

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
COORDENAÇÃO DE ENGENHARIA MECÂNICA  
ENGENHARIA MECÂNICA**

**EDILSON GABRIEL VERUZ**

**MÉTODO DOS ELEMENTOS FINITOS UTILIZANDO  
FUNÇÕES DE FORMA DE ALTA ORDEM APLICADAS  
EM ESTRUTURAS EM ESTADO PLANO DE TENSÃO**

**TRABALHO DE CONCLUSÃO DE CURSO**

**GUARAPUAVA  
2020**

**EDILSON GABRIEL VERUZ**

**MÉTODO DOS ELEMENTOS FINITOS UTILIZANDO FUNÇÕES DE  
FORMA DE ALTA ORDEM APLICADAS EM ESTRUTURAS EM  
ESTADO PLANO DE TENSÃO**

Trabalho de Conclusão de Curso de Graduação, apresentado à disciplina de TCC 2, do curso de Engenharia Mecânica da Coordenação de Engenharia Mecânica (COEME) da Universidade Tecnológica Federal do Paraná (UTFPR) como requisito parcial para obtenção do título de Engenheiro Mecânico.

**Orientador:** Prof<sup>o</sup> Msc. Luan Jose Franchini  
Ferreira

**GUARAPUAVA**

**2020**



Ministério da Educação  
**Universidade Tecnológica Federal do Paraná**  
Câmpus Guarapuava  
Coordenação de Engenharia Mecânica



## **TERMO DE APROVAÇÃO**

**MÉTODO DOS ELEMENTOS FINITOS UTILIZANDO FUNÇÕES DE FORMA DE ALTA ORDEM  
APLICADAS EM ESTRUTURAS EM ESTADO PLANO DE TENSÃO**

por

**EDILSON GABRIEL VERUZ**

Este Trabalho de Conclusão de Curso foi apresentado em Guarapuava, Paraná na data 28 de maio de 2020 como requisito parcial para a obtenção do título de Bacharel em Engenharia Mecânica. O candidato foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

---

Luan José Franchini Ferreira  
Prof. Orientador

---

David Lira Nuñez  
Membro Da Banca

---

Carlos Francisco Pecapedra Souza  
Membro Da Banca

---

Aldo Przybysz  
Coordenador do Curso de Engenharia Mecânica

---

Edilson Gabriel Veruz

- O Termo de Aprovação assinado encontra-se na Coordenação do Curso -

# Resumo

VERUZ, Edilson Gabriel; MÉTODO DOS ELEMENTOS FINITOS UTILIZANDO FUNÇÕES DE FORMA DE ALTA ORDEM APLICADAS EM ESTRUTURAS EM ESTADO PLANO DE TENSÃO. 86 f. Trabalho de Conclusão de Curso, Engenharia Mecânica, Universidade Tecnológica Federal do Paraná, Guarapuava, 2020.

Desde sua popularização, o Método dos Elementos Finitos (MEF) recebeu diversas abordagens, com diferentes níveis de detalhamento. Uma dessas abordagens, o triângulo de deformação constante (CST) é muito difundido no meio acadêmico, principalmente para problemas de Estado Plano de Tensão (EPT) por sua simplicidade de aplicação e boa aproximação de soluções reais. Contudo em projetos estruturais, devido ao compromisso com a performance e a segurança, novas abordagens foram empregadas visando diminuir o erro sem elevar o custo computacional, a exemplo da aplicação de funções de interpolação de ordens superiores. Diante disso, este trabalho buscou abordar o MEF para estruturas submetidas a um EPT utilizando funções de forma de alta ordem para elementos triangulares, construídas através do conceito de produto tensorial entre funções unidimensionais usando bases modais com os polinômios ortogonais de Jacobi. O algoritmo foi desenvolvido na linguagem de programação Python com auxílio da plataforma Jupyter Notebook e, os resultados de tensão e de deslocamento obtidos foram comparados com software comercial. O emprego das funções de forma de alta ordem ao método obteve êxito ao se equiparar aos resultados de softwares específicos sem a necessidade de refino das malhas das geometrias estudadas.

**Palavras-chaves:** Método dos Elementos Finitos, Estado Plano de Tensão, Polinômios Ortogonais de Jacobi, Python.

# Abstract

VERUZ, Edilson Gabriel; FINITE ELEMENT METHOD USING HIGH ORDER SHAPE FUNCTIONS APPLIED TO STRUCTURES IN A STRESS PLANE STATE. 86 f. Course Completion Work, Mechanical Engineering, Federal Technology University - Paraná, Guarapuava, 2020.

Since its popularization, the Finite Element Method (FEM) has received several approaches, with different levels of detail. One of these approaches, the constant strain triangle (CST) is very widespread in the academic environment, mainly for Plane State of Stress (PSS) problems due to its simplicity of application and good approximation of real solutions. However, in structural projects, due to the commitment to performance and security, new approaches were used to reduce the error without increasing the computational cost, such as the application of interpolation functions of higher orders. Therefore, this work sought to approach the FEM for structures submitted to an EPT using high order form functions for triangular elements, built through the concept of tensor product between unidimensional functions using modal bases with Jacobi's orthogonal polynomials. The algorithm was developed in the Python programming language with the aid of the Jupyter Notebook platform, and the stress and displacement results obtained were compared with commercial software. The use of functions shape of a high order to the method was successful in being equal to the results of specific software without the need to refine the meshes of the studied geometries.

**Key-words:** Finite Elements Method, Plane State of Stress, Jacobi's Orthogonal Polynomials, Python

# Lista de ilustrações

Figura 2.1 – Placa retangular com furo central com carga axial . . . . .	18
Figura 2.2 – Geometria aproximada por elementos triangulares . . . . .	18
Figura 2.3 – Representação geométrica por diversos gêneros de elementos não-estruturados (triângulos) e elementos estruturados (quadriláteros) . . . . .	19
Figura 2.4 – Elemento Infinitesimal em um Estado Triaxial de tensão. . . . .	22
Figura 2.5 – Representação do Estado Plano de Tensão . . . . .	22
Figura 2.6 – Transformações das equações de um sistema físico . . . . .	24
Figura 3.1 – Sistema considerado como meio contínuo e sua discretização por elementos finitos . . . . .	26
Figura 3.2 – Geometria discretizada por meio da biblioteca <i>Triangle</i> . . . . .	27
Figura 3.3 – Coordenadas naturais baricêntricas para o triângulo linear . . . . .	29
Figura 3.4 – Elemento triangular plano quadrático . . . . .	31
Figura 3.5 – Elemento triangular plano quártico . . . . .	32
Figura 3.6 – Processo de montagem da Matriz Global de Rigidez e do Vetor Global de Carregamento . . . . .	36
Figura 3.7 – Placa quadrada discretizada por dois elementos. Os nós da geometria estão identificados pela numeração global . . . . .	36
Figura 3.8 – Esquemático do funcionamento geral do algoritmo do MEF para problemas em EPT. . . . .	42
Figura 4.1 – Placa quadrangular engastada com carregamento axial . . . . .	48
Figura 4.2 – Placa quadrada discretizada por triângulos lineares . . . . .	48
Figura 4.3 – Convergência da tensão máxima de Von Mises para o problema da placa quadrada. . . . .	49
Figura 4.4 – Campos de tensões de Von Mises para o problema da placa quadrada. Utilizando uma malha fixa de 968 elementos. A suavidade do campo aumenta com o grau de interpolação empregado. . . . .	50
Figura 4.5 – Campos de tensões de Von Mises para o problema da placa quadrada utilizando o refinamento -h . . . . .	51
Figura 4.6 – Campos de deslocamentos totais para o problema da placa quadrada. A suavidade é percebida desde a ordem 1, pois o comportamento de $\{u\}$ é linear ao longo do elemento. . . . .	52

Figura 4.7 – Energia de deformação (J) para o problema da placa quadrada. A taxa de convergência da energia de deformação aumenta ao passo que se eleva o grau dos polinômios de interpolação . . . . .	53
Figura 4.8 – Campos de tensões de Von Mises e de deslocamento total para o problema da placa quadrada resolvido no software Ansys 2019 R3 versão estudantil. Malha utilizada contou com 90 nós e 146 elementos . . . . .	54
Figura 4.9 – Distribuição dos pontos na geometria com aumento da densidade de nós na região próxima ao engaste. Malha de 200 elementos e 121 nós . . . . .	57
Figura 4.10–Placa em formato de "L"com carregamento axial . . . . .	58
Figura 4.11–Placa em formato de "L"discretizada por triângulos . . . . .	59
Figura 4.12–Convergência da tensão máxima de Von Mises para o problema da placa em "L" . . . . .	60
Figura 4.13–Campos de tensões de Von Mises para o problema da placa em "L". Em todas as situações utilizou-se malha fixa com 864 elementos . . . . .	61
Figura 4.14–Campos de tensões de Von Mises para o problema da placa em "L". Refinamento -h . . . . .	62
Figura 4.15–Campos de deslocamentos totais para o problema da placa em "L". Em todas as situações utilizou-se malha fixa com 864 elementos . . . . .	63
Figura 4.16–Convergência da Energia de Deformação (J) para o problema da placa em "L". . . . .	64
Figura 4.17–Campos de tensões e de deslocamentos totais para o problema da placa em "L"resolvido no software Ansys 2019 R3 versão estudantil. A malha utilizada contou com 83 nós e 126 elementos . . . . .	65
Figura 4.18–Campo de tensões de Von Mises para o problema da placa em "L"com refino da malha solucionado no software Ansys 2019 R3 versão estudantil. A malha utilizada contou com 3.318 GDL e 3.168 elementos . . . . .	66
Figura 4.19–Placa retangular com furo central com carregamento axial . . . . .	67
Figura 4.20–Seção analisada da placa retangular com furo central com carregamento axial . . . . .	68
Figura 4.21–Placa retangular com furo central discretizada por triângulos . . . . .	68
Figura 4.22–Convergência da tensão máxima de Von Mises para o problema da placa circular com furo central. A taxa de convergência aumenta com o grau das funções interpoladoras utilizadas. . . . .	70
Figura 4.23–Campo de tensões de Von Mises obtido para o teste VS08 - Placa com um furo - do manual de validação . . . . .	70
Figura 4.24–Campos de tensões de Von Mises para a placa retangular com furo e carga axial. A malha para as três situações contém 324 elementos . . . . .	72
Figura 4.25–Campos de tensões de Von Mises para a placa retangular com furo e carga axial. Refinamento -h . . . . .	72

Figura 4.26–Campos de deslocamentos totais para a placa retangular com furo e carga axial. A malha para as três situações contém 324 elementos . . .	73
Figura 4.27–Energia de Deformação (J) para o problema da placa com furo central. A taxa de convergência em função de P demonstra comportamento monotônico, com maiores velocidades para $P > 1$ . . . . .	74
Figura 4.28–Campos de tensões de Von Mises e deslocamentos totais para o problema da placa retangular com furo em uma malha de 95 nós e 135 elementos. Resultados obtidos pelo software Ansys 2019 R3 versão estudantil . . .	75



# Lista de tabelas

Tabela 3.1 – Coordenadas nodais da geometria . . . . .	28
Tabela 3.2 – Tabela de conectividade dos elementos . . . . .	28
Tabela 3.3 – Índices de tensorização para o elemento triangular plano quadrático . .	31
Tabela 3.4 – Índices de tensorização para o elemento triangular plano quártico . . .	32
Tabela 3.5 – Tabela conectividade para o exemplo da 3.7. Utilizou-se a numeração global para realizar o mapeamento dos nós. . . . .	37
Tabela 3.6 – Tabela LOCE na matriz global de rigidez e no vetor global de carregamento	37
Tabela 3.7 – Matriz de rigidez do elemento 1 com índices de localização na matriz global de rigidez . . . . .	37
Tabela 3.8 – Matriz de rigidez do elemento 2 com índices de localização na matriz global de rigidez . . . . .	38
Tabela 3.9 – Condições de contorno para a placa quadrada discretizada por dois elementos . . . . .	39
Tabela 4.1 – Propriedades geométricas e mecânicas da placa quadrada . . . . .	47
Tabela 4.2 – Comparativo dos resultados do software e o algoritmo para o problema da placa quadrada . . . . .	55
Tabela 4.3 – Distribuição de pontos, em porcentagem, ao longo da geometria . . . .	55
Tabela 4.4 – Tensão máxima de Von Mises obtida para cada caso de distribuição de pontos apresentados na Tabela 4.3. . . . .	56
Tabela 4.5 – Propriedades geométricas e mecânicas da placa retangular em formato de "L" . . . . .	58
Tabela 4.6 – Comparativo dos resultados do software e o algoritmo para o problema da placa em formato "L" . . . . .	66
Tabela 4.7 – Propriedades geométricas e mecânicas da placa com furo central. . . .	67
Tabela 4.8 – Comparativo dos resultados do software e o algoritmo para a placa retangular com furo central . . . . .	74

# Lista de abreviaturas e siglas

CST	<i>Constant Strain Triangle</i> ( Triângulo de Deformação Constante)
EDP	Equação Diferencial Parcial
GDL	Grau de Liberdade
MEF	Método dos Elementos Finitos
MEF-AO	Método dos Elementos Finitos de Alta Ordem
PTV	Princípio dos Trabalhos Virtuais
LOCE	Localização dos Elementos

# Lista de símbolos

$\Omega$	Domínio da geometria
$\Gamma$	Fronteira da geometria
$\sigma_i$	Componente $i$ de tensão normal
$\tau_{ij}$	Componente $ij$ de tensão de cisalhamento
$\varepsilon_i$	Deformação específica na direção $i$
$\gamma_j$	Deformação por cisalhamento $ij$
$E$	Módulo de elasticidade
$G$	Módulo de elasticidade transversal
$\nu$	Coefficiente de Poisson
$D$	Matriz constitutiva
$\nabla$	Operador Gradiente
$\phi$	Função arbitrária
$\Psi$	Deslocamento nodal arbitrário
$W_i$	$i$ -ésima Função Peso
$u$	Solução exata do campo de deslocamento
$\tilde{u}$	Solução Aproximada de $u$
$t_e$	Espessura do elemento
$K^e$	Matriz elementar de rigidez
$K$	Matriz global de rigidez do sistema
$F^e$	Vetor elementar de carregamento
$F$	Vetor global carregamento do sistema

$\xi$	Coordenada local do elemento na direção $\xi$
$\eta$	Coordenada local do elemento na direção $\eta$
$L_1$	Coordenada natural do elemento triangular na direção $L_1$
$L_2$	Coordenada natural do elemento triangular na direção $L_2$
$L_3$	Coordenada natural do elemento triangular na direção $L_3$
$\mathcal{L}$	Função de interpolação de Lagrange
$N_i$	$i$ -ésima Função de Forma
$P^{\alpha,\beta}$	Polinômio ortogonal de Jacobi de peso $\alpha$ e $\beta$
$P$	Ordem de interpolação
$p$	Índice de tensorização da direção $L_1$
$q$	Índice de tensorização da direção $L_2$
$r$	Índice de tensorização da direção $L_3$
$\phi_p$	Função de base para o índice de tensorização $p$
$\phi_q$	Função de base para o índice de tensorização $q$
$\phi_r$	Função de base para o índice de tensorização $r$

