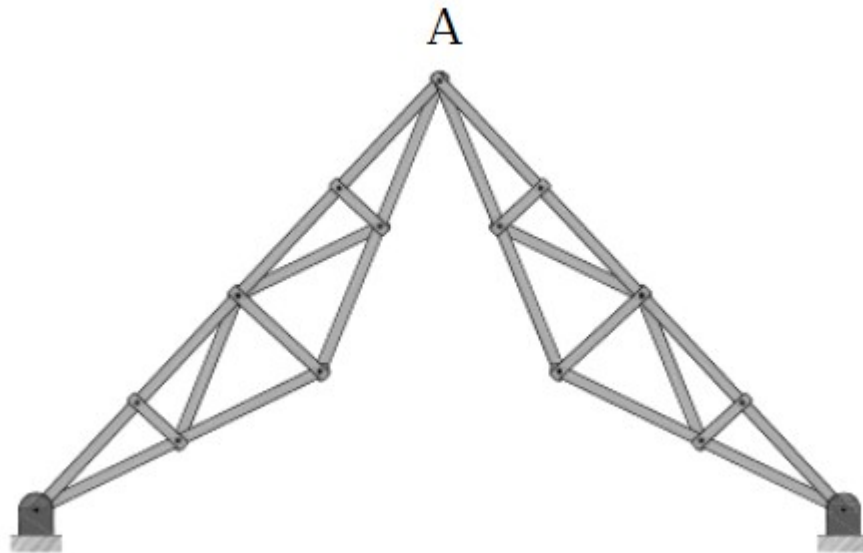
Ainda segundo Gomes (2016), é importante classificar treliças quanto à sua lei de formação, podendo ser definidas como simples, compostas e complexas. A treliça simples é formada a partir de um triângulo inicial, em cada nó é adicionado duas novas barras colineares. Na figura 3 é possível visualizar a formação de uma treliça simples, resultando numa treliça Howe.



**Fonte: Gomes (2016, p. 18).**

A treliça composta é constituída por um triângulo base, acrescentando-se duas barras não colineares em cada novo nó. Existe também a formação de uma treliça composta através da união de duas ou mais treliças simples, na figura 4 é possível visualizar uma treliça composta formada através da junção, no ponto A, de duas treliças simples (GOMES, 2016).

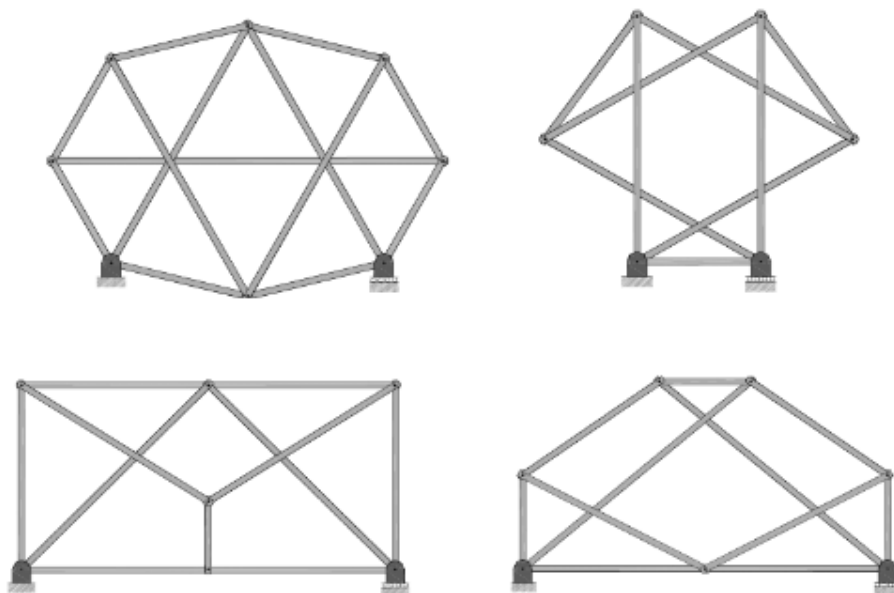
**Figura 4 – Treliça composta**



**Fonte: Gomes (2016, p. 19).**

Treliças complexas podem ser constituídas de combinações de barras em forma de triângulo, quadrilátero ou até mesmo polígonos. Além disso podem apresentar barras que se cruzam sem haver vínculos entre elas, tal situação está evidenciada na figura 5. Sua classificação se dá por exclusão, ou seja, quando a treliça não é simples nem composta (GOMES, 2016).

**Figura 5 – Treliças complexas**



**Fonte: Gomes (2016, p. 21).**



## 4.2 Método dos Elementos Finitos

O desenvolvimento moderno do Método dos Elementos Finitos (MEF) teve início com Hrennikoff em de 1941 no campo na engenharia estrutural. McHenry em 1943 utilizou uma rede de elementos unidimensionais para a solução de tensões em sólidos contínuos. Somente em 1956 o primeiro tratamento dos elementos bidimensionais foi proposto por Turner, derivando matrizes de rigidez para elementos de treliças e vigas, delineando então o processo atualmente denominado como método da rigidez direta (LOGAN, 2015).

Visando atender a necessidade da época, o MEF obteve acelerado desenvolvimento e reconhecimento, com papel fundamental na execução de projetos bem-sucedidos de espaçonaves, mísseis e cápsulas espaciais, estendendo a aplicações em outras áreas da Engenharia (SORIANO, 2009).

Algumas das etapas na formulação da análise dos elementos finitos são comuns em todos os problemas, sejam eles estruturais, de transferência de calor ou de fluxo de fluídos (HUTTON, 2004).

O pré-processamento é descrito como a definição do modelo, nesta etapa são incluídas as propriedades físicas do material que compõe o elemento, os tipos de elementos a serem utilizados (uni, bi e tridimensionais), o domínio geométrico do problema, as propriedades geométricas dos elementos (comprimento e área), as conectividades dos elementos e as restrições físicas (condições de contorno) (HUTTON, 2004).

A etapa de solução inclui a montagem das equações algébricas governantes, em formato de matrizes e cálculo dos valores desconhecidos das variáveis primárias (deslocamentos). Ainda nesta etapa são calculadas respostas adicionais como reações de apoio e esforços internos nos elementos (HUTTON, 2004).

No pós-processamento é realizada a apresentação e análise da solução. Nesta etapa é importante o bom senso por parte do engenheiro para validar se os resultados são fisicamente razoáveis (HUTTON, 2004).

Segundo Logan (2015) existem três principais métodos que podem ser utilizados para derivação das equações presentes nos elementos finitos de um sistema físico, sendo eles: método direto, métodos variacionais e os métodos dos resíduos ponderados. No primeiro existe dois tipos de abordagens, método da força ou método da flexibilidade e método do deslocamento ou método da rigidez, nos quais as forças internas e os deslocamentos nodais são utilizados como incógnitas, respectivamente.