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>14</b>
<b>1.1</b>	<b>JUSTIFICATIVA</b>	<b>15</b>
<b>1.2</b>	<b>OBJETIVOS</b>	<b>16</b>
1.2.1	Objetivos Gerais	16
1.2.2	Objetivos Específicos	16
<b>2</b>	<b>FORMULAÇÃO DO MÉTODO DOS ELEMENTOS FINITOS PARA SÓLIDOS LINEARES-ELÁSTICOS</b>	<b>17</b>
<b>2.1</b>	<b>O Método dos Elementos Finitos</b>	<b>17</b>
<b>2.2</b>	<b>Os Tipos de Elementos</b>	<b>19</b>
<b>2.3</b>	<b>Os Elementos de Ordem Superior</b>	<b>20</b>
<b>2.4</b>	<b>A Forma Forte do Problema</b>	<b>21</b>
<b>2.5</b>	<b>O Estado Plano de Tensão</b>	<b>21</b>
2.5.1	As Hipóteses do Estado Plano de Tensão	23
2.5.2	As Equações de Equilíbrio	23
<b>2.6</b>	<b>A Forma Fraca do Problema</b>	<b>24</b>
<b>3</b>	<b>ASPECTOS METODOLÓGICOS</b>	<b>26</b>
<b>3.1</b>	<b>A Geração da Malha</b>	<b>27</b>
<b>3.2</b>	<b>As Funções de Forma</b>	<b>28</b>
3.2.1	Índices de tensorização	31
<b>3.3</b>	<b>Transformações Geométricas e os Sistemas de Coordenadas</b>	<b>32</b>
<b>3.4</b>	<b>Matriz de Elasticidade e a Matriz Jacobiana</b>	<b>33</b>
<b>3.5</b>	<b>As Matrizes Elementares</b>	<b>34</b>
<b>3.6</b>	<b>Construção das Matrizes Globais</b>	<b>35</b>
3.6.1	Condições de contorno	38
<b>3.7</b>	<b>Integração Numérica</b>	<b>40</b>
<b>3.8</b>	<b>Algoritmos e Suas Considerações</b>	<b>41</b>
3.8.1	Arquivo de entrada	43
<b>3.9</b>	<b>O Software Ansys</b>	<b>46</b>
<b>4</b>	<b>RESULTADOS E DISCUSSÕES</b>	<b>47</b>
<b>4.1</b>	<b>Estudo de Caso 1: Placa Quadrada</b>	<b>47</b>
<b>4.2</b>	<b>Estudo de Caso 2: Placa Retangular em Formato de L</b>	<b>58</b>
<b>4.3</b>	<b>Estudo de caso 3: Placa retangular com furo central</b>	<b>67</b>

---

<b>5</b>	<b>CONCLUSÃO</b> . . . . .	<b>77</b>
	<b>REFERÊNCIAS</b> . . . . .	<b>78</b>
<b>APÊNDICE A</b>	<b>CÓDIGO IMPLEMENTADO EM <i>PYTHON</i></b> . . . . .	<b>80</b>

# Capítulo 1

## Introdução

Com o avanço tecnológico, engenheiros se deparam com projetos complexos que demandam confiabilidade e segurança e, atingir esses objetivos é reflexo da modelagem matemática apropriada. Mecânica dos sólidos, fenômenos dos transportes e demais ciências da engenharia são usadas para descrever o comportamento físico dos sistemas e, normalmente, a formulação resulta em expressões matemáticas, denotadas por um conjunto de Equações Diferenciais Parciais.

A solução analítica para essas Equações Diferenciais Parciais, em geral, é apenas concebida para casos simples e em problemas reais de engenharia, adota-se estratégias de aproximação, como o Método dos Elementos Finitos (MEF). O emprego dessa ferramenta permite a análise de geometrias complexas com a possibilidade de modificar os parâmetros de entrada e sem a necessidade da criação de protótipos, reduzindo custos associados a produção sem perder a fidelidade de um modelo real (DHATT; LEFRANÇOIS; TOUZOT, 2012).

O MEF é dotado de três características básicas que lhe torna um atrativo. Primeiro, é a divisão do domínio contínuo, comumente complexo, em subdomínios denominados elementos finitos. Segundo, para cada elemento, a ideia de que se pode aproximar qualquer função contínua por combinações lineares de polinômios algébricos e terceiro, as relações algébricas de cada coeficiente indeterminado dos elementos podem ser obtidos para cada elemento, satisfazendo as equações governantes, por integração numérica. Nesse sentido, o MEF pode ser visto como a aplicação, elemento a elemento, dos métodos variacionais - Rayleigh-Ritz - ou resíduos ponderados - Galerkin (REDDY, 1993).

Apesar do MEF atender bem a maioria dos problemas de engenharia, para algumas aplicações é necessário que o erro seja mínimo. Derivações do MEF surgiram com o intuito de reduzir o erro através de: refinamento da malha a partir do aumento do número de elementos e de tamanhos variados (versão -h); aumento do grau polinomial de interpolação (versão -p); e a combinação dos casos anteriores (versão -hp) (BABUŠKA; GUO, 1992)

A eficiência associada aos métodos citados muito se deve as funções de aproximação utilizadas. A literatura traz uma variedade de procedimentos para a construção dessas,

destacando-se as funções que utilizam entidades físicas - funções de Lagrange e Serendipity, por exemplo - e as que usam bases modais. Dentre as funções de forma com bases modais, Karniadakis e Sherwin (2013) apresentam a construção utilizando produto tensorial de funções unidimensionais.

Na mecânica estrutural, o MEF se demonstra uma ferramenta importante, dadas as necessidades e complexidades envolvidas no comportamento físico dos projetos. Dentro da teoria da elasticidade, situações nas quais estruturas possuem largura e comprimento muito superiores a espessura, são utilizadas de hipóteses simplificadoras, o EPT (REDDY, 1993).

Essas considerações englobam diversas situações reais e podem ser aplicáveis em vigas, placas finas e vasos de pressão validando a necessidade de softwares de solução especializados nessas estruturas.

Sendo assim, o objetivo deste trabalho é desenvolver um algoritmo de MEF que utilize o produto tensorial de funções unidimensionais para a construção das funções de forma para elementos não estruturados proposto por Karniadakis e Sherwin (2013) e leve em consideração as hipóteses de EPT para o procedimento de solução.

## 1.1 JUSTIFICATIVA

Devido a escassez dos recursos naturais e o desenvolvimento de tecnologias, os projetos de engenharia passaram a ter compromisso com a redução de custo, o uso consciente dos materiais disponíveis e a máxima performance. Visto que os métodos analíticos possuem limitações para solucionar os problemas reais e os testes empíricos são dispendiosos e requerem tempo, um recurso encontrado é o emprego de métodos numéricos para obter soluções próximas às reais.

O tratamento apropriado dos problemas de estado plano de tensão, com o uso do MEF e a incorporação desses por linguagens de programação permite a criação de programas poderosos e responsivos, essenciais para o desenvolvimento de diversas áreas da engenharia (SMITH; GRIFFITHS; MARGETTS, 2013).

Por outro lado, dado a limitação de soluções de EDP associadas a problemas de elasticidade, é importante a comparação das respostas dos sistemas obtidas pelos algoritmos com softwares já consolidados no mercado, a exemplo do Ansys, verificando a coerência dos resultados.

Portanto, é desejável o estudo e o desenvolvimento de um algoritmo simples e responsivo, para problemas mecânicos, considerando estruturas submetidas ao Estado Plano de Tensão, utilizando o método dos elementos finitos como ferramenta de solução visando obter soluções de deslocamento e tensão e, comparar seus resultados com software comercial.



## 1.2 OBJETIVOS

### 1.2.1 Objetivos Gerais

São objetivos gerais implementar um algoritmo de solução do Método dos Elementos Finitos de Alta Ordem (MEF-AO) para estruturas mecânicas submetidas a um Estado Plano de Tensão abordando o uso de funções de alta ordem, modais e hierárquicas.

### 1.2.2 Objetivos Específicos

Para atender o descrito acima, os seguintes passos serão tomados:

1. Implementar algoritmo de solução do Método dos Elementos Finitos (MEF) na linguagem de programação Python;
2. Construir as funções de interpolação de alta ordem por meio do produto tensorial de funções unidimensionais de bases modais utilizando polinômios ortogonais de Jacobi;
3. Aplicar algoritmo, através de simulações, em problemas práticos de Estado Plano de Tensão.
4. Comparar resultados obtidos com o software comercial Ansys R3 2019 Student Version.

# Capítulo 2

## Formulação do Método dos Elementos Finitos Para Sólidos Lineares-Elásticos

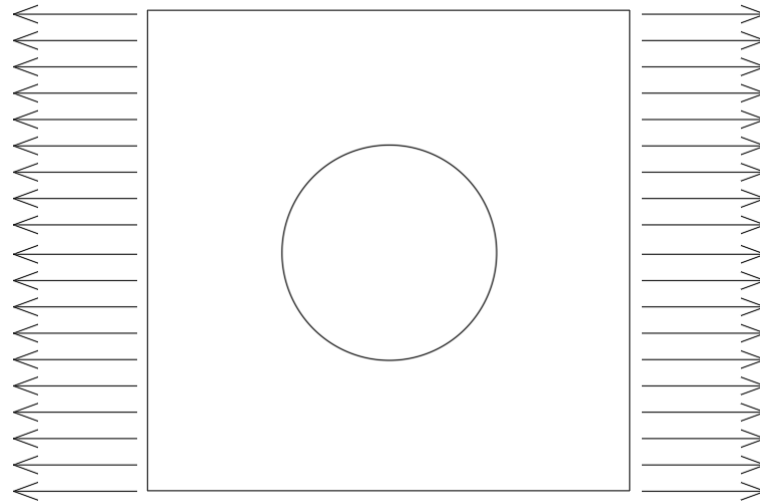
Neste capítulo, serão discutidos tópicos essenciais para a compreensão e desenvolvimento do Método dos Elementos Finitos de Alta Ordem e suas ferramentas de resolução para estruturas mecânicas submetidas a um Estado Plano de Tensão.

### 2.1 O Método dos Elementos Finitos

Os problemas de engenharia desafiam engenheiros a trabalhar com projetos mais caros, que requerem maior confiabilidade, segurança e restrições rigorosas. Para garantir tais características, modelos matemáticos devem ser elaborados para descrever o comportamento requerido em serviço. Em geral, o comportamento desses problemas é governado por Equações Diferenciais Parciais (EDP) e a solução dessas por cálculo analítico é difícil, ou até mesmo insolúvel com as ferramentas disponíveis. Para contornar essa situação, ferramentas numéricas foram desenvolvidas com o propósito de solucionar essas equações. Nesse sentido, o MEF atende a maioria dos problemas, podendo ser aplicado em diversas situações como problemas que possuem regiões lineares ou não-lineares com domínio uni, bi ou tridimensional (DHATT; LEFRANÇOIS; TOUZOT, 2012).

O MEF trabalha com o princípio de aproximar uma geometria, de domínio contínuo  $\Omega$ , por um número finito de elementos conectados por nós (FISH; BELYTSCHKO, 2007). Na Figura 2.1, por exemplo, a placa fina com um furo centralizado submetida a um carregamento axial, tem o domínio aproximado por elementos triangulares conforme Figura 2.2a. A caracterização de um domínio contínuo por elementos finitos é denominado malha.

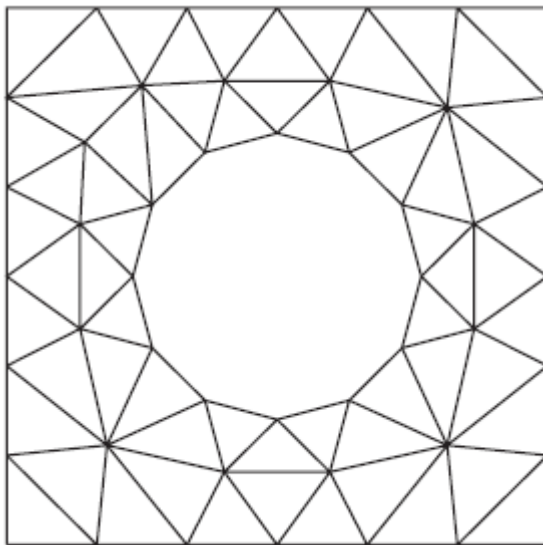
Observa-se que a descrição da geometria pela Figura 2.2a apresenta um erro de aproximação geométrica e, a medida que se aumenta o número de elementos, com o refino da malha especialmente na região circular, Figura 2.2b, o erro de aproximação geométrica é reduzido.

**Figura 2.1 – Placa retangular com furo central com carga axial**

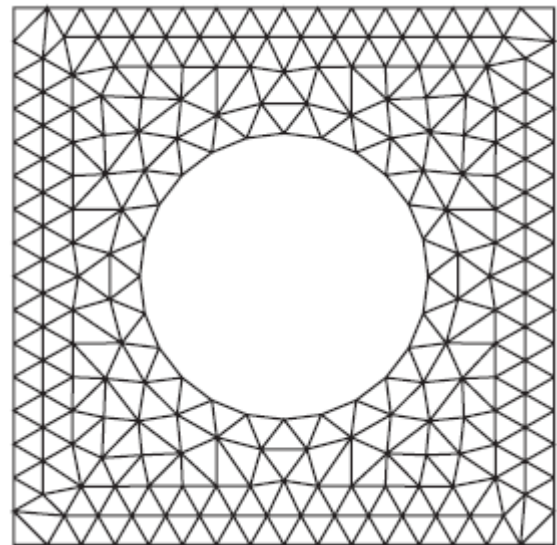
Fonte – Autoria Própria

**Figura 2.2 – Geometria aproximada por elementos triangulares**

(a) Malha inicial



(b) Malha refinada



Fonte – Fish e Belytschko (2007)

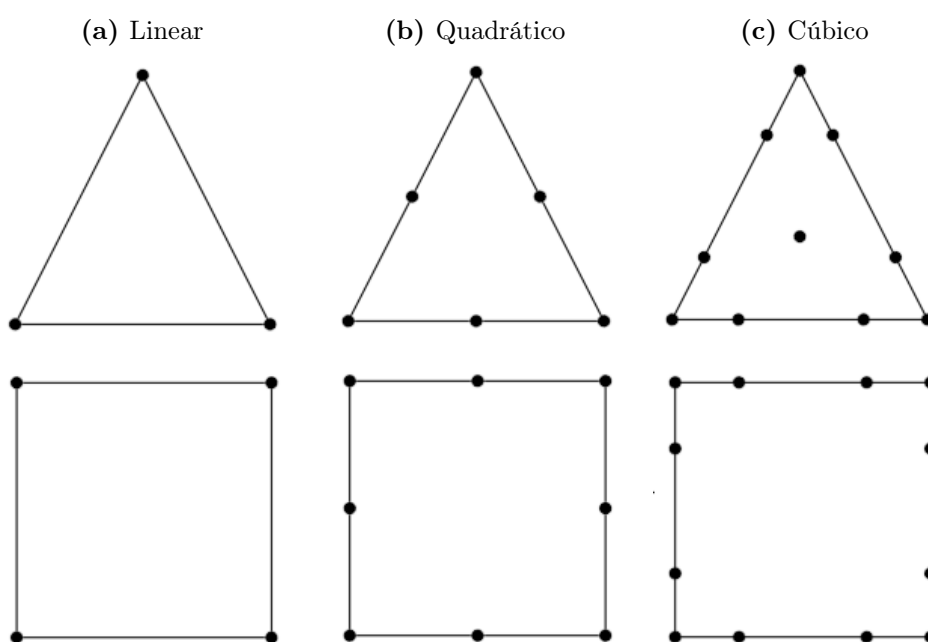
Outros modos de garantir a acurácia dos resultados são as escolhas dos elementos de aproximação conveniente para a geometria, a exemplo de elementos triangulares para contornos curvos, assim como o grau  $P$  das funções de interpolação, tratadas nas seções seguintes.