No desenvolvimento deste trabalho foi usado o método dos deslocamentos por apresentar uma formulação mais simples em relação a outros métodos, além de ter uma gama de programas de uso geral que incorporam esta mesma formulação para expressar seus resultados.

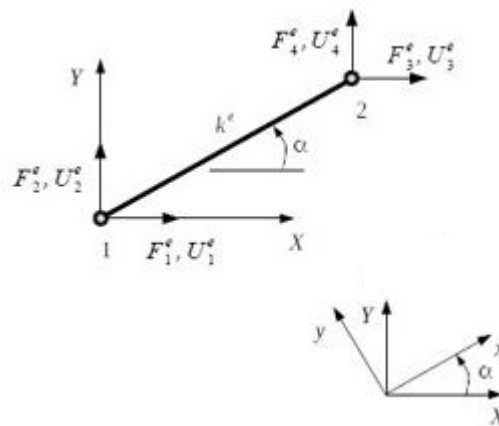
### 4.3 Método da rigidez direta aplicado à análise de treliças planas

Nesta etapa será demonstrado como relacionar forças nodais com seus respectivos deslocamentos, reações de apoio e esforços internos.

O primeiro passo no método da rigidez direta é identificar os elementos e os nós presentes no modelo estrutural, posteriormente é definido um sistema de eixos de referência, sendo eles um local e um global.

Na figura 6 é apresentado um elemento finito de treliça plana na qual  $F^{e_1}$ ,  $F^{e_2}$ ,  $F^{e_3}$  e  $F^{e_4}$  são os vetores de forças e  $U^{e_1}$ ,  $U^{e_2}$ ,  $U^{e_3}$  e  $U^{e_4}$  são os vetores de deslocamentos nodais do elemento segundo o sistema de coordenadas globais.

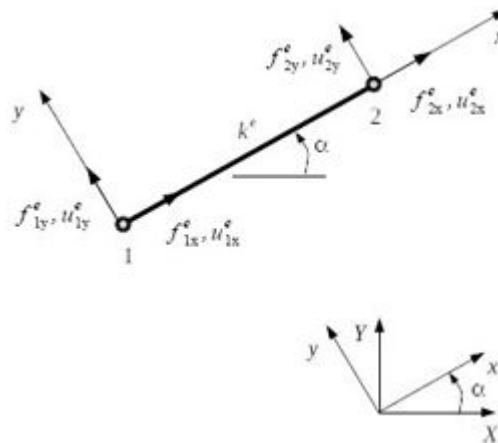
Figura 6 - Elemento finito de treliça plana e coordenadas globais



Fonte: adaptado de Fish e Belytschko (2007, p. 22).

Já na figura 7 é mostrado um elemento finito de treliça plana na qual  $f^{e_{1x}}$ ,  $f^{e_{1y}}$ ,  $f^{e_{2x}}$  e  $f^{e_{2y}}$  são os vetores de forças e  $u^{e_{1x}}$ ,  $u^{e_{1y}}$ ,  $u^{e_{2x}}$  e  $u^{e_{2y}}$  são os vetores de deslocamentos nodais do elemento segundo o sistema de coordenadas locais.

**Figura 7 - Elemento finito de treça plana e coordenadas locais**



Fonte: adaptado de Fish e Belytschko (2007, p. 22).

Primeiramente precisa-se encontrar os deslocamentos nodais  $\{U\}$  resolvendo o sistema da equação (4).

$$\{U\} = [K]^{-1}\{P\} \quad (4)$$

Onde  $[K^{-1}]$  é a matriz inversa de rigidez global e  $\{P\}$  é o vetor de cargas.

Segundo Logan (2015) a matriz de rigidez global  $[K]$  é obtida através da equação (5).

$$[K] = \sum_{e=1}^{nel} [\beta^e]^T [K^e] [\beta^e] \quad (5)$$

Onde  $[\beta^e]$  é a matriz de incidência cinemática. Se trata de uma matriz booleana estritamente constituída por 0 e 1, ela relaciona os deslocamentos globais nas extremidades do elemento com os deslocamentos globais da estrutura. Logan (2015) afirma que matriz global de rigidez do elemento  $[K^e]$  é obtida pela equação (6).

$$[K^e] = [T^e]^T [k^e] [T^e] \quad (6)$$

Onde  $[k^e]$  é a matriz local de rigidez do elemento dada pela equação (7) (LOGAN, 2015).

$$[k^e] = \frac{E^e A^e}{L^e} \begin{pmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad (7)$$

Sendo  $E^e$  o módulo de elasticidade do material,  $A^e$  a área da seção transversal e  $L^e$  o comprimento, ambos do elemento  $e$ .

E  $[T^e]$  é a matriz de transformação de coordenadas mostrada na equação (8) (LOGAN, 2015).

$$[T^e] = \begin{pmatrix} \cos(\alpha) & \sin(\alpha) & 0 & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 & 0 \\ 0 & 0 & \cos(\alpha) & \sin(\alpha) \\ 0 & 0 & -\sin(\alpha) & \cos(\alpha) \end{pmatrix} \quad (8)$$

As reações de apoio  $\{R\}$ , consideradas como forças externas desconhecidas são encontradas resolvendo o sistema contido na equação (9).

$$\{R\} = [K]\{U\} - \{P\} \quad (9)$$

Onde  $\{P\}$  é o vetor de cargas externas conhecidas.

Os esforços nas extremidades dos elementos são calculados pela equação (10).

$$\{f^e\} = [T^e][K^e]\{U^e\} \quad (10)$$

Sendo  $\{f^e\}$  os esforços internos nas extremidades do elemento. Vale ressaltar que sua convenção de sinais é baseada nas coordenadas de esforços e deslocamentos mostradas na figura 7.

## 5 MATERIAIS E MÉTODOS

### 5.1 Implementação computacional

Para a implementação do código de análise de treliças planas pelo MEF, foi necessário realizar uma revisão da literatura, de modo a compreender os fundamentos teóricos do método e convertê-los em linguagem computacional.

Posteriormente foi extraído de Carreira (2019) o fluxograma da Figura 13 para possibilitar o desenvolvimento de um código computacional em linguagem Scilab versão 6.0.2 (64-bit).

Nos dados de entrada é definido qual o modelo que será solucionado. Neles foram abrangidos:

- A quantidade de nós da treliça;
- A quantidade de elementos da estrutura;
- As coordenadas cartesianas de cada nó da treliça;
- A matriz de conectividade das barras relacionando os nós iniciais e finais de cada elemento;
- O vetor de forças nodais;
- As condições de contorno, restringindo alguns movimentos, sem elas o sistema apresenta movimento de corpo rígido impossibilitando o equilíbrio;
- As propriedades dos materiais, como o módulo de elasticidade e a área da seção transversal dos elementos.