## 2.2 Os Tipos de Elementos

Durante a discretização de estruturas, o engenheiro também deve se preocupar com o tipo de elemento empregado, pois eles também afetam a acuracidade da solução tanto quanto a eficiência computacional do algoritmo (CHANDRUPATLA et al., 2002).

Para a discretização bidimensional, a literatura apresenta, majoritariamente, resultados para elementos triangulares e quadrilaterais, conforme Figura 2.3.

**Figura 2.3 – Representação geométrica por diversos gêneros de elementos não-estrutturados (triângulos) e elementos estruturados (quadriláteros)**



Fonte – Adaptado de Dhatt, Lefrançois e Touzot (2012)

Em geral, os elementos triangulares lineares são mais recorrentes na geração de malha, devido a facilidade de descrever qualquer geometria, até mesmo em topologias com contornos curvos, através do refino da malha a partir do aumento de elementos. Entretanto, é intrínseco dos elementos triangulares a baixa acurácia em relação aos resultados (FISH; BELYTSCHKO, 2007).

Por outro lado, os elementos quadrilaterais possuem maior precisão dos resultados, porém, requerem maiores tempos computacionais, devido ao aumento de incógnitas pelo acréscimo de um nó por elemento (FISH; BELYTSCHKO, 2007).

Na geração de malhas com elementos quadrangulares, deve-se evitar o surgimento de elementos degenerados, isto é, onde um nó colapsa superimpondo suas coordenadas a um outro nó, aproximando-no de um elemento triangular. As implicações desse efeito é impossibilidade do cálculo das soluções nesse ponto, necessitando de um remalhamento da geometria (FISH; BELYTSCHKO, 2007).

### 2.3 Os Elementos de Ordem Superior

Apesar da simplicidade da ideia de discretizar determinada geometria por uma malha, em alguns casos, onde a geometria possui contornos complexos, a aproximação pelos elementos lineares não é tão precisa. A fim de contornar tal situação, torna-se atrativo o uso de elementos de ordens mais altas para gerar a malha, ou seja, os elementos triangulares e quadrilaterais, com a adição de nós (físicos ou não), que podem ter seus contornos interpolados por funções de ordem superior - a exemplo de funções quadráticas e cúbicas conforme Figura 2.3 (FISH; BELYTSCHKO, 2007).

Segundo Reddy (1993), a discretização do domínio  $\Omega$  por um elemento qualquer, cada intervalo do elemento é descrito por uma função de forma polinomial  $N_i$ , cujo grau do polinômio é determinado pelo número  $(n - 1)$  de pontos do intervalo. Em elementos finitos de alta ordem, adiciona-se pontos dentro de cada intervalo afim de melhorar a aproximação. Em geral, utiliza-se a interpolação de Lagrange em elementos de alta ordem devido a característica do polinômio  $\mathcal{L}_i^n(x)$  adquirir o valor real quando aplicado no nó.

A adição de um número elevado de pontos no intervalo, intuitivamente, traria uma solução muito próxima da solução exata, contudo, a adição demasiada de pontos provocaria um comportamento indesejado no polinômio de Lagrange. Na faixa central do intervalo, o erro de aproximação seria baixo, entretanto, conforme se aproxima das extremidades, o polinômio interpolador se afasta do valores reais da EDP, aumentando seu erro. Esse comportamento é conhecido como Fenômeno Runge. Normalmente, opta-se por utilizar funções interpoladoras entre grau 5 e grau 9 (REDDY, 1993).

Na literatura, são propostas outras classes de funções de forma associadas aos entes topológicos dos elementos - tais como vértices e arestas -, a exemplo das funções de forma hierárquicas que utilizam os polinômios de Legendre e Chebyshev para casos unidimensionais e o uso do produto tensorial na construção das funções hierárquicas para os casos bi e tridimensionais (EDGAR; SURANA, 1996).

Bittencourt, Vazquez e Vazquez (2007) introduzem e desenvolvem as funções de forma que utilizam o produto tensorial entre funções unidimensionais de bases modais. Diferentemente das funções de forma já apresentadas, nesta situação, o acréscimo do número de graus de liberdade (GDL) ocorre sem a adição de nós físicos no elemento, isto é, as funções de interpolação não são associadas as variáveis nodais. Esse tratamento, é comum o uso dos polinômios ortogonais de Jacobi para construção das bases modais pois, desempenha a melhora da esparsidade das matrizes de rigidez e de massa conforme a escolha das funções pesos.

Desse modo, cabe ao projetista a escolha adequada das funções de forma para cada situação, uma vez que elas são essenciais para o desempenho do algoritmo e acuracidade da resposta.

## 2.4 A Forma Forte do Problema

Segundo Fish e Belytschko (2007), para utilizar o MEF é necessário desenvolver as equações governantes e as condições de contorno do problema, essas consideradas a forma forte do problema.

Em problemas bidimensionais, as equações governantes, em geral são EDP sobre regiões geométricas complexas, ou seja, os contornos  $\Gamma$  de um domínio bidimensional  $\Omega$ . Como o MEF visa aproximar o domínio  $\Omega$  por uma malha de elementos, a solução obtida contém erros devido a discretização do problema assim como pela aproximação do domínio.

Vários problemas de engenharia tem seu comportamento denotado pela Equação de Poisson,  $\nabla \cdot (k\nabla u) = f$  em  $\Omega$ , onde  $\nabla$  é o operador gradiente,  $k$  é uma constante do material e  $f$  é a variável de origem (REDDY, 1993).

Em problemas de elasticidade, a Equação de Poisson adota a forma da Equação 2.1, que é considerada a forma forte

$$K\nabla^2 \mathbf{u} = F \quad (2.1)$$

Onde  $K$  denota a rigidez do sistema,  $\mathbf{u} = \{ u \ v \}^T$  é campo de deslocamentos procurado tal que  $u$  e  $v$  são as componentes em  $x$  e  $y$ , respectivamente e,  $F$  representa as forças aplicadas no sistema.

## 2.5 O Estado Plano de Tensão

Na mecânica dos sólidos, considera-se que todas as estruturas mecânicas possuem domínio tridimensional, contudo, durante a formulação dos problemas, engenheiros podem optar pela simplificação das estruturas, de modo que elas possam ser representadas bidimensionalmente, ou até mesmo unidimensionalmente.

No estado geral de tensão, apresentado na Figura 2.4, o elemento de tensão é caracterizado por 3 componentes independentes de tensão normal  $\sigma_i$  e 3 componentes independentes de tensão de cisalhamento  $\tau_{ij}$  e é representado pela notação compacta do tensor de tensão 2<sup>a</sup> ordem descrito pela Equação 2.2 (HIBBELER, 2011) (CHANDRUPATLA et al., 2002).

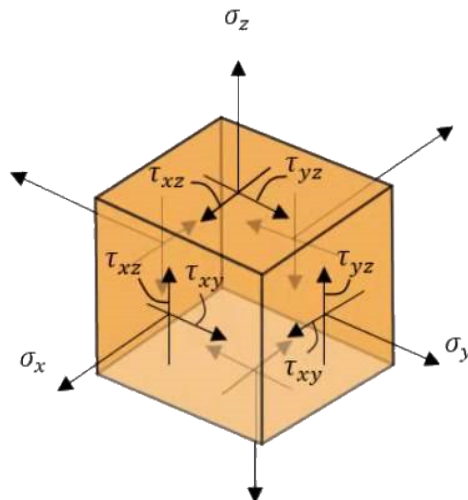
$$\boldsymbol{\sigma} = \left\{ \sigma_x \ \sigma_y \ \sigma_z \ \tau_{yz} \ \tau_{xz} \ \tau_{xy} \right\}^T \quad (2.2)$$

Quando conveniente, as estruturas podem ser analisadas em um único plano, assim, o estado geral de tensão é simplificado para um EPT (HIBBELER, 2011).

Portanto, o EPT para um elemento infinitesimal é descrito pela Equação 2.3.

$$\sigma_z = \tau_{xz} = \tau_{yz} = 0 \quad (2.3)$$

**Figura 2.4 – Elemento Infinitesimal em um Estado Triaxial de tensão.**

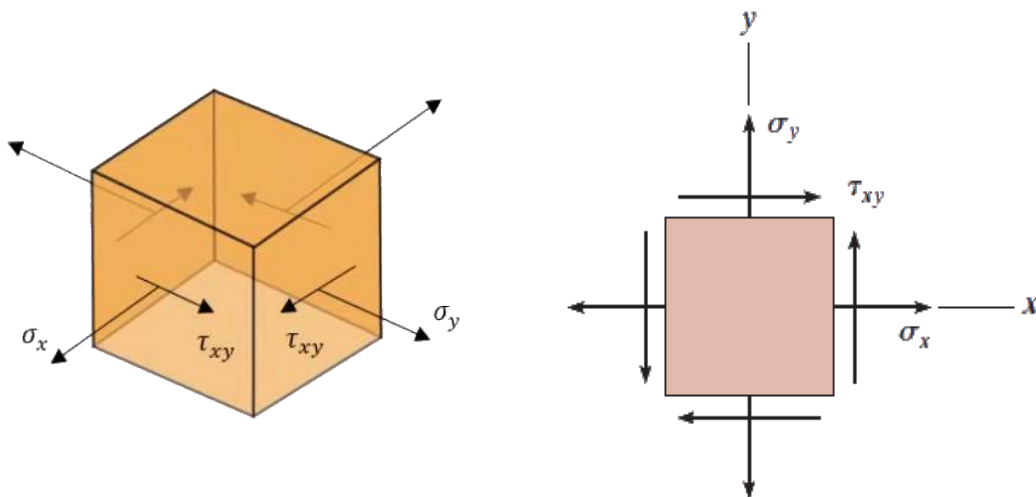


Fonte – Hibbeler (2011)

E as demais componentes de tensão,  $\sigma_x$ ,  $\sigma_y$  e  $\tau_{xy}$  pode-se atribuir valores não-nulos, como representado pelas Figuras 2.5a e 2.5b.

**Figura 2.5 – Representação do Estado Plano de Tensão**

- (a) Elemento Infinitesimal em Estado Plano de Tensão. (b) Representação Bidimensional do Estado Plano de Tensão.



Fonte – Hibbeler (2011)

Nessa condição, vigas com forças aplicadas transversalmente, vasos de pressão com paredes finas, cilindros pressurizados com paredes grossas abertos nas extremidades, e placas finas com carregamento distribuído ao longo da espessura exemplificam o EPT.

### 2.5.1 As Hipóteses do Estado Plano de Tensão

Como explanado na seção 2.5, durante a formulação e análise de diversos problemas estruturais, algumas simplificações são realizadas para reescrevê-lo bidimensionalmente. Contudo, para que o corpo esteja submetido ao EPT, as seguintes hipóteses devem ser satisfeitas:

- Não pode existir carregamentos na direção perpendicular ao plano, sendo a direção  $z$  na Figura 2.5a;
- As dimensões do plano em análise devem ser, no mínimo, 10 vezes superior a sua espessura.

### 2.5.2 As Equações de Equilíbrio

Admitindo a hipótese que os materiais submetidos ao EPT são isotrópicos, homogêneos e possuem comportamento elástico-linear, livros-textos como Hibbeler (2011) desenvolvem os conceitos de tensão e deformação, e formulam-nos matematicamente e seus resultados serão utilizados nesse trabalho.

Ao aplicar os resultados da equação 2.3 na lei de Hooke generalizada para um elemento tridimensional, as relações de deformação-tensão obtidas são expressas pela equação 2.4.

$$\begin{aligned}
 \varepsilon_x &= \frac{1}{E} (\sigma_x - \nu \sigma_y) \\
 \varepsilon_y &= \frac{1}{E} (\sigma_y - \nu \sigma_x) \\
 \gamma_{xy} &= \frac{\tau_{xy}}{G} \\
 \gamma_{xz} &= \gamma_{yz} = 0 \\
 \varepsilon_z &= -\frac{\nu}{E} (\sigma_x + \sigma_y)
 \end{aligned} \tag{2.4}$$

Onde

$$G = \frac{E}{2(1 + \nu)}$$

é o módulo de elasticidade transversal e  $\nu$  é coeficiente de Poisson.

Trabalhando os resultados da equação 2.4, obtém-se a relação inversa de tensão dada por  $\boldsymbol{\sigma} = D\boldsymbol{\varepsilon}$  (CHANDRUPATLA et al., 2002) e na forma matricial apresentada por 2.5.