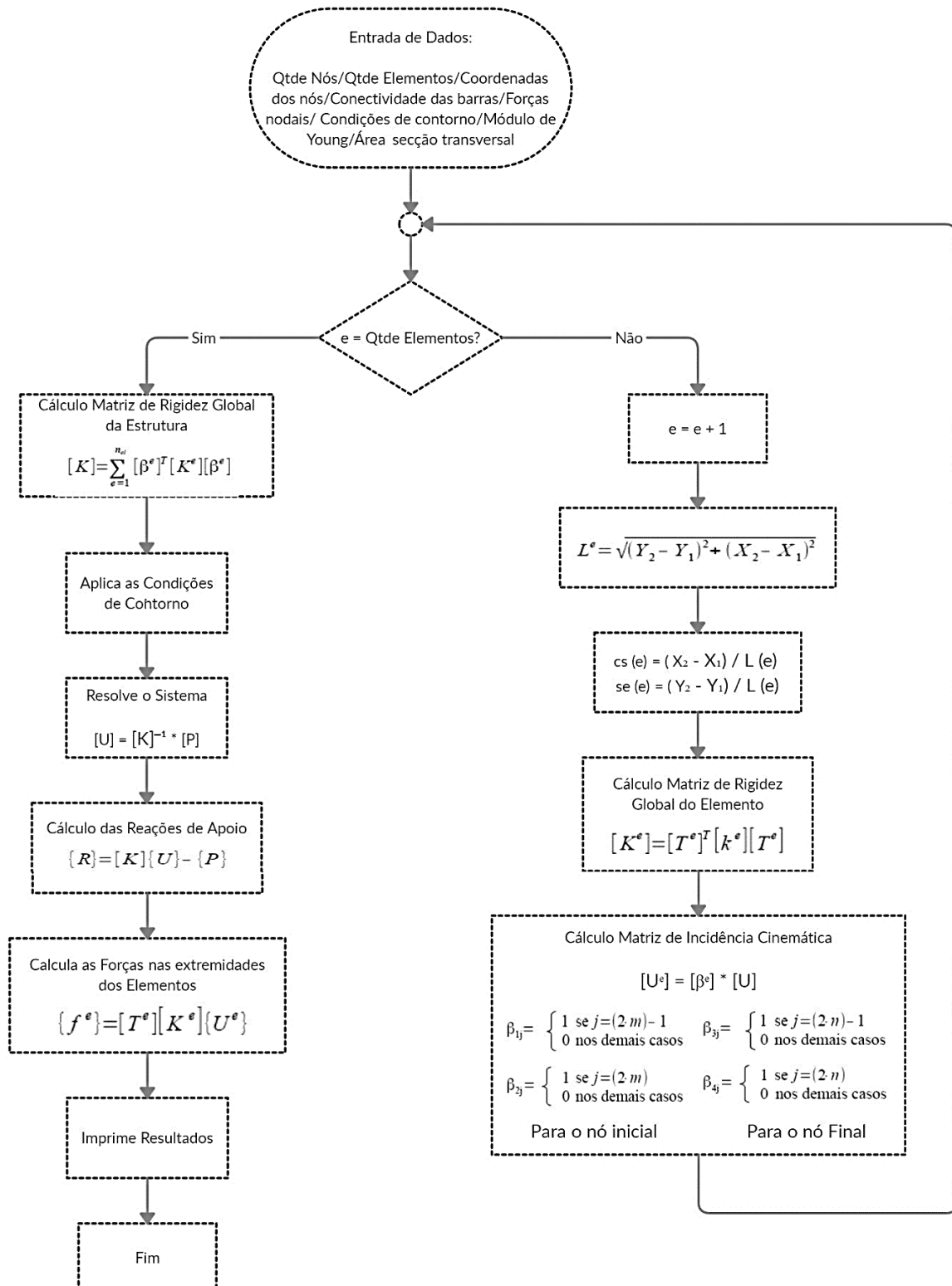
A etapa de entrada de dados é bastante sensível a erros, porque nela os dados são tratados pelo usuário e qualquer divergência entre o problema real e o modelo proposto acarretam em resultados discrepantes.

As unidades de medida definidas na implementação estão de acordo com o Sistema Internacional de Unidades (SI), os modelos com unidades diferentes do SI devem ser convertidos previamente pelo usuário.

Após o processamento dos dados, na etapa de saídas dos resultados, o usuário receberá os deslocamentos nodais, os esforços no elemento e as reações de apoio.

O fluxograma mostrado na figura 8 representa o algoritmo de cálculo de treliças planas pelo método dos elementos finitos.

**Figura 8 – Fluxograma para solução de treliças planas**



Fonte: adaptado de Carreira (2019, p. 43).

## **5.2 Validação do código**

Para validar o código implementado foram processados quatro exemplos de treliças planas. Posteriormente os resultados obtidos pelo código implementado foram comparados aos obtidos pelos programas Ftool, versão 4.00.0 e o Risa 2D, versão educacional 1.0.

## 6 RESULTADOS E DISCUSSÕES

### 6.1 Implementação computacional

O código desenvolvido tem cerca de 245 linhas de código em linguagem Scilab e um tamanho aproximado de 8 Kb. O código com a configuração para resolução do Exemplo 1 (ilustrado pela Figura 13) é disponibilizado no Apêndice A.

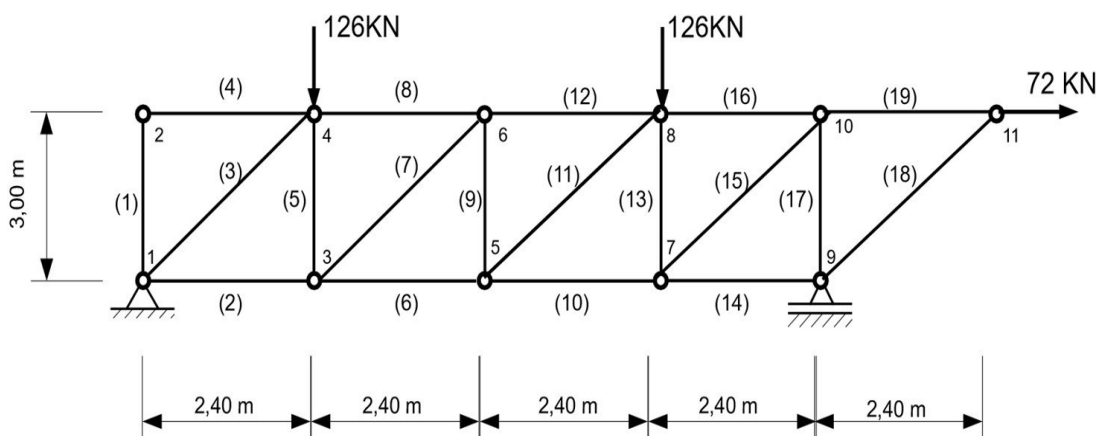
A partir dos dados de entrada, o código com Scilab permite o cálculo dos exemplos aqui proposto em segundos e devolve os resultados. O código implementado soluciona diversos tipos de treliças planas, sendo ela isostática ou hiperestática e não tem restrição quanto ao tamanho da treliça a ser resolvida. A quantidade máxima de nós da treliça é limitada pela memória e processador disponível no computador.