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = \frac{E}{1 - \nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} \tag{2.5}$$

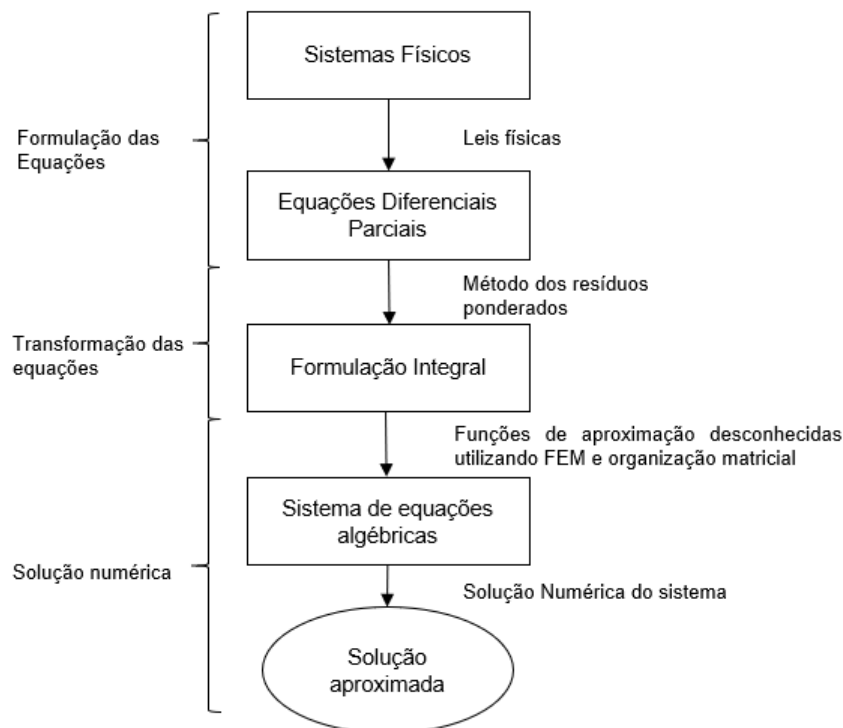


## 2.6 A Forma Fraca do Problema

Ao empregar o MEF como ferramenta de solução, Dhatt, Lefrançois e Touzot (2012) citam que o método visa discretizar a formulação integral (ou variacional) em uma série de equações algébricas que resolvidas numericamente se obtém uma solução aproximada. O fluxo desse procedimento é demonstrado na Figura 2.6.

Nesse sentido, obter a forma fraca do problema de elasticidade plana, Equação 2.1, é essencial para aplicar o método e na literatura, o método de Garlekin, ou Resíduos Ponderados, pode ser utilizado para obtenção da forma variacional da forma forte.

**Figura 2.6 – Transformações das equações de um sistema físico**



Fonte – Adaptado de Dhatt, Lefrançois e Touzot (2012)

Ao considerar que a discretização do domínio pelo MEF traz consigo os erros de aproximação, deseja-se que a diferença (resíduo) entre a solução exata  $u$  e a solução aproximada  $\tilde{u}$  seja zero ou satisfatoriamente próxima de zero, utiliza-se o método dos resíduos ponderados como mostrado na Equação 2.6.

$$\int_{\Omega} W_i (K\tilde{u} - F) d\Omega = 0 \quad i = 1 \text{ até } n \quad (2.6)$$

Onde  $W_i$  são as funções peso, linearmente independentes e integráveis no domínio  $\Omega$ .

Em Chandrupatla et al. (2002), a Aproximação de Garlekin consiste em um dos método de resíduos ponderados em que as funções pesos são construídas de acordo com as

funções da base  $N_i$  utilizadas para encontrar a solução aproximada  $\tilde{u}$ , conforme Equação 2.7, o que garante uma redução da ordem das derivadas e a formulação fraca do problema.

$$W = \phi = \sum_{i=1}^n \Psi_i N_i \quad (2.7)$$

A forma fraca em um problema de elementos finitos consiste em reescrever a EDP que governa o sistema na forma integral.

No livro texto de Chandrupatla et al. (2002), é desenvolvido matematicamente a formulação fraca para problemas de elasticidade a partir das equações de equilíbrio, o que foge do escopo desse trabalho, sendo omitido, utilizando apenas o resultado da forma fraca para problemas bidimensionais representado pela Equação 2.8.

$$\int_{\Omega} \sigma^T \varepsilon(\phi) t d\Omega - \left( \int_{\Omega} \phi^T \mathbf{f} t d\Omega + \int_{\Gamma} \phi^T \mathbf{T} t d\Gamma + \sum_i \phi_i \mathbf{P}_i \right) = 0 \quad (2.8)$$

A Equação 2.8, quando aplicado em problemas lineares elásticos é precisamente o Princípio dos Trabalhos Virtuais (PTV): o primeiro termo da equação representa o trabalho virtual interno do sistema, onde  $\sigma^T$  denota o tensor de tensão de 2ª ordem em notação compacta,  $\varepsilon(\phi)$  representa a deformação específica aplicado em um vetor arbitrário de deslocamento virtual  $\phi$  condizente com as condições de contorno e  $t$  a espessura da geometria; e o termo entre parênteses representa o trabalho virtual externo, onde  $f$  representa o vetor de forças de corpo,  $T$  é a força de tração representando eventuais carregamentos distribuídos na superfícies do corpo e  $P_i$  representa cargas pontuais aplicadas em um nó.

Para uma região bidimensional discretizada, a forma fraca é dada pela Equação 2.9.

$$\sum_e \int_e \varepsilon^T \mathbf{D} \varepsilon(\phi) t d\Omega - \left( \sum_e \int_e \phi^T \mathbf{f} t d\Omega + \int_{\Gamma} \phi^T \mathbf{T} t d\Gamma + \sum_i \phi_i^T \mathbf{P}_i \right) = 0 \quad (2.9)$$

# Capítulo 3

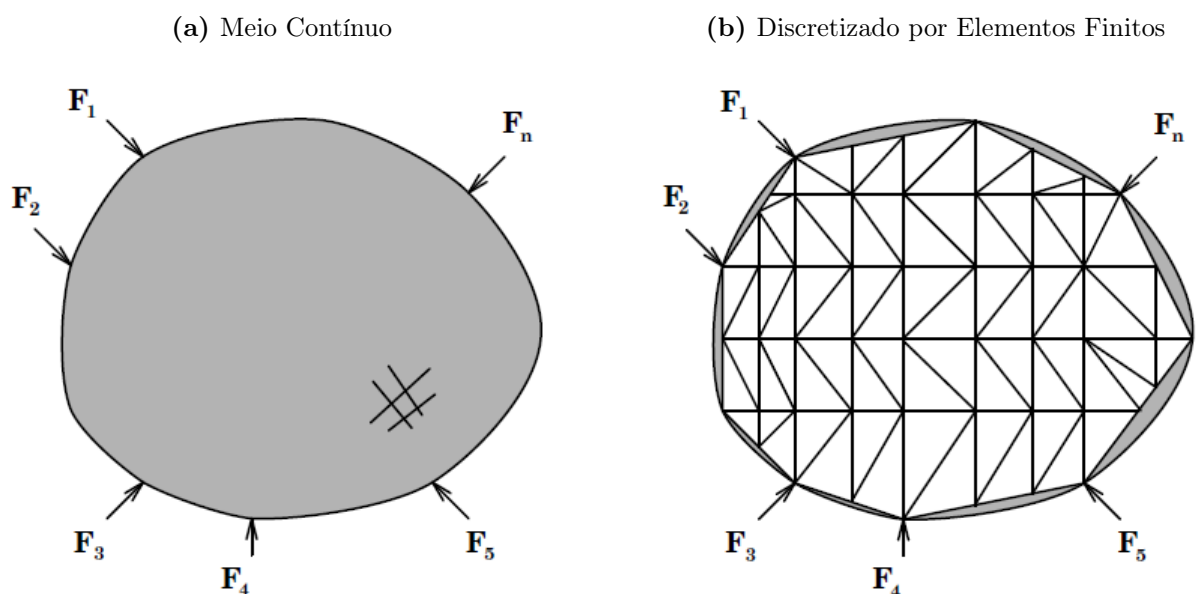
## Aspectos Metodológicos

Conforme Bittencourt (2007), em problemas estruturais, o MEF visa dividir o domínio do corpo contínuo, Figura 3.1a, em elementos finitos, Figura 3.1b, e obter a solução da Equação 2.1 nos pontos de interesse e nas demais posições, os deslocamentos são determinados por interpolação das funções de forma.

Para problemas em regime elástico no meio contínuo, o propósito da Equação 2.1 é encontrar o campo de deslocamentos  $u = \{ u \ v \}^T$ . Entretanto, como citado na Seção 2.6, não é possível resolver diretamente a equação com o MEF na forma diferencial, necessitando obter a forma integral da Equação 2.1, que é dada pela Equação 2.9 já discretizada.

Segundo Chandrupatla et al. (2002), a forma fraca para problemas elásticos é o

**Figura 3.1 – Sistema considerado como meio contínuo e sua discretização por elementos finitos**



Fonte – Bittencourt (2007)

PTV, cujo o primeiro termo da Equação 2.9 representa o trabalho virtual interno e, o termo entre parênteses o trabalho virtual externo, conforme as Equações 3.1 e 3.2.

$$\text{Trabalho Virtual Interno} = \Psi^T[\mathbf{K}]\{\mathbf{u}\} \quad (3.1)$$

$$\text{Trabalho Virtual Externo} = \Psi^T\{\mathbf{F}\} \quad (3.2)$$

Dessa maneira, nota-se que existe uma relação entre a rigidez, o carregamento e os deslocamentos nodais do sistema, ao substituir as Equações 3.1 e 3.2 em 2.9, representado pela Equação 3.3.

$$\{\mathbf{u}\} = [\mathbf{K}]^{-1}\{\mathbf{F}\} \quad (3.3)$$

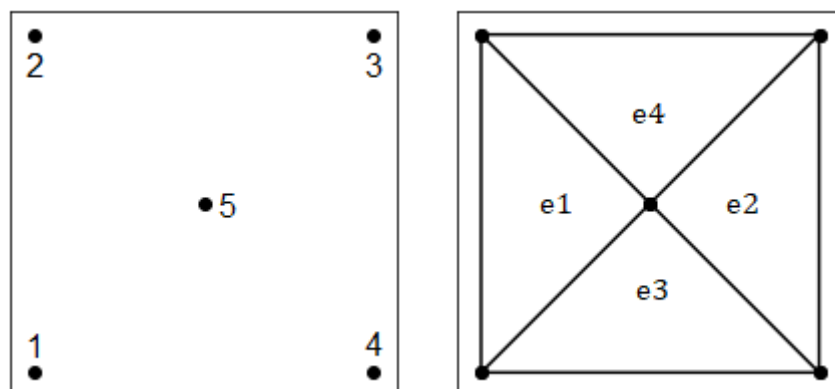
A Equação 3.3 fornece a solução para o modelo discretizado da Equação 2.1. Assim, a acuracidade dos resultados do MEF reside no procedimento de modelagem e nas estratégias de solução.

### 3.1 A Geração da Malha

Nesse estudo, para realizar a discretização das geometrias, utilizar-se-á a biblioteca *Triangle*, disponível para a linguagem Python (RUFAT, 2009). A escolha dessa biblioteca se dá pelo fato que, diferentemente de softwares comerciais, a malha não é limitada por um número máximo de elementos, sendo possível refiná-la até o ponto que exceda o limite da precisão computacional.

**Figura 3.2 – Geometria discretizada por meio da biblioteca *Triangle***

- (a) Inserção dos pontos de interesse da geometria  
 (b) Malha obtida por meio da triangulação de Delaunay



Fonte – Autoria Própria

A biblioteca *Triangle* permite por meio da inserção dos pontos de interesse da geometria a geração de uma malha triangular, baseado na triangulação de Delaunay, isto

é, para cada triângulo da malha, existe uma circunferência que contém unicamente os seus vértices (RUFAT, 2009).

Por exemplo, uma geometria quadrangular de lado unitário, insere-se os vértices da figura assim como um ponto central, com as coordenadas descritas na Tabela 3.1 e, como resultado, tem-se a geração da malha, conforme Figura 3.2.

**Tabela 3.1 – Coordenadas nodais da geometria**

Número do Nó	Coordenada x	Coordenada y
1	0,0	0,0
2	0,0	1,0
3	1,0	1,0
4	1,0	0,0
5	0,5	0,5

**Fonte – Autoria Própria**

Além da malha, a biblioteca provê a tabela conectividade dos elementos, Tabela 3.2, essencial para encontrar a correspondência entre a numeração global e local e os graus de liberdade da geometria (CHANDRUPATLA et al., 2002). Destaca-se que os nós  $a$ ,  $b$  e  $c$  da tabela conectividade são relativos a numeração local do elemento e podem diferir da numeração global.

**Tabela 3.2 – Tabela de conectividade dos elementos**

Elemento	Identificação do nó		
	Nó a	Nó b	Nó c
e1	2	1	5
e2	5	4	3
e3	4	5	1
e4	5	3	2

**Fonte – Autoria Própria**

Deve-se salientar que ao passo que a geometria adquire contornos mais complexos, adota-se a estratégia de adicionar mais pontos para suavizar o contorno e melhorar a aproximação da geometria

Após a obtenção da malha, Chandrupatla et al. (2002) discute que o MEF usufrui do conceito de funções de forma para determinar os deslocamentos nodais de cada elemento. A seguir é discutido a construção das funções de forma para elementos triangulares.

### 3.2 As Funções de Forma

Segundo Bittencourt (2014), para elementos triangulares emprega-se as coordenadas naturais de área na construção das funções de forma devido a propriedade de simetria rotacional.

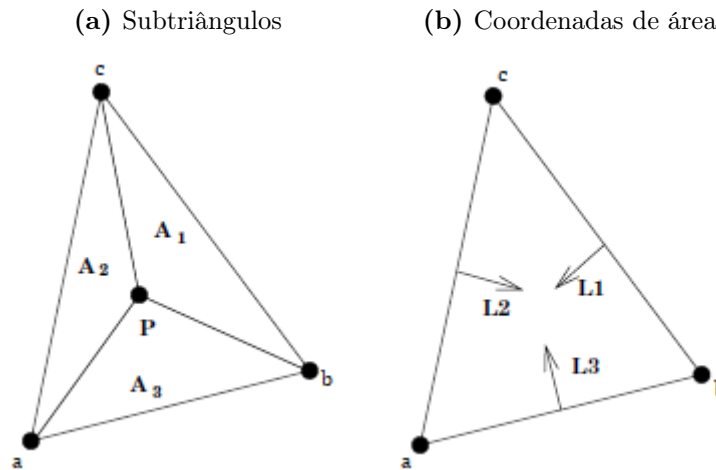
Ainda de acordo com o autor, subdividindo um triângulo de área  $A$  em três subtriângulos de área  $A_1$ ,  $A_2$  e  $A_3$ , as coordenadas de área  $0 \leq L_i = \frac{A_i}{A} \leq 1$ , Figura 3.3,

são definidas como a razão entre as áreas dos subtriângulos e a área original, conforme a Equação 3.4.

$$\frac{A_1}{A} + \frac{A_2}{A} + \frac{A_3}{A} = 1 \rightarrow L_1 + L_2 + L_3 = 1 \quad (3.4)$$

Tendo em vista que uma das coordenadas de área é dependente das demais, por exemplo, pode-se escrever  $L_3$  em função de  $L_1$  e  $L_2$ .

**Figura 3.3 – Coordenadas naturais baricêntricas para o triângulo linear**



Fonte – Bittencourt (2007)

O procedimento adotado nesse estudo emprega o uso de produto tensorial entre polinômios unidimensionais expressos em coordenadas naturais para a obtenção das funções de forma e, a expressão geral das funções de forma para triângulos é apresentada pela Equação 3.5

$$\phi_{pqr}^{(e)}(L_1, L_2, L_3) = \phi_p^{(e)}(L_1) \phi_q^{(e)}(L_2) \phi_r^{(e)}(L_3) \quad (3.5)$$

Onde  $\phi_p$ ,  $\phi_q$  e  $\phi_r$  são funções unidimensionais nas direções, respectivamente, de  $L_1$ ,  $L_2$  e  $L_3$  e,  $p$ ,  $q$  e  $r$  são os índices de tensorização associados as entidades topológicas (nós e arestas) do triângulo (BITTENCOURT, 2014).

Para a construção das funções de forma de grau superior a 1, utilizar-se-á as bases modais, ou seja, que não dependem das coordenadas das variáveis nodais em sua construção (BITTENCOURT; VAZQUEZ; VAZQUEZ, 2007). Dessa forma, os polinômios empregados devem atender tal condição e, nesse estudo, são considerados os polinômios ortogonais de Jacobi.

Dada a ordem  $P$  de interpolação, por exemplo, para a coordenada natural  $L_1$ , os polinômios da base podem ser definidos pela Equação 3.6, onde  $P_{p-1}^{\alpha_1, \beta_1}$  são os polinômios

ortogonais de Jacobi,  $\alpha_i$  e  $\beta_i$  são os pesos dos polinômios ortogonais e  $p$  são os índices de tensorização.

$$\phi_p^{(e)}(L_1) = \begin{cases} 1 & p = 0 \\ L_1 & p = P \\ L_1 P_{p-1}^{\alpha_1, \beta_1} (2L_1 - 1) & 0 < p < P \end{cases} \quad (3.6)$$

Analogamente, para as demais direções, a definição apresentada pela Equação 3.6 também pode ser usada.

A opção de usar bases modais é em razão da dificuldade ou até mesmo a impossibilidade de gerar malhas com nós físicos nas arestas e faces dos elementos para ordens mais altas, dificultando o uso de bases nodais. Apesar da existência de técnicas de colocação de nós em malhas lineares baseados em quadraturas de integração que contornam a dificuldade do uso de bases nodais, há outra vantagem no uso dos polinômios ortogonais de Jacobi que é proporcionada pela melhora da esparsidade das matrizes de rigidez e de massa (BITTENCOURT; VAZQUEZ; VAZQUEZ, 2007).

Ainda conforme o autor, apesar do uso das bases modais para a construção das funções de forma com graus de interpolação superiores a 1, as funções de vértices, nessa abordagem, dependem das coordenadas nodais.

Assim, utilizando da expressão geral das funções de forma, Equação 3.5, e as definições das bases modais, obtém-se as seguintes expressões:

- As funções de forma de vértice:

$$\begin{aligned} \phi_{P00}^{(e)}(L_1, L_2, L_3) &= \phi_P^{(e)}(L_1) \phi_0^{(e)}(L_2) \phi_0^{(e)}(L_3) = L_1 \\ \phi_{0P0}^{(e)}(L_1, L_2, L_3) &= \phi_0^{(e)}(L_1) \phi_P^{(e)}(L_2) \phi_0^{(e)}(L_3) = L_2 \\ \phi_{00P}^{(e)}(L_1, L_2, L_3) &= \phi_0^{(e)}(L_1) \phi_0^{(e)}(L_2) \phi_P^{(e)}(L_3) = L_3 \end{aligned} \quad (3.7)$$

- As funções de forma de aresta:

$$\begin{aligned} \phi_{pq0}^{(e)}(L_1, L_2, L_3) &= \phi_p^{(e)}(L_1) \phi_q^{(e)}(L_2) \phi_0^{(e)}(L_3) \\ &= L_1 L_2 P_{p-1}^{\alpha_1, \beta_1} (2L_1 - 1) P_{q-1}^{\alpha_2, \beta_2} (2L_2 - 1), \quad (p + q = P) \\ \phi_{p0r}^{(e)}(L_1, L_2, L_3) &= \phi_p^{(e)}(L_1) \phi_0^{(e)}(L_2) \phi_r^{(e)}(L_3) \\ &= L_1 L_3 P_{p-1}^{\alpha_1, \beta_1} (2L_1 - 1) P_{r-1}^{\alpha_3, \beta_3} (2L_3 - 1), \quad (p + r = P) \\ \phi_{0qr}^{(e)}(L_1, L_2, L_3) &= \phi_0^{(e)}(L_1) \phi_q^{(e)}(L_2) \phi_r^{(e)}(L_3) \\ &= L_2 L_3 P_{q-1}^{\alpha_2, \beta_2} (2L_2 - 1) P_{r-1}^{\alpha_3, \beta_3} (2L_3 - 1), \quad (q + r = P) \end{aligned} \quad (3.8)$$

- As funções de forma de face:

$$\begin{aligned} \phi_{pqr}^{(e)}(L_1, L_2, L_3) &= \phi_p^{(e)}(L_1) \phi_q^{(e)}(L_2) \phi_r^{(e)}(L_3) \\ &= L_1 L_2 L_3 P_{p-1}^{\alpha_1, \beta_1} (2L_1 - 1) P_{q-1}^{\alpha_2, \beta_2} (2L_2 - 1) P_{r-1}^{\alpha_3, \beta_3} (2L_3 - 1) \end{aligned} \quad (3.9)$$

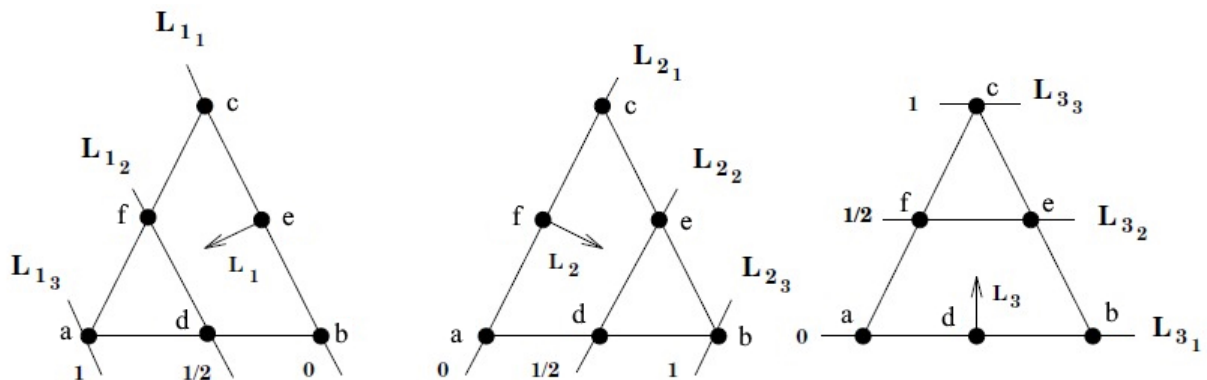
### 3.2.1 Índices de tensorização

A dedução das funções de forma para o elemento triangular plano é dado pelo produto tensorial dos polinômios unidimensionais de Jacobi conforme a Equação 3.5. Para isso, os índices de tensorização  $p$ ,  $q$  e  $r$  devem corresponder adequadamente as posições das funções unidimensionais na base de suas respectivas coordenadas naturais  $L_1$ ,  $L_2$  e  $L_3$ .

Neste estudo, utilizar-se-á as funções de forma de alta ordem de grau 2 e 4 em comparativo às funções lineares e a abordagem utilizada para a construção dos índices é a mesma empregada por Bittencourt (2007)

Segundo o autor, cada direção  $L_i$  possuirá linhas constantes que assumem valores entre 1 e  $P + 1$  e que estão localizadas entre 0 e 100% do comprimento da coordenada natural em análise. Dessa forma, os índices  $p$ ,  $q$  e  $r$  de um nó são determinados pelos valores assumidos pelas linhas constantes que o intersectam, independente da ordem de interpolação. Por exemplo, o elemento triangular plano quadrático, Figura 3.4, o nó  $a$  é determinado pelas linhas de valores constante 3, 1 e 1 respectivamente, das direções  $L_1$ ,  $L_2$  e  $L_3$ . Em geral, o valor das linhas constantes é representado pelo subíndice  $n$  da coordenada da coordenada natural,  $L_{in}$ .

Figura 3.4 – Elemento triangular plano quadrático



Fonte – Adaptado de Bittencourt (2007)

Para os demais nós do elemento triangular plano quadrático, os índices de tensorização estão dispostos na Tabela 3.3.

Tabela 3.3 – Índices de tensorização para o elemento triangular plano quadrático

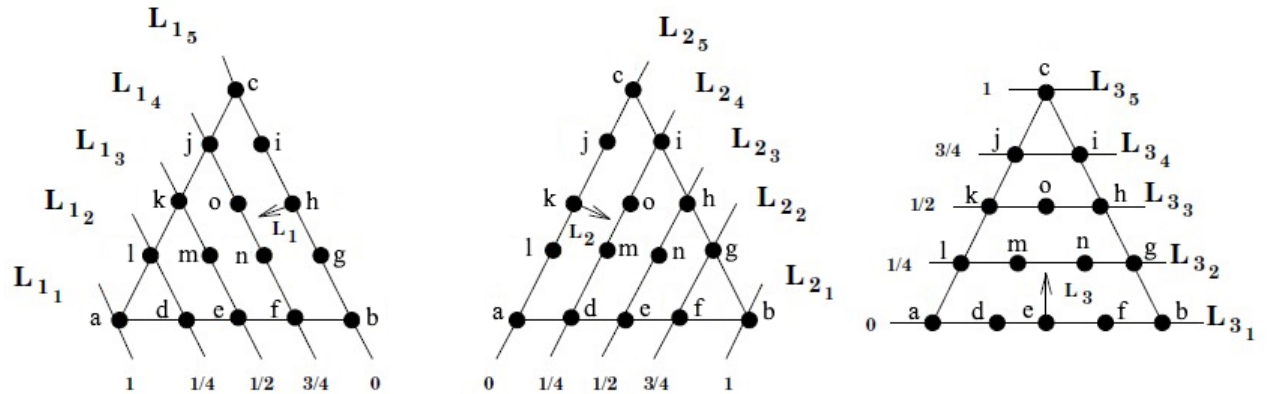
	a	b	c	d	e	f
p	3	1	1	2	1	2
q	1	3	1	2	2	1
r	1	1	3	1	2	2

Fonte – Adaptado de Bittencourt (2007)



Quanto as funções de forma de alta ordem de grau 4, além das funções de aresta, há a ocorrência das funções de face, apresentadas como nós na Figura 3.5.

Figura 3.5 – Elemento triangular plano quártico



Fonte – Adaptado de Bittencourt (2007)

Os índices de tensorização para as funções de forma quárticas são apresentadas na Tabela 3.4

Tabela 3.4 – Índices de tensorização para o elemento triangular plano quártico

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
p	3	1	1	4	3	2	1	1	1	2	3	4	3	2	2
q	1	3	1	2	2	1	4	3	2	1	1	1	2	3	2
r	1	1	3	1	2	2	2	3	4	4	3	2	2	2	3

Fonte – Adaptado de Bittencourt (2007)

### 3.3 Transformações Geométricas e os Sistemas de Coordenadas

Na literatura, as transformações geométricas são tratadas como essenciais para a obtenção do sistema de equações algébricas. Dada a necessidade de representar o domínio  $\Omega$ , as representações geométricas mais comuns dos elementos são:

- Subparamétricos: as transformações geométricas utilizam funções de interpolação de grau inferior às funções de interpolação dos deslocamentos.
- Isoparamétricos: as transformações geométricas utilizam funções de interpolação de grau idêntico às funções de interpolação dos deslocamentos.
- Superparamétricos: as transformações geométricas utilizam funções de interpolação de grau superior às funções de interpolação dos deslocamentos.

Nessas condições, é importante escolher adequadamente as transformações geométricas, pois elas permitem relacionar as coordenadas locais ( $L_1, L_2$ ) com as coordenadas globais ( $x, y$ ) através da matriz Jacobiana  $J$  para cada elemento.

De acordo com Bittencourt (2007), ao trabalhar com dois sistemas de referência, é necessário que existam transformações que relacionem o sistema global e o sistema local de referências, pois, facilita o mapeamento de elementos da malha e o cálculo de matrizes dos elementos finitos. Diante disso, as Equações 3.10 e 3.11 apresentam relações entre as coordenadas locais  $L_1$  e  $L_2$  e as coordenadas globais  $x$  e  $y$ .

$$x(L_1, L_2) = \sum_{a=1}^n X_a^e N_a(L_1, L_2) = \{X\}^T \{N\} \quad (3.10)$$

$$y(L_1, L_2) = \sum_{a=1}^n Y_a^e N_a(L_1, L_2) = \{Y\}^T \{N\} \quad (3.11)$$

Tais transformações são importantes durante o cálculo da matriz de rigidez e do vetor do carregamento do elemento, pois requerem as derivadas das funções de forma em relação as coordenadas globais  $x$  e  $y$  (BITTENCOURT, 2014).

Neste trabalho, utilizou-se as transformações subparamétricas para o mapeamento dos elementos triangulares, ou seja, o número de funções de interpolação é menor que o número total de graus de liberdade.