### 6.2 Validação dos resultados

#### 6.2.1 Exemplo 1

O Exemplo 1 cuja geometria, carregamentos e condições de contorno são apresentados na Figura 9, foi extraído de Araujo (2015). Utilizou-se o programa Ftool, versão 4.00.0, para obter os resultados usados para avaliar o software implementado.

**Figura 9 – Exemplo 1: Geometria, ações atuantes e condições de contorno**



Fontes: adaptado de Araujo (2015, p. 42).

As propriedades das barras do Exemplo 1 estão apresentadas na Tabela 1.

**Tabela 1 - Exemplo 1: Área e módulo de elasticidade das barras**

Elementos	Área da seção transversal (cm <sup>2</sup> )	Módulo de Young (GPa)
1 a 19	1	200

Fonte: adaptado de Araujo (2015, p. 43).



Na Tabela 2 são mostrados os deslocamentos nodais obtidos com o Ftool e com o código desenvolvido em Scilab.

**Tabela 2 – Deslocamentos nodais relativos ao exemplo 1**

Nó	Prog, Desenv.		Prog, Ftool		Diferença	
	$u_x$ (mm)	$u_y$ (mm)	$u_x$ (mm)	$u_y$ (mm)	$u_x$ (%)	$u_y$ (%)
1	0	0	0	0	0	0
2	63,340	0	63,340	0	0	0
3	18,570	-79,903	18,576	-79,903	0,032	0
4	63,340	-83,278	63,340	-83,278	0	0
5	34,992	-97,301	34,992	-97,301	0	0
6	53,404	-100,676	53,404	-100,676	0	0
7	49,248	-76,447	49,248	-76,447	0	0
8	45,628	-98,722	45,628	-98,722	0	0
9	49,248	0	49,248	0	0	0
10	40,012	-22,275	40,012	-22,275	0	0
11	48,652	0,477	48,652	0,477	0	0

**Fontes: Autoria própria (2022).**

Pode-se notar que para o exemplo 1 o erro do deslocamento do nó 3 no sentido do eixo x é de 0,032%. Já para o sentido do eixo y os resultados são idênticos.

Na Tabela 3 são disponibilizadas as reações de apoio dos dois programas em questão.

**Tabela 3 – Reações de apoio relativos ao exemplo 1**

Nó	Prog, Desenv.		Prog, Ftool		Diferença	
	$R_x$ (kN)	$R_y$ (kN)	$R_x$ (kN)	$R_y$ (kN)	$R_x$ (%)	$R_y$ (%)
1	-72,000	103,500	-72,000	103,500	0	0
9	0	148,500	0	148,500	0	0

**Fontes: Autoria própria (2022).**

Os resultados obtidos das reações de apoio na tabela 3 não apresentam diferença.

Por último, são apresentados na Tabela 4 os esforços internos.

**Tabela 4 – Esforços internos relativos ao exemplo 1**

<b>Elemento</b>	<b>Prog, Desenv. N (kN)</b>	<b>Prog, Ftool N (kN)</b>	<b>Diferença (%)</b>
1	0	0	0
2	154,800	154,800	0
3	-132,545	-132,545	0
4	0	0	0
5	-22,500	-22,500	0
6	136,800	136,800	0
7	28,814	28,814	0
8	-82,800	-82,800	0
9	-22,500	-22,500	0
10	118,800	118,800	0
11	28,814	28,814	0
12	-64,800	-64,800	0
13	-148,500	-148,500	0
14	0	0	0
15	190,173	190,173	0
16	-46,800	-46,800	0
17	-148,500	-148,500	0
18	0	0	0
19	72,000	72,000	0

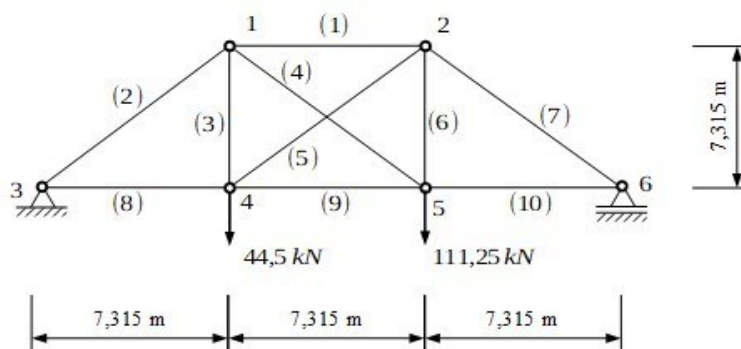
**Fontes: Autoria própria (2022).**

Os esforços normais apresentados na tabela 4 e obtidos com o código desenvolvido são idênticos aos resultados obtidos com o Ftool.

### 6.2.2 Exemplo 2

O exemplo 2, foi extraído de McCormac (2007). O programa Ftool apresenta limitações na obtenção dos resultados de treliças complexas, pois é adicionado automaticamente um nó no encontro das barras (4) e (5), alterando o modelo estrutural proposto na figura 10. Devido tal limitação, os resultados para fins de comparação foram obtidos através do programa Risa 2D, versão educacional 1.0.

**Figura 10 – Exemplo 2: Geometria, ações atuantes e condições de contorno**



Fonte: adaptado de McCormac (2007, p. 327).

As propriedades das barras do Exemplo 2 estão apresentadas na Tabela 5.

**Tabela 5 - Exemplo 2: Propriedades das barras**

Elementos	Área da seção transversal (cm <sup>2</sup> )	Módulo de Young (GPa)
1	12,904	199,955
2	12,904	199,955
3	6,452	199,955
4	6,452	199,955
5	6,452	199,955
6	6,452	199,955
7	12,904	199,955
8	12,904	199,955
9	12,904	199,955
10	12,904	199,955

Fonte: adaptado de adaptado de McCormac (2007, p. 327)

Os dados acima foram propostos na literatura em outras unidades de medida, havendo a necessidade de convertê-las para o SI antes de serem inseridas no código. O exemplo apresentado inicialmente consta in<sup>2</sup> para as áreas e psi para o módulo de elasticidade.

Na Tabela 6 são apresentados os deslocamentos nodais obtidos com o programa Risa 2D e com o código desenvolvido em Scilab.

**Tabela 6 - Deslocamentos nodais relativos ao exemplo 2**

Nó	Prog, Desenv.		Risa 2D		Diferença	
	$u_x$ (mm)	$u_y$ (mm)	$u_x$ (mm)	$u_y$ (mm)	$u_x$ (%)	$u_y$ (%)
1	4,731	-10,082	4,732	-10,081	0,029	0,012
2	2,013	-11,236	2,014	-11,235	0,038	0,011
3	0	0	0	0	0	0
4	1,892	-12,216	1,894	-12,217	0,102	0,011
5	3,590	-15,892	3,589	-15,891	0,016	0,007
6	6,113	0	6,112	0	0,012	0

Fontes: Autoria própria (2022).

É possível verificar que para o exemplo 2 o erro dos deslocamentos dos nós no sentido do eixo x tem valor máximo de 0,102%, enquanto que para o sentido do eixo y o valor máximo da diferença é de 0,012%.

Na Tabela 7 são apresentadas as reações de apoio obtidas pelo código de cálculo desenvolvido e com o software Risa 2D.

**Tabela 7 - Reações de apoio relativas ao exemplo 2**

Nó	Prog, Desenv.		Risa 2D		Diferença	
	$R_x$ (kN)	$R_y$ (kN)	$R_x$ (kN)	$R_y$ (kN)	$R_x$ (%)	$R_y$ (%)
3	0	66,754	0	66,723	0	0,046
6	0	88,996	0	88,964	0	0,035

Fontes: Autoria própria (2022).

Ao analisar os resultados elencados na Tabela 7, pode-se observar que o erro neste caso é de 0,046% para o nó 3 e 0,035% para o nó 6.

Os resultados referentes aos esforços internos nas extremidades do elemento são apresentados na tabela 8.

**Tabela 8 - Esforços internos relativos ao exemplo 2**

Elemento	Prog, Desenv.	Risa 2D	Diferença
	N (kN)	N (kN)	(%)
1	-95,873	-95,857	0,016
2	-94,404	-94,332	0,077
3	-37,636	-37,600	0,094
4	41,180	41,157	0,056
5	9,708	9,726	0,187
6	82,131	82,102	0,036
7	-125,868	-125,843	0,020
8	66,754	66,703	0,077
9	59,890	59,881	0,014
10	89,008	88,984	0,027

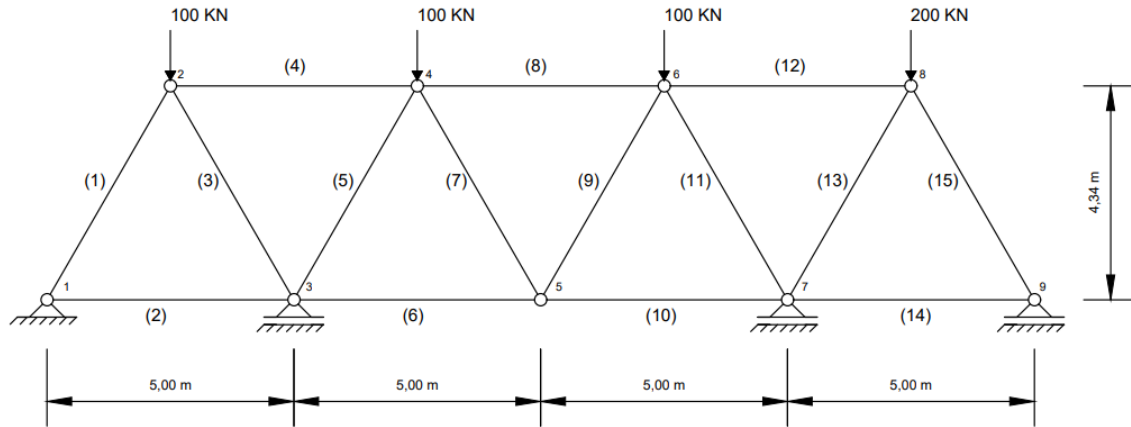
Fontes: Autoria própria (2022).

Quando se compara os resultados da Tabela 8, os esforços internos obtidos apresentam, com exceção do elemento 5, diferenças abaixo de 0,1%.

### 6.2.3 Exemplo 3

O exemplo 3, ilustrado pela Figura 11, foi extraído de Souza (1971). O software Ftool versão 4.00.0 foi utilizado para geração de resultados de comparação.

**Figura 11 – Exemplo 3: Geometria, ações atuantes e condições de contorno**



Fonte: adaptado de Souza (1971, p. 18).

As propriedades das barras do Exemplo 3 estão apresentadas na Tabela 9.

**Tabela 9 - Exemplo 3: Propriedades das barras**

Elementos	Área da seção transversal (cm <sup>2</sup> )	Módulo de Young (GPa)
1	30	200
2	5	200
3	30	200
4	30	200
5	30	200
6	5	200
7	5	200
8	30	200
9	5	200
10	5	200
11	30	200
12	30	200
13	30	200
14	5	200
15	30	200

Fonte: adaptado de Souza (1971, p. 18).

Na Tabela 10 têm-se os deslocamentos nodais obtidos com o Ftool e com o código desenvolvido em Scilab.

**Tabela 10 – Deslocamentos nodais relativos ao exemplo 3**

Nó	Prog, Desenv.		Prog, Ftool		Diferença	
	$u_x$ (mm)	$u_y$ (mm)	$u_x$ (mm)	$u_y$ (mm)	$u_x$ (%)	$u_y$ (%)
1	0	0	0	0	0	0
2	0,900	-0,665	0,901	-0,665	0,111	0
3	0,378	0	0,378	0	0	0
4	1,254	-1,584	1,254	-1,584	0	0
5	1,052	-1,509	1,052	-1,509	0	0
6	1,156	-1,377	1,156	-1,377	0	0
7	1,561	0	1,561	0	0	0
8	1,565	-1,588	1,565	-1,588	0	0
9	3,213	0	3,214	0	0,031	0

Fontes: Autoria própria (2022).

Pode-se notar que para o exemplo 3 as diferenças de deslocamentos dos nós 2 e 9 no sentido do eixo x são de 0,111% e 0,031% respectivamente, e para o sentido do eixo y não foram encontradas diferenças.

Na Tabela 11 têm-se as reações de apoio obtidas com os códigos de cálculo em questão.

**Tabela 11 – Reações de apoio relativas ao exemplo 3**

Nó	Prog, Desenv.		Prog, Ftool		Diferença	
	$R_x$ (kN)	$R_y$ (kN)	$R_x$ (kN)	$R_y$ (kN)	$R_x$ (%)	$R_y$ (%)
1	0	13,139	0	13,139	0	0
2	0	0	0	0	0	0
3	0	183,987	0	183,987	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0
7	0	245,484	0	245,484	0	0
8	0	0	0	0	0	0
9	0	57,390	0	57,390	0	0

Fontes: Autoria própria (2022).