### 3.4 Matriz de Elasticidade e a Matriz Jacobiana

De acordo com Bittencourt (2014) e conforme apresentado na subseção 2.5.2, as relações de deformação são essenciais durante o processo de modelagem do MEF para problemas de EPT e são representadas como um conjunto de derivadas parciais do campo de deslocamento  $\{u\}$ , Equação 3.12.

$$\{\varepsilon\} = \{\varepsilon_x \quad \varepsilon_y \quad \gamma_{xy}\} = \left\{ \frac{\partial u}{\partial x} \quad \frac{\partial v}{\partial y} \quad \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \right\} \quad (3.12)$$

Na forma discretizada, as componentes dos campos de deslocamentos  $u$  e  $v$  são descritas por uma combinação linear entre os deslocamentos nodais  $u_i$  e  $v_i$  e suas respectivas funções de forma  $N_i$ . Ao realizar a diferenciação do campos de deslocamento para encontrar as componentes de deformação  $\{\varepsilon\}$ , destaca-se a matriz de elasticidade  $[B]$ , Equação 3.14.

$$\{\varepsilon\} = \left\{ \frac{\partial \{u\}[N]^T}{\partial x} \quad \frac{\partial \{v\}[N]^T}{\partial y} \quad \frac{\partial \{v\}[N]^T}{\partial x} + \frac{\partial \{u\}[N]^T}{\partial y} \right\} = [B]\{u\} \quad (3.13)$$

A matriz de elasticidade  $[B]$  trata-se de uma composição de matrizes  $B_i$  de derivadas parciais das funções de forma  $N_i$  em relação as direções  $x$  e  $y$ , conforme Equação 3.15. A finalidade dessa matriz é computar as deformações em qualquer ponto interno do elemento triangular em análise e como consequência, empregada também para o cálculo de tensões.

$$[B] = [B_1 B_2 \dots] \quad (3.14)$$

$$[B_i] = \begin{bmatrix} \frac{\partial N_i}{\partial x} & 0 \\ 0 & \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial y} & \frac{\partial N_i}{\partial x} \end{bmatrix} \quad (3.15)$$

No procedimento de obtenção da matriz  $[B]$ , observa-se que as funções de forma estão em função das coordenadas locais  $L_1$  e  $L_2$  e é necessário aplicar a regra da cadeia para se obter as derivadas globais, conforme mostrado em forma matricial na Equação 3.16, para um nó  $a$ .

$$\begin{Bmatrix} \frac{\partial N_a}{\partial L_1} \\ \frac{\partial N_a}{\partial L_2} \end{Bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial L_1} & \frac{\partial y}{\partial L_1} \\ \frac{\partial x}{\partial L_2} & \frac{\partial y}{\partial L_2} \end{bmatrix} \begin{Bmatrix} \frac{\partial N_a}{\partial x} \\ \frac{\partial N_a}{\partial y} \end{Bmatrix} \rightarrow \begin{Bmatrix} \frac{\partial N_a}{\partial x} \\ \frac{\partial N_a}{\partial y} \end{Bmatrix} = [J]^{-1} \begin{Bmatrix} \frac{\partial N_a}{\partial L_1} \\ \frac{\partial N_a}{\partial L_2} \end{Bmatrix} \quad (3.16)$$

Nesse sentido, a matriz  $[J]$ , Equação 3.17, que se desenvolve na Equação 3.16 é denominada matriz Jacobiana de transformação e essa relaciona as derivadas das funções de forma entre os sistemas de coordenadas local e global (BITTENCOURT, 2014).

$$[J] = \begin{bmatrix} \{x\}^T \frac{\partial \{N\}}{\partial L_1} & \{y\}^T \frac{\partial \{N\}}{\partial L_1} \\ \{x\}^T \frac{\partial \{N\}}{\partial L_2} & \{y\}^T \frac{\partial \{N\}}{\partial L_2} \end{bmatrix} \quad (3.17)$$

Vale salientar que a matriz Jacobiana de transformação desempenha outros papéis na abordagem de elementos finitos, por exemplo, o cálculo da área do elemento e a determinação de forças de corpo (CHANDRUPATLA et al., 2002).

### 3.5 As Matrizes Elementares

Como apresentado anteriormente na Equação 3.3, a solução do MEF para os deslocamentos nodais é obtida diretamente através do produto inversão da matriz de rigidez e do vetor de carregamento do sistema. A matriz de rigidez e o vetor de carregamento, nesse aspecto, são essenciais para obtenção da solução e, ao manipular a forma fraca discretizada, Equação 2.9, obtém-se a seguinte relação para a matriz rigidez (CHANDRUPATLA et al., 2002):

$$[K^e] = t_e \int_0^1 \int_0^{1-L_2} [B]^T [D] [B] |\det[J]| dL_1 dL_2 \quad (3.18)$$

Apesar do tratamento, a Equação 3.18 não é restrita a problemas de EPT, sendo aplicável em problemas de elementos finitos baseado em deslocamentos, desde que o tratamento seja adequado a análise empregada (COOK, 1994).

Ainda de acordo com Cook (1994), como a matriz constituinte  $[D]$  é constante, a matriz rigidez elementar  $[K^e]$  depende da diferenciação das funções de forma da matriz de elasticidade  $[B]$ . Isso implica que o comportamento e as limitações da modelagem de elementos finitos depende da escolha apropriada do tipo de elemento, das funções de forma e do refino da malha.

Por outro lado, a representação do vetor de carregamento pelo MEF é dada pela equação 3.19 (CHANDRUPATLA et al., 2002):

$$\{F^e\} = \begin{cases} t_e \left[ \int_0^1 \int_0^{1-L_2} \{N\}^T \det[J] dL_1 dL_2 \right] \begin{Bmatrix} f_x \\ f_y \end{Bmatrix} & (3.19a) \\ \int_{-1}^1 Nq(\xi) \det[J] d\xi & \text{onde } \xi = 2L - 1 & (3.19b) \\ \sum_i P_i & & (3.19c) \end{cases}$$

Dentre as cargas mecânicas que o compõe, incluem-se: as forças de corpo, Equação 3.19a, que atuam sobre o volume da estrutura; as forças de tração, Equação 3.19b, que são distribuídas ao longo da fronteira  $\Gamma$ ; e as forças concentradas, Equação 3.19c, que incluem momentos e cargas pontuais (COOK, 1994).

Tais relações de rigidez e carregamento são resolvidas elemento a elemento e, conseguinte, conforme a técnica de montagem empregada, se constrói a matriz global de rigidez e do vetor de carregamento.

### 3.6 Construção das Matrizes Globais

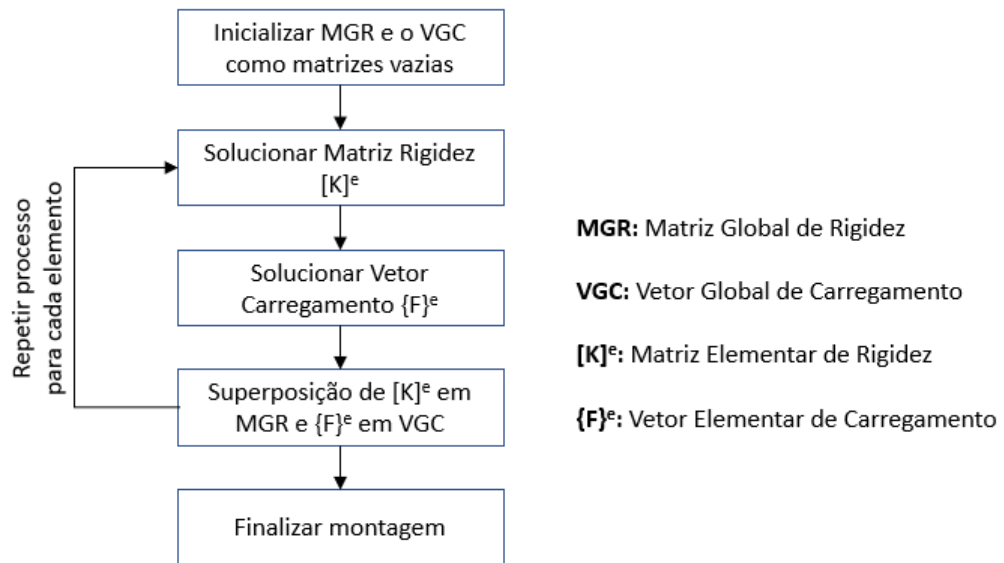
Conforme Dhatt, Lefrançois e Touzot (2012), em problemas de mecânica estrutural, a montagem das matrizes globais consiste em construir a matriz global de rigidez  $[K]$  e o vetor global de carregamento  $\{F\}$  através da superposição da parcela de contribuição na rigidez e no carregamento de cada elemento, respectivamente. A Figura 3.6 apresenta o fluxo esquemático da construção das matrizes globais.

O processo de montagem deve respeitar a continuidade dos deslocamentos e o equilíbrio de forças em cada nó de modo que resulte no sistema global de equações, e nesse sentido Dhatt, Lefrançois e Touzot (2012) enfatizam a necessidade da tabela de conectividade dos nós para a construção da tabela de localização dos elementos (LOCE) nas matrizes globais.

Neste estudo, empregou-se uma técnica de montagem denominada cardinalidade, baseada na numeração global dos nós e seus respectivos graus de liberdade. O método visa localizar a posição dos graus de liberdade elementares nas matrizes globais relacionando-os com a numeração global dos nós, de modo que os graus de liberdade na direção  $u$  e  $v$  são representados por  $2n - 1$  e  $2n$ , respectivamente, onde  $n$  é a numeração global do nó.

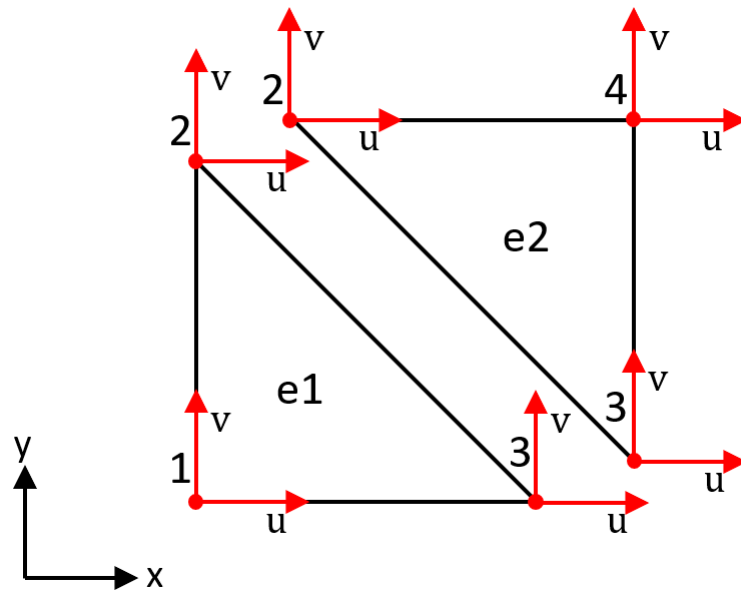
Por exemplo, considere a placa quadrada discretizada por 2 elementos lineares, Figura 3.7, e sua tabela conectividade, 3.5.

Figura 3.6 – Processo de montagem da Matriz Global de Rigidez e do Vetor Global de Carregamento



Fonte – Adaptado de Dhatt, Lefrançois e Touzot (2012)

Figura 3.7 – Placa quadrada discretizada por dois elementos. Os nós da geometria estão identificados pela numeração global



Fonte – Autoria Própria

**Tabela 3.5 – Tabela conectividade para o exemplo da 3.7. Utilizou-se a numeração global para realizar o mapeamento dos nós.**

Elemento	Número do nó		
	Nó a	Nó b	Nó c
e1	2	1	3
e2	3	4	2

Fonte – Autoria Própria

A partir da tabela de conectividade do problema, Tabela 3.5, e usando a regra de numeração dos GDL, obtém-se a tabela de localização dos elementos (LOCE) das matrizes locais nas matrizes globais, 3.6.

**Tabela 3.6 – Tabela LOCE na matriz global de rigidez e no vetor global de carregamento**

	Nó a		Nó b		Nó c	
	u	v	u	v	u	v
Elemento 1	3	4	1	2	5	6
Elemento 2	5	6	7	8	3	4

Fonte – Autoria Própria

Após resolver a Equação 3.18 para cada elemento, nomeia-se as linhas e as colunas das matrizes elementares de rigidez obtidas de acordo com a ordem da numeração obtida na tabela LOCE, conforme apresentado nas Tabelas 3.7 e 3.8.

**Tabela 3.7 – Matriz de rigidez do elemento 1 com índices de localização na matriz global de rigidez**

	3	4	1	2	5	6
3	$K_{11}^{(e1)}$	$K_{12}^{(e1)}$	$K_{13}^{(e1)}$	$K_{14}^{(e1)}$	$K_{15}^{(e1)}$	$K_{16}^{(e1)}$
4	$K_{21}^{(e1)}$	$K_{22}^{(e1)}$	$K_{23}^{(e1)}$	$K_{24}^{(e1)}$	$K_{25}^{(e1)}$	$K_{26}^{(e1)}$
1	$K_{31}^{(e1)}$	$K_{32}^{(e1)}$	$K_{33}^{(e1)}$	$K_{34}^{(e1)}$	$K_{35}^{(e1)}$	$K_{36}^{(e1)}$
2	$K_{41}^{(e1)}$	$K_{42}^{(e1)}$	$K_{43}^{(e1)}$	$K_{44}^{(e1)}$	$K_{45}^{(e1)}$	$K_{46}^{(e1)}$
5	$K_{51}^{(e1)}$	$K_{52}^{(e1)}$	$K_{53}^{(e1)}$	$K_{54}^{(e1)}$	$K_{55}^{(e1)}$	$K_{56}^{(e1)}$
6	$K_{61}^{(e1)}$	$K_{62}^{(e1)}$	$K_{63}^{(e1)}$	$K_{64}^{(e1)}$	$K_{65}^{(e1)}$	$K_{66}^{(e1)}$

Fonte – Autoria Própria

**Tabela 3.8 – Matriz de rigidez do elemento 2 com índices de localização na matriz global de rigidez**

	5	6	7	8	3	4
5	$K_{11}^{(e2)}$	$K_{12}^{(e2)}$	$K_{13}^{(e2)}$	$K_{14}^{(e2)}$	$K_{15}^{(e2)}$	$K_{16}^{(e2)}$
6	$K_{21}^{(e2)}$	$K_{22}^{(e2)}$	$K_{23}^{(e2)}$	$K_{24}^{(e2)}$	$K_{25}^{(e2)}$	$K_{26}^{(e2)}$
7	$K_{31}^{(e2)}$	$K_{32}^{(e2)}$	$K_{33}^{(e2)}$	$K_{34}^{(e2)}$	$K_{35}^{(e2)}$	$K_{36}^{(e2)}$
8	$K_{41}^{(e2)}$	$K_{42}^{(e2)}$	$K_{43}^{(e2)}$	$K_{44}^{(e2)}$	$K_{45}^{(e2)}$	$K_{46}^{(e2)}$
3	$K_{51}^{(e2)}$	$K_{52}^{(e2)}$	$K_{53}^{(e2)}$	$K_{54}^{(e2)}$	$K_{55}^{(e2)}$	$K_{56}^{(e2)}$
4	$K_{61}^{(e2)}$	$K_{62}^{(e2)}$	$K_{63}^{(e2)}$	$K_{64}^{(e2)}$	$K_{65}^{(e2)}$	$K_{66}^{(e2)}$

Fonte – Autoria Própria

Nesse sentido, cada coeficiente  $K_{ij}^{(e)}$  tem sua posição  $K_{ij}$  na matriz global definido pelo número da linha e da coluna que receberam da tabela LOCE, por exemplo, o coeficiente  $K_{43}^{(e1)}$  tem a posição na matriz global na linha  $i = 2$  e na coluna  $j = 1$  mostrado na Equação 3.20.