Os resultados obtidos das reações de apoio na tabela 11 não apresentam diferenças. Por fim, são apresentados na Tabela 12 os esforços internos.

Tabela 12 – Esforços internos relativos ao exemplo 3

Elemento	Prog, Desenv.	Prog, Ftool	Diferença (%)
	N (kN)	N(kN)	
1	-15,163	-15,163	0
2	7,568	7,568	0
3	-100,242	-100,242	0
4	42,467	42,467	0
5	-112,087	-112,087	0
6	13,481	13,481	0
7	-3,317	-3,317	0
8	-11,825	-11,825	0
9	3,317	3,317	0
10	10,170	10,170	0
11	-118,722	-118,722	0
12	49,090	49,090	0
13	-164,578	-164,578	0
14	33,059	33,059	0
15	-66,231	-66,231	0

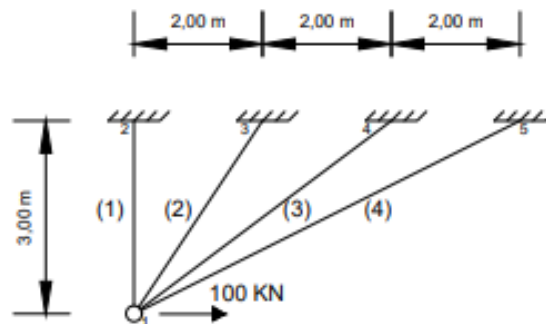
Fontes: Autoria própria (2022).

Os esforços normais obtidos e apresentados na tabela 3 demonstram a ausência de diferenças entre as respostas do código desenvolvido e o do Ftool.

#### 6.2.4 Exemplo 4

O exemplo 4 utilizado, ilustrado na Figura 12, também foi extraído de Souza (1971) e os resultados de comparação foram obtidos com o software Ftool, versão 4.00.0.

Figura 12 – Exemplo 4: Geometria, ações atuantes e condições de contorno



Fonte: adaptado de Souza (1971, p. 114).

As propriedades das barras do Exemplo 4 estão apresentadas na Tabela 13.

**Tabela 13 - Exemplo 4: Propriedades das barras**

Elementos	Área da seção transversal (cm <sup>2</sup> )	Módulo de Young (GPa)
1	1	200
2	2	200
3	3	200
4	4	200

Fonte: adaptado de Souza (1971, p. 114).

Na Tabela 14 são mostrados os deslocamentos nodais obtidos com o Ftool e com o programa desenvolvido em Scilab. É possível verificar que para o exemplo 4 os deslocamentos nodais não apresentam diferenças.

**Tabela 14 - Deslocamentos nodais relativos ao exemplo 4**

Nó	Prog, Desenv.		Prog, Ftool		Diferença	
	$u_x$ (mm)	$u_y$ (mm)	$u_x$ (mm)	$u_y$ (mm)	$u_x$ (%)	$u_y$ (%)
1	11,112	-8,261	11,112	-8,261	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0

Fontes: Autoria própria (2022).

Na Tabela 15 são apresentadas as reações de apoio obtidas para o Exemplo 4.

**Tabela 15 - Reações de apoio relativas ao exemplo 4**

Nó	Prog, Desenv.		Prog, Ftool		Diferença	
	$R_x$ (kN)	$R_y$ (kN)	$R_x$ (kN)	$R_y$ (kN)	$R_x$ (%)	$R_y$ (%)
1	0	0	0	0	0	0
2	0	55,073	0	55,069	0	0,007
3	4,367	6,550	4,365	6,549	0,046	0,015
4	-37,758	-28,318	-37,758	-28,317	0	0,004
5	-66,609	-33,304	-66,607	-33,302	0,003	0,006

Fontes: Autoria própria (2022).

Ao analisar os resultados elencados, pode-se observar que a diferença, neste caso, é de 0,046% para o nó 3 e 0,003% para o nó 5 ambos no sentido do eixo x, enquanto que para o sentido do eixo y a diferença máxima de 0,015%.

Os resultados referentes aos esforços internos nas extremidades de cada elemento são apresentados na tabela 16.



**Tabela 16 - Esforços internos relativos ao exemplo 4**

Elemento	Prog, Desenv.	Prog, Ftool	Diferença
	N (kN)	N (kN)	(%)
1	55,073	55,069	0,007
2	7,872	7,871	0,013
3	-47,198	-47,196	0,004
4	-74,471	-74,468	0,004

Fontes: Autoria própria (2022).

Quando se compara os esforços internos obtidos com o Ftool e com o código implementado, com exceção do elemento 2 que apresenta diferença de 0,013%, as diferenças nos demais permaneceram abaixo de 0,01%.

Por fim, é possível resumir as diferenças referentes aos deslocamentos, reações de apoio e esforços internos dos quatro casos. Para o Exemplo 1 a diferença é de 0,032 %, ao passo que para a estrutura do exemplo 2 a média das diferenças se aproxima de 0,057%, enquanto que para os exemplos 3 e 4 a média das diferenças entre as respostas obtidas fica em torno de 0,071% e 0,011%, respectivamente. Com isso podemos concluir pela eficiência e acurácia na resposta do código computacional desenvolvido para a análise elástica de treliças planas.

## 7 CONCLUSÃO

Um dos fatores que podem contribuir para o aumento do erro percentual de uma análise numérica é a precisão de casas decimais em que o problema é proposto. Em razão das conversões a serem feitas, o erro pode ser transferido desde o processo de conversão de unidades de medida até o resultado final obtido. Em todos os casos os erros foram insignificantes.

Existe também a possibilidade de resolver manualmente as estruturas demonstradas por meio dos processos clássico (método das forças e método dos deslocamentos), o que seria um processo extenso e propenso a erros. Ao passo que se um analista utilizar um código de cálculo, como o desenvolvido neste trabalho, após inseridos os dados iniciais do problema, ele pode obter os resultados em segundos. Vale ressaltar que o ciclo de processamento varia a depender da quantidade de nós, as configurações do computador (memória disponível, velocidade do processador, por exemplo) também exercem influência no tempo de resposta do programa.

Conclui-se que, devido a acurácia obtida, os valores alcançados pelo presente trabalho são fidedignos. Assim, foi estabelecido um código computacional capaz de solucionar diversos casos de treliças planas, além daquelas apresentadas neste trabalho.