$$[K] = \begin{bmatrix}
 K_{33}^{(e1)} & K_{34}^{(e1)} & K_{31}^{(e1)} & K_{32}^{(e1)} & K_{35}^{(e1)} & K_{36}^{(e1)} & 0 & 0 \\
 K_{43}^{(e1)} & K_{44}^{(e1)} & K_{41}^{(e1)} & K_{42}^{(e1)} & K_{45}^{(e1)} & K_{46}^{(e1)} & 0 & 0 \\
 K_{13}^{(e1)} & K_{14}^{(e1)} & K_{11}^{(e1)} + K_{55}^{(e2)} & K_{12}^{(e1)} + K_{56}^{(e2)} & K_{15}^{(e1)} + K_{51}^{(e2)} & K_{16}^{(e1)} + K_{52}^{(e2)} & K_{53}^{(e2)} & K_{54}^{(e2)} \\
 K_{23}^{(e1)} & K_{24}^{(e1)} & K_{21}^{(e1)} + K_{65}^{(e2)} & K_{22}^{(e1)} + K_{66}^{(e2)} & K_{25}^{(e1)} + K_{61}^{(e2)} & K_{26}^{(e1)} + K_{62}^{(e2)} & K_{63}^{(e2)} & K_{64}^{(e2)} \\
 K_{53}^{(e1)} & K_{54}^{(e1)} & K_{51}^{(e1)} + K_{15}^{(e2)} & K_{52}^{(e1)} + K_{16}^{(e2)} & K_{55}^{(e1)} + K_{11}^{(e2)} & K_{56}^{(e1)} + K_{12}^{(e2)} & K_{13}^{(e2)} & K_{14}^{(e2)} \\
 K_{63}^{(e1)} & K_{64}^{(e1)} & K_{61}^{(e1)} + K_{25}^{(e2)} & K_{62}^{(e1)} + K_{26}^{(e2)} & K_{65}^{(e1)} + K_{21}^{(e2)} & K_{66}^{(e1)} + K_{22}^{(e2)} & K_{23}^{(e2)} & K_{24}^{(e2)} \\
 0 & 0 & K_{35}^{(e2)} & K_{36}^{(e2)} & K_{31}^{(e2)} & K_{32}^{(e2)} & K_{33}^{(e2)} & K_{34}^{(e2)} \\
 0 & 0 & K_{45}^{(e2)} & K_{46}^{(e2)} & K_{24}^{(e2)} & K_{24}^{(e2)} & K_{24}^{(e2)} & K_{24}^{(e2)}
 \end{bmatrix} \quad (3.20)$$

Por outro lado, uma posição  $K_{ij}$  na matriz global pode ser ocupada por diversos coeficientes das matrizes locais de rigidez, a exemplo dos coeficientes  $K_{25}^{(e1)}$  e  $K_{61}^{(e2)}$  que tem seus efeitos superimpostos na posição  $K_{45}$ .

Apesar do exemplo acima se tratar da construção da matriz global de rigidez, o conceito pode ser expandido para o vetor global de carregamento atentando-se durante o processo de identificação dos GDL do vetor elementar de carregamento apenas nomear suas linhas.

Vale salientar que em elementos de alta ordem, a tabela LOCE é construída de mesmo modo, mas como os nós de arestas e nós de faces não são físicos, são numerados posteriormente a identificação completa dos nós de vértice.

### 3.6.1 Condições de contorno

Em problemas de mecânica estrutural, em geral, as estruturas são submetidas as restrições físicas que vinculam seus contornos ou fronteiras a condições de deslocamento

conhecidas. Na mecânica dos sólidos, algumas das condições de contorno mais utilizadas para problemas de EPT são as que restringem movimento em uma ou duas direções, ou seja, podem restringir o movimento na direção  $u$  e/ou  $v$  (HIBBELER, 2011).

No método dos elementos finitos, conforme Bittencourt (2007), após a obtenção da matriz de rigidez e do vetor de carregamento, deve-se aplicar as condições de contorno, ou seja, impor ao problema que determinadas regiões assumam valores conhecidos de deslocamento.

Por exemplo, ao considerar que a lateral esquerda da Figura 3.7 esteja engastada, impõem-se aos nós que a compõem que o deslocamento nas direções  $u$  e  $v$  sejam zero, conforme indica a Tabela 3.9.

Se a quantidade apresentada nas colunas "Direção  $u$ " ou "Direção  $v$ " equivaler a 1, significa que aqueles graus de liberdade possuem condições de contorno, caso contrário, o valor será zero. Desse modo, se houver condições de contorno nos GDL  $u$  e  $v$ , as direções assumirão os valores das colunas "Valor na direção  $u$ " ou "Valor na direção  $v$ ", respectivamente.

**Tabela 3.9 – Condições de contorno para a placa quadrada discretizada por dois elementos**

Número do nó	Direção $u$	Direção $v$	Valor na direção $u$	Valor na direção $v$
1	1	1	0	0
2	1	1	0	0

Fonte – Autoria Própria

Para aplicar as condições de contorno no algoritmo, utiliza-se a técnica denominada de rearranjo da matriz de rigidez e do vetor de carregamento. Ela é caracterizada por zerar as respectivas linhas e colunas que representa os graus de liberdade que possuem as condições de contorno. Após essa operação, a técnica prevê a imposição do valor 1 nas posições  $K_{ij}$  onde  $i$  e  $j$  assumem o número do grau de liberdade zerado. Por exemplo, ao considerar a segunda linha da Tabela 3.9, o nó condicionado corresponde ao segundo nó da numeração global e conforme a regra da construção tabela LOCE, atribui-se os graus de liberdade 3 e 4, respectivamente, e assim, as posições  $K_{33}$  e  $K_{44}$  devem ser substituídas pelo valor 1. Nesse sentido, a Equação 3.21 apresenta a matriz de rigidez após a aplicação



das condições de contorno.

$$[K] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & K_{55}^{(e1)} + K_{11}^{(e2)} & K_{56}^{(e1)} + K_{12}^{(e2)} & K_{13}^{(e2)} & K_{14}^{(e2)} \\ 0 & 0 & 0 & 0 & K_{65}^{(e1)} + K_{21}^{(e2)} & K_{66}^{(e1)} + K_{22}^{(e2)} & K_{23}^{(e2)} & K_{24}^{(e2)} \\ 0 & 0 & 0 & 0 & K_{32}^{(e2)} & K_{33}^{(e2)} & K_{34}^{(e2)} \\ 0 & 0 & 0 & 0 & K_{24}^{(e2)} & K_{24}^{(e2)} & K_{24}^{(e2)} & K_{24}^{(e2)} \end{bmatrix} \quad (3.21)$$

Quanto ao vetor carregamento  $\{F\}$ , ao passo que o elemento da linha  $i$  ou da coluna  $j$  da matriz de rigidez  $[K]$  é zerado do grau de liberdade em questão, o vetor deve ser rearranjado através da subtração do valor original da carga com o produto da condição de contorno e o valor de  $K_{ij}$  da posição substituída por zero. Por exemplo, ao zerar a coluna que representa o grau de liberdade 2, ao zerar o terceiro elemento da coluna,  $K_{32}$ , a terceira posição do vetor de carga deverá ser substituída por  $F_{31} - 0 \cdot K_{32}$ .

### 3.7 Integração Numérica

O uso de coordenadas locais na modelagem matemática do problema de elementos finitos permite a simplificação dos limites de integração das matrizes elementares apresentadas na seção 3.5. No entanto, em algumas funções não é possível obter suas expressões analíticas, tornando necessário a aplicação técnicas de integração numérica (BITTENCOURT, 2007).

Nesse estudo, utilizou-se a quadratura de Gauss-Legendre como recurso, pois para um número  $n$  de pontos de amostragem fornece o resultado exato de integração unidimensional  $I$  para um polinômio de grau  $P = 2n - 1$ , conforme a Equação 3.22.

$$I = \int_{-1}^1 f(\xi) d\xi = \sum_{l=1}^n H_l f(\xi_l) \quad (3.22)$$

De acordo com Bittencourt (2007), essa técnica consiste em obter a solução da integral pelo somatório dos valores da função em pontos de amostragem dentro do intervalo  $[-1, 1]$  multiplicados por coeficientes de ponderação  $H_l$ , de tal forma que melhore a precisão do resultado da integração com um menor número de pontos a comparado a regra de Newton-Cotes.

Além disso, a quadratura pode ser expandida para funções de duas variáveis,  $f(\xi, \eta)$ , quais são recorrentes nos modelos de elementos finitos para elementos quadrangulares bidimensionais a exemplo da Equação 3.23.

$$I = \int_{-1}^1 \int_{-1}^1 f(\xi, \eta) d\xi d\eta \quad (3.23)$$

A Equação 3.23 pode ser resolvida numericamente considerando apenas a parte em  $\eta$  e mantendo constante  $\xi$ . Para  $n_2$  pontos de amostragem na direção  $\eta$  e aplicando na Equação 3.22, obtém-se a Equação 3.24.

$$I = \int_{-1}^1 f(\xi, \eta) d\eta = \sum_{j=1}^{n_2} H_j f(\xi, \eta_j) = g(\xi) \quad (3.24)$$

O resultado da Equação 3.24 é, em essência, uma função de uma variável  $g(\xi)$ . Deste modo, ao aplicar  $n_1$  pontos de integração na direção  $\xi$ , a integral de  $g(\xi)$  é expressa pela Equação 3.25.

$$I = \int_{-1}^1 g(\xi) d\xi = \sum_{i=1}^{n_1} H_i g(\xi_i) \quad (3.25)$$

Substituindo a Equação 3.24 em 3.25, tem-se como resultado da quadratura a Equação 3.26.

$$I = \int_{-1}^1 \int_{-1}^1 g(\xi) d\xi = \sum_{i=1}^{n_1} H_i \left[ \sum_{j=1}^{n_2} H_j f(\xi_i, \eta_j) \right] \quad (3.26)$$

Onde os  $n_1$  e  $n_2$  pontos de amostragem  $\xi_i$  e  $\eta_j$  são dados pelos zeros dos polinômios de Legendre  $P_n(\xi)$  e  $P_n(\eta)$  respectivamente. No que tange as ponderações  $H_i$  e  $H_j$ , essas são alcançadas pela solução de  $\frac{2}{(1-t_i^2)[P'_n(\xi_i)]^2}$  e  $\frac{2}{(1-t_j^2)[P'_n(\eta_j)]^2}$ , respectivamente.

No caso da integração numérica de elementos triangulares, como as coordenadas empregadas são naturais  $L_1$  e  $L_2$  e no domínio  $[0, 1]$ , o processo de integração é análogo, mas deve ser especificados nas variáveis e nos limites de integração correspondentes (BITTENCOURT, 2007).

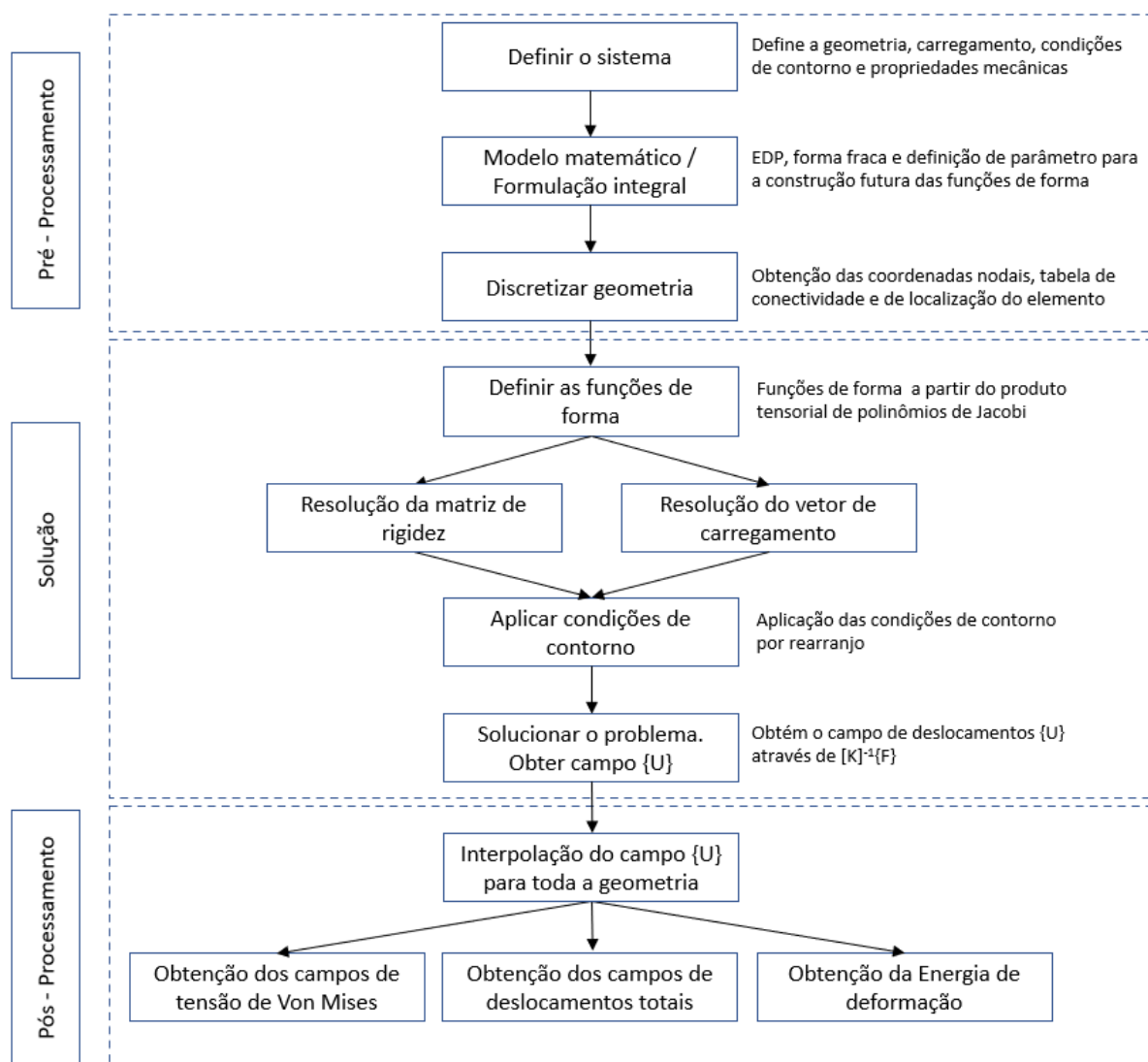
### 3.8 Algoritmos e Suas Considerações

Para a construção de um código de MEF, inicialmente, é necessário escolher uma linguagem de programação que atenda os objetivos descritos na seção 1.2. Nesse sentido, a linguagem *Python* se demonstra uma alternativa, visto que está consolidada no mercado e possui aplicações nas áreas da educação, engenharia, ciência e negócios (FOUNDATION, 2020).

A escolha desta linguagem é devido as características de alto nível, orientação a objetos, tipagem dinâmica e forte e, principalmente, a facilidade de aprendizado. Além disso, o *Python* é código aberto (*open source*) e figura dentre as linguagens de programação mais populares do mundo, com forte participação de usuários em fóruns e comunidades e a disponibilização das mais diversas bibliotecas (FOUNDATION, 2020).

Desta forma, para estruturar o código na linguagem escolhida é importante entender o sentido do fluxo de entrada e saída de informações para cada problema e esse é apresentado na Figura 3.8.