## REFERÊNCIAS

- ARAÚJO, Camila Machado de. **Desenvolvimento de um algoritmo para análise estática e dinâmica de treliças planas**. 2015. 54 f. Tese (Doutorado) - Curso de Engenharia Mecânica, Universidade Federal Fluminense, Rio de Janeiro, 2015.
- CARREIRA, Marcelo Rodrigo. **Introdução ao Método dos Elementos Finitos**. Campo Mourão, PR: Universidade Tecnológica Federal do Paraná, 2019. Não publicado. 344 p.
- FISH, Jacob; BELYTSCHKO, Ted. **Um primeiro curso em elementos finitos**. Rio de Janeiro: Livros Técnicos Científicos Editora S.A., 2007.
- GOMES, Maria Idália da Silva. **Estudo e análise de treliças**. 2016. Disponível em: [https://www.researchgate.net/publication/301298120\\_Estudo\\_e\\_Analise\\_de\\_Trelicas](https://www.researchgate.net/publication/301298120_Estudo_e_Analise_de_Trelicas). Acesso em: 01 set. 2020.
- HUTTON, David V. **Fundamentals of finite element analysis**. New York: McGraw-Hill, 2004.
- LOGAN, Daryl L. **A First Course in the Finite Element Method**. 6. ed. Boston: Cengage Learning, 2015. 3 – 4 p.
- MCCORMAC, Jack C. **Structural Analysis: using classical and matrix methods**. 4. ed. Danvers: John Wiley And Sons, Inc., 2007. 327 – 328 p.
- SANTOS, Marcus Rogério Oliveira dos. **Aplicação do método clássico na resolução de sistemas planos hiperestáticos**. 2001. 115 f. - Curso de Engenharia Civil, Escola Politécnica da USP, São Paulo, 2001.
- SORIANO, Humberto Lima. **Elementos Finitos: Formulação e aplicação na estática e dinâmica das estruturas**. Rio de Janeiro: Editora Ciência Moderna, 2009. 8 – 12 p.
- SOUZA, João Carlos de Antunes e. **Exercícios de estática das estruturas**. São Paulo: Universidade de São Paulo, 1971. 3.01 – 23.01 p.
- SOUZA, M.; RODRIGUES, R. **Sistemas estruturais de edificações e exemplos**. Notas de aula — Faculdade de Engenharia Civil, Arquitetura e Urbanismo - Departamento de Estruturas UNICAMP, Campinas, 2008. 29 – 30 p.

**APÊNDICE A - Código Computacional Desenvolvido**

```

xdel(winsid());
clear;
clc;
nnos=11;
nel=19;
no=[0,0; 0,3; 2.4,0; 2.4,3; 4.8,0; 4.8,3; 7.2,0; 7.2,3; 9.6,0; 9.6,3; 12,3]
conect=[1,2; 1,3; 1,4; 2,4; 3,4; 3,5; 3,6; 4,6; 5,6; 5,7; 5,8; 6,8; 7,8; 7,9; 7,10; 8,10; 9,10;
9,11; 10,11];
F=[0;0; 0;0; 0;0; 0;-126E3; 0;0; 0;0; 0;0; 0;-126E3; 0;0; 0;0; 72E3;0;];
C=[1;1; 0;0; 0;0; 0;0; 0;0; 0;0; 0;0; 0;0; 0;1; 0;0; 0;0;]
E =200E9*[1; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1];
A=1E-4*[1; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1];
mfprintf(%io(2),'***** \n');
mfprintf(%io(2),' Programa para Resolução de Treliças Espaciais pela Rigidez Direta
\n');
mfprintf(%io(2),'***** \n');
mfprintf(%io(2),' \n');
mfprintf(%io(2),'***** Dados de Entrada ***** \n');
mfprintf(%io(2),' \n');
mfprintf(%io(2),'----- Coordenadas dos nós ----- \n');
mfprintf(%io(2),'Nó\t x\t y\n');
for i=1:nnos
    mfprintf(%io(2),'%d\t %5.3f\t %5.3f\n',i, no(i,1),no(i,2));
end
mfprintf(%io(2),'----- \n');
mfprintf(%io(2),' \n');
mfprintf(%io(2),'----- Conectividade dos elementos ----- \n');
mfprintf(%io(2),'elemento\t Nó inicial\t Nó final\n');
for i=1:nel
    mfprintf(%io(2),'%d\t %d\t %d\n',i, conect(i,1),conect(i,2));
end
mfprintf(%io(2),'----- \n');
mfprintf(%io(2),' \n');
mfprintf(%io(2),'----- Forças aplicadas ----- \n');
mfprintf(%io(2),'Nó\t Fx\t Fy\n');
for i=1:nnos
    mfprintf(%io(2),'%d\t %0.2f\t %0.2f\n',i, F(2*i-1),F(2*i));
end
mfprintf(%io(2),'----- \n');
mfprintf(%io(2),' \n');
mfprintf(%io(2),'----- Condições de contorno ----- \n');
mfprintf(%io(2),' 1= restringido 0 = livre \n');
mfprintf(%io(2),' \n');
mfprintf(%io(2),'Nó\t Ux\t Uy\n');
for i=1:nnos
    mfprintf(%io(2),'%d\t %d\t %d\n',i, C(2*i-1),C(2*i));
end
mfprintf(%io(2),'----- \n');

```

```

mfprintf(%io(2),' \n');
mfprintf(%io(2),'----- Dados dos elementos ----- \n');
mfprintf(%io(2),'Elemento\t Area\t E\n');
for i=1:nel
    mfprintf(%io(2),'%d\t %5.3e\t %5.3e\n',i, A(i),E(i));
end
mfprintf(%io(2),'----- \n');
mfprintf(%io(2),' \n');
mfprintf(%io(2),'***** \n');
mfprintf(%io(2),' \n');
mfprintf(%io(2),' \n');
K=zeros(2*nnos,2*nnos);
L=zeros(nel);
for e=1:nel
    X1=no(conect(e,1),1);
    X2=no(conect(e,2),1);
    Y1=no(conect(e,1),2);
    Y2=no(conect(e,2),2);
    L(e)=sqrt((X2-X1)^2+(Y2-Y1)^2);
    cx(e)=(X2-X1)/L(e);
    cy(e)=(Y2-Y1)/L(e);
    k(e)=E(e)*A(e) / L(e);
    Ke=zeros(4,4);
    Ke(1,1)=k(e)*cx(e)^2;
    Ke(1,2)=k(e)*cx(e)*cy(e);
    Ke(1,3)=-k(e)*cx(e)^2;
    Ke(1,4)=-k(e)*cx(e)*cy(e);
    Ke(2,1)=Ke(1,2);
    Ke(2,2)=k(e)*cy(e)^2;
    Ke(2,3)=-k(e)*cx(e)*cy(e);
    Ke(2,4)=-k(e)*cy(e)^2;
    Ke(3,1)=Ke(1,3);
    Ke(3,2)=Ke(2,3);
    Ke(3,3)=k(e)*cx(e)^2;
    Ke(3,4)=k(e)*cx(e)*cy(e);
    Ke(4,1)=Ke(1,4);
    Ke(4,2)=Ke(2,4);
    Ke(4,3)=Ke(3,4);
    Ke(4,4)=k(e)*cy(e)^2;
    m=conect(e,1);
    n=conect(e,2);
    B=zeros(4,2*nnos);
    for j=1:2*nnos
        if j==(2*m-1) then
            B(1,j)=1;
        end
        if j==(2*m) then
            B(2,j)=1;
        end
    end
end

```

```

    if j==(2*n-1) then
        B(3,j)=1;
    end
    if j==(2*n) then
        B(4,j)=1;
    end
end
end
mfprintf(%io(2),'----- Matriz de incidência cinemática do elemento %d ----- \n',e);
for i=1:4
    for j=1:2*nnos
        if j < 2*nnos then
            mfprintf(%io(2),'%2.2f# \t', B(i,j));
        end

        if j == 2*nnos then
            mfprintf(%io(2),'%2.2f## \n',B(i,j));
        end
    end
end
end
mfprintf(%io(2),'----- \n');
K2=zeros(2*nnos,2*nnos);
K3=zeros(2*nnos,2*nnos);
K3=(B'*Ke*B);
for i=1:2*nnos
    for j=1:2*nnos
        K(i,j)=K(i,j)+K3(i,j);
        K2(i,j)=K(i,j);
    end
end
end
mfprintf(%io(2),'----- Matriz de rigidez da estrutura ----- \n');
for i=1:2*nnos
    for j=1:2*nnos
        if j < 2*nnos then
            mfprintf(%io(2),'%5.3e# \t', K(i,j));
        end
        if j == 2*nnos then
            mfprintf(%io(2),'%5.3e## \n',K(i,j));
        end
    end
end
end
mfprintf(%io(2),'----- \n');
mfprintf(%io(2),'***** Cálculos iniciais ***** \n');
mfprintf(%io(2),' \n');
mfprintf(%io(2),'----- Comprimento, cossems e rigidez axial dos elementos ---- \n');
mfprintf(%io(2),'Elemento\t L\t CX\t CY\t K\n');
for i=1:nel
    mfprintf(%io(2),'%d\t %5.3f\t %5.3f\t %5.3f\t %5.3f\n',i, L(i),cx(i), cy(i), k(i));
end

```

```

mfprintf(%io(2),'----- \n');
mfprintf(%io(2),'***** \n');
mfprintf(%io(2),'\n');
mfprintf(%io(2),'\n');
F2=F;
for i=1:2*nnos
    if C(i)== 1 then
        for j=1:2*nnos
            K2(i,j)=0.0;
            K2(j,i)=0.0;
        end
        F2(i)=0.0;
        K2(i,i)=1.0;
    end
end
mfprintf(%io(2),'-- Matriz de rigidez da estrutura com as cond. de contorno ---- \n');
for i=1:2*nnos
    for j=1:2*nnos
        if j < 2*nnos then
            mfprintf(%io(2),'%5.3e# \t', K2(i,j));
        end
        if j == 2*nnos then
            mfprintf(%io(2),'%5.3e## \n',K2(i,j));
        end
    end
end
mfprintf(%io(2),'----- \n');
sp=sparse(K2);
U=lusolve(sp,F2);
mfprintf(%io(2),'***** Resultados ***** \n');
mfprintf(%io(2),'\n');
mfprintf(%io(2),'----- Deslocamentos nodais ----- \n');
l=0;
mfprintf(%io(2),'Nó\t UX\t    UY\n');
for i=1:+2:2*nnos //
    l=l+1;
    mfprintf(%io(2),'%d\t %.5e\t %.5e\n',l, U(i), U(i+1));
end
mfprintf(%io(2),'----- \n');
mfprintf(%io(2),'\n');
mfprintf(%io(2),'\n');
RF=zeros(2*nnos,1);
R=zeros(2*nnos,1);
RF=K*U;
R=RF-F;
mfprintf(%io(2),'----- Reações de apoio----- \n');
l=0;
mfprintf(%io(2),'Nó\t RX\t RY\n');
for i=1:+2:2*nnos

```



```

l=l+1;
    mfprintf(%io(2),'%d\t %.2f\t %.2f\n',l, R(i), R(i+1));
end
mfprintf(%io(2),'----- \n');
mfprintf(%io(2),' \n');
mfprintf(%io(2),' \n');
N=zeros(nel);
for e=1:nel
    Ke=zeros(4,4);
    Ke(1,1)=k(e)*cx(e)^2;
    Ke(1,2)=k(e)*cx(e)*cy(e);
    Ke(1,3)=-k(e)*cx(e)^2;
    Ke(1,4)=-k(e)*cx(e)*cy(e);
    Ke(2,1)=Ke(1,2);
    Ke(2,2)=k(e)*cy(e)^2;
    Ke(2,3)=-k(e)*cx(e)*cy(e);
    Ke(2,4)=-k(e)*cy(e)^2;
    Ke(3,1)=Ke(1,3);
    Ke(3,2)=Ke(2,3);
    Ke(3,3)=k(e)*cx(e)^2;
    Ke(3,4)=k(e)*cx(e)*cy(e);
    Ke(4,1)=Ke(1,4);
    Ke(4,2)=Ke(2,4);
    Ke(4,3)=Ke(3,4);
    Ke(4,4)=k(e)*cy(e)^2;
    Te=zeros(4,4);
    Te(1,1)= cx(e);
    Te(1,2)= cy(e);
    Te(2,1)= -cy(e);
    Te(2,2)= cx(e);
    Te(3,3)= cx(e);
    Te(3,4)= cy(e);
    Te(4,3)= -cy(e);
    Te(4,4)= cx(e);
    ue(1)=U((conect(e,1))^2-1);
    ue(2)=U((conect(e,1))^2);
    ue(3)=U((conect(e,2))^2-1);
    ue(4)=U((conect(e,2))^2);
    fe=zeros(2,1);
    fe=Te*Ke*ue;
    if e==1 then
        mfprintf(%io(2),'----- Esforços e Tensões nos Elementos ----- \n');
        mfprintf(%io(2),'Elemento\t Esforço Normal\t \t Tensão Normal\n');
    end
    mfprintf(%io(2),'%d\t %.2f\t %.5e\n',e, -fe(1),- fe(1)/A(e));
    if e==nel then
        mfprintf(%io(2),'----- \n');
        mfprintf(%io(2),' \n');
        mfprintf(%io(2),' \n');
    end
end

```

```
mfprintf(%io(2), '***** FIM ***** \n');  
end  
end
```