Figura 3.8 – Esquemático do funcionamento geral do algoritmo do MEF para problemas em EPT.



Fonte – Autoria Própria

O algoritmo é constituído de três fases distintas denominadas por "Pré-Processamento", "Solução" e "Pós-Processamento".

A fase de "Pré-Processamento" consiste em obter o problema do meio contínuo em condições aplicáveis no MEF, isto é, a forma discretizada do problema. Após a definir as propriedades geométricas e mecânicas, os apoios e os carregamentos da estrutura, inicia-se efetivamente os trabalhos em *Python*, e na subsecção 3.8.1 é discutido detalhadamente esta etapa.

Em seguida, a fase de "solução" engloba o desenvolvimento das funções de forma, a resolução das matrizes elementares de rigidez e carregamento, a montagem das matrizes globais de rigidez e carregamento, aplicação das condições de contorno e a obtenção dos deslocamentos nodais  $u$ . Foge do escopo do trabalho a discussão da construção do algoritmo de solução na linguagem *Python*, sendo apresentado unicamente a sua implementação no

Apêndice A.

A última fase, o "Pós-processamento", ocorre o tratamento da solução e a obtenção de resultados que são funções do deslocamento, a exemplo das tensões de Von Mises e a energia de deformação.

Como os deslocamentos são encontrados apenas nos pontos de interesse, em geral, realiza-se a interpolação de  $\mathbf{u}$  para as demais regiões da geometria garantindo melhores resoluções para os campos de deslocamentos.

A energia de deformação associada a estrutura pode ser obtida por meio do produto  $\{\mathbf{u}\}^T [\mathbf{K}]\{\mathbf{u}\}$ , onde  $[\mathbf{K}]$  é a matriz global de rigidez sem a aplicação das condições de contorno e  $\{\mathbf{u}\}$  são os deslocamentos nodais.

Quanto as tensões de Von Mises, ao manipular a Equação 2.5, obtém-se a expressão  $\boldsymbol{\sigma} = [\mathbf{D}][\mathbf{B}]\{\mathbf{u}\}$ , onde  $[\mathbf{D}]$  é a matriz constituinte do material e  $[\mathbf{B}]$  a matriz de elasticidade. Como a matriz de elasticidade adquire características numéricas nos pontos de integração para o método de solução utilizado, logo as componentes de tensão são recuperadas diretamente apenas nos pontos de integração de cada elemento requerendo interpolação nas demais posições.

O pós-processamento dos dados podem ter diversas abordagens, cabendo ao usuário escolher a maneira que atenda as suas necessidades e, neste trabalho, o resultado do pós-processamento da solução é representado graficamente pelas curvas de convergências da tensão máxima de Von Mises, da energia de deformação, assim como os campos de deslocamentos e os campos de tensões de Von Mises ao longo da geometria empregando-se, assim, a biblioteca gráfica *Matplotlib* (HUNTER, 2007).

Destaca-se ainda, a capacidade da biblioteca na construção de gráficos bidimensionais mapeados por cores conforme uma escala que varia de acordo com a intensidade de uma variável (HUNTER, 2007), mostrando-se fundamental para a obtenção dos campos de tensões de Von Mises e os campos de deslocamentos apresentados no capítulo 4.

### 3.8.1 Arquivo de entrada

Para o funcionamento adequado do código, o usuário deve inserir um conjunto de informações a respeito do problema denominado arquivo de entrada. Nesse estão compreendidos as características essenciais do sistema discretizado para realizar a solução do problema.

Dentre essas informações, o grau das funções de interpolação, o módulo de elasticidade  $E$ , o coeficiente de Poisson  $\nu$  e a espessura da geometria estão pré-configurados em linear, 210 GPa, 0,30 e 1 mm, respectivamente, podendo alterá-los conforme as necessidades antes de iniciar a solução. Quanto aos demais dados da geometria, a exemplo da matriz de coordenadas, de conectividade e LOCE, o usuário deve se atentar ao formato aceito pelo algoritmo.

Para a geração da malha, como mostrado na seção 3.1, este trabalho empregou o

formato de saída das matrizes de coordenadas e conectividade da biblioteca *Triangle*. Por exemplo, para a geometria apresentada na Figura 3.2, as coordenadas nodais e a tabela de conectividade dos elementos, respectivamente tabelas 3.1 e 3.2, são representadas por matrizes bidimensionais conforme as Equações 3.27 e 3.28.

$$\text{coordenadas} = \text{array}([[0., 0.], [0., 1.], [1., 1.], [1., 0.], [0.5, 0.5]]) \quad (3.27)$$

$$\text{conectividade} = \text{array}([[1, 0, 4], [4, 3, 2], [3, 4, 0], [4, 2, 1]]) \quad (3.28)$$

Como os índices da linguagem de programação adotada inicia em zero, nota-se que a Equação 3.28 usufrui dos índices das posições nodais da Equação 3.27 na construção dos elementos.

Consequente, dada a ordem de interpolação, constrói-se a tabela LOCE, também no formato de matriz bidimensional onde cada coluna representa a localização dos GDL de um elemento, Equação 3.29.

$$\begin{aligned} \text{LOCE} = \text{array}([ & [2, 8, 6, 8], \\ & [3, 9, 7, 9], \\ & [0, 6, 8, 4], \\ & [1, 7, 9, 5], \\ & [8, 4, 0, 2], \\ & [9, 5, 1, 3]]) \end{aligned} \quad (3.29)$$

Em decorrência dos índices iniciarem em zero, a numeração dos GDL nas direções  $u$  e  $v$  são alteradas para  $2n$  e  $2n + 1$ , respectivamente, onde  $n$  é a numeração global do nó, agora baseada nos índices de armazenamento da matriz de coordenadas.

Também vale salientar que para as ordens superiores, os GDL nas colunas devem ser organizadas, inicialmente, em GDL de vértice, em seguida, de aresta e por fim, de face.

Para o carregamento associado a estrutura, a representação ocorre por meio de uma matriz  $M_{ix5}$ , onde  $i$  representa o número de arestas que contém cargas aplicadas. Atribuindo um carregamento uniforme de 10.000 N/m na lateral direita para o exemplo da Figura 3.2, obtém-se como resultado a Equação 3.30

$$\text{carregamento} = \text{array} ([2, 1, 2, 3, 10000]) \quad (3.30)$$

Cada coluna da matriz de carregamento possui uma informação única e deve ser preenchida adotando os seguintes critérios:

- 1<sup>a</sup> coluna: número do índice do elemento na matriz conectividade que contém a aresta com carga aplicada;

- 2<sup>a</sup> coluna: direção do carregamento, sendo 1 paralelo a direção  $x$  e 2 paralelo a direção  $y$ ;
- 3<sup>a</sup> coluna: número do índice na matriz de coordenadas do primeiro nó que compõe a aresta;
- 4<sup>a</sup> coluna: número do índice na matriz de coordenadas do segundo nó que compõe a aresta; e
- 5<sup>a</sup> coluna: intensidade do carregamento uniforme na aresta em  $N/m$ .

Para ordens superiores a linear, não se deve incluir os modos de alta ordem na matriz de carregamento, uma vez que a construção do vetor de carga é baseada nos nós físicos que compõem a aresta e na tabela LOCE, qual contém informações sobre os nós modais.

Quanto as condições de contorno do problema, estas também devem ser representadas por uma matriz  $M_{j \times 5}$ , com  $j$  representando o número de nós condicionados aos valores iniciais associados. Nesse sentido, cada coluna da matriz deve ser preenchida adotando o seguinte procedimento:

- 1<sup>a</sup> coluna: número do índice na matriz de coordenadas do nó segundo que possui condição de contorno;
- 2<sup>a</sup> coluna: verifica se a condição de contorno se aplica na direção  $u$ , sendo o valor lógico 1 como caso afirmativo e 0 como caso negativo;
- 3<sup>a</sup> coluna: verifica se a condição de contorno se aplica na direção  $v$ , sendo o valor lógico 1 como caso afirmativo e 0 como caso negativo;
- 4<sup>a</sup> coluna: valor da condição de contorno na direção  $u$ , em metros, caso a segunda coluna tenha valor lógico 1; e
- 5<sup>a</sup> coluna: valor da condição de contorno na direção  $v$ , em metros, caso a terceira coluna tenha valor lógico 1.

Por exemplo, para funções de forma lineares, considerando um engaste na lateral esquerda da Figura 3.2, o resultado do procedimento descrito anteriormente é dado pela Equação 3.31.

$$\text{condições de contorno} = \text{array} ([[0, 1, 1, 0, 0], \\ [1, 1, 1, 0, 0]]) \quad (3.31)$$

Ao passo que se eleva a ordem de interpolação, ocorre o surgimento de nós de aresta, inclusive nas arestas apoiadas, fazendo-se necessário condicioná-los aos respectivos deslocamentos.

O código de solução de MEF foi desenvolvido baseado no sistema internacional de unidades, sendo necessário ao usuário se atentar ao discretizar a geometria e gerar as matrizes nas unidades adequadas.

### **3.9 O Software Ansys**

O software Ansys é uma ferramenta computacional de simulação da companhia de nome homônimo que utiliza o MEF para solução de problemas mecânicos, de fluidos e elétricos oferecendo suporte nas áreas da ciência, engenharia e educação (ANSYS, 2020).

Nesse último, a empresa oferece uma versão estudantil do seu software, possuindo limitações quanto a complexidade das geometrias e sua discretização, mas atendendo bem a maioria das situações.

Para problemas mecânicos, é possível realizar análises de problemas no espaço plano no programa comercial, o que garante um cenário mais próximo ao do algoritmo construído, permitindo a comparação dos resultados.

Desta forma, os estudos de casos propostos no capítulo 4, a fim de verificar a coerência dos valores obtidos pelo algoritmo, solucionou-os no software comercial e comparou-se os resultados através do erro relativo com os valores de convergência.

# Capítulo 4

## Resultados e Discussões

Neste capítulo, será realizado a modelagem e a solução de geometrias planas pelo método dos elementos finitos. O refino  $-h$  é utilizado apenas nos elementos lineares onde o tamanho dos elementos será reduzido a cada etapa de refinamento. Para o refino  $-p$ , malha é fixada e a ordem de interpolação dos elementos é aumentada e, no presente trabalho serão utilizadas as aproximações com  $P = 2$  e  $4$ . Quanto ao refino  $-hp$ , trata-se da composição das duas técnicas de refino em que o tamanho do elemento é reduzido ao mesmo passo que a ordem de aproximação será elevada para  $P = 2$  e  $4$  neste trabalho.

### 4.1 Estudo de Caso 1: Placa Quadrada

Estruturas com geometrias simples, que possuem fácil fabricação são amplamente empregadas devido aos seus aspectos construtivos e financeiros envolvidos. Nesse sentido, geometrias com formas circulares, triangulares ou quadrilaterais são preferíveis à geometrias complexas.

A proposta de executar o algoritmo para uma geometria simples é verificar se os resultados obtidos são razoáveis e assim atender a maior parte das geometrias utilizadas em serviço.

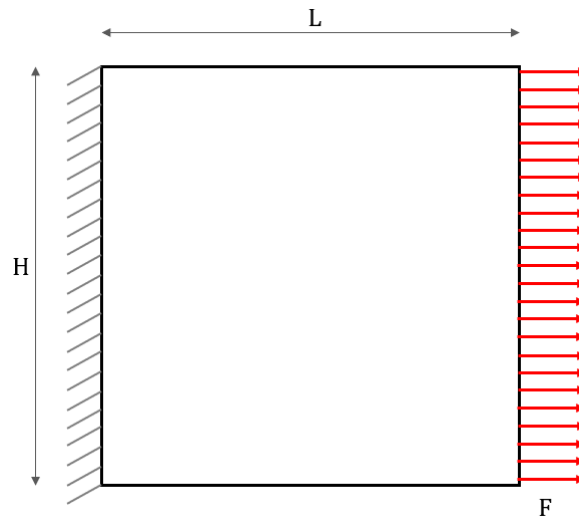
Nesse sentido, considere a placa quadrada engastada na lateral esquerda com carregamento axial na lateral direita, Figura 4.1, e as propriedades geométricas e mecânicas dispostas na Tabela 4.1.

**Tabela 4.1 – Propriedades geométricas e mecânicas da placa quadrada**

	L	H	E	$\nu$	t	F
Valor	100	100	210	0,3	1	100
Unidades	mm	mm	GPa	-	mm	N/mm

**Fonte – Autoria Própria**



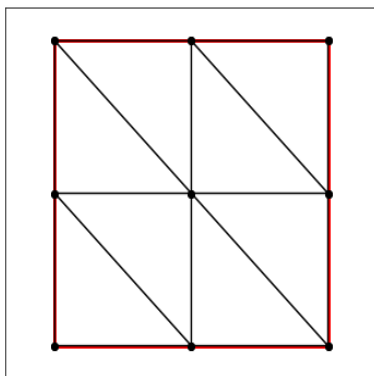
**Figura 4.1 – Placa quadrangular engastada com carregamento axial**

Fonte – Autoria Própria

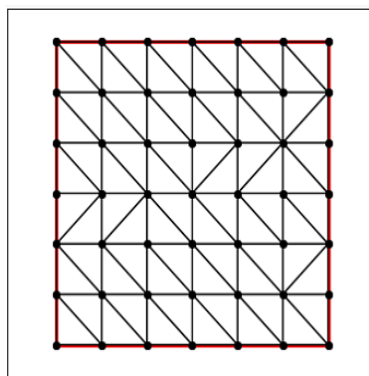
Para este problema, optou-se pela construção de uma malha homogênea, de modo que os pontos estejam distribuídos igualmente por toda extensão da geometria. Quanto ao refino da malha, utilizou-se a mesma estratégia de construção e suas representações estão expressas na Figura 4.2.

**Figura 4.2 – Placa quadrada discretizada por triângulos lineares**

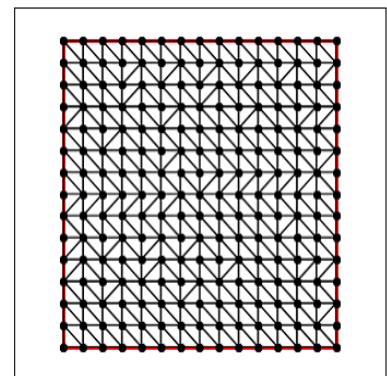
(a) Malha com 8 elementos



(b) Malha com 144 elementos



(c) Malha com 784 elementos



Fonte – Autoria Própria

A condição de engaste da placa é aplicada no algoritmo por meio da imposição dos deslocamentos iguais a zero nas direções  $u$  e  $v$  dos nós físicos e modais que compõem a aresta. Para o carregamento uniforme  $F$ , o algoritmo computa os valores de força nas direções  $u$  e  $v$  nos nós físicos e modais através da solução da Equação 3.19b nas arestas dos elementos em que o carregamento atua.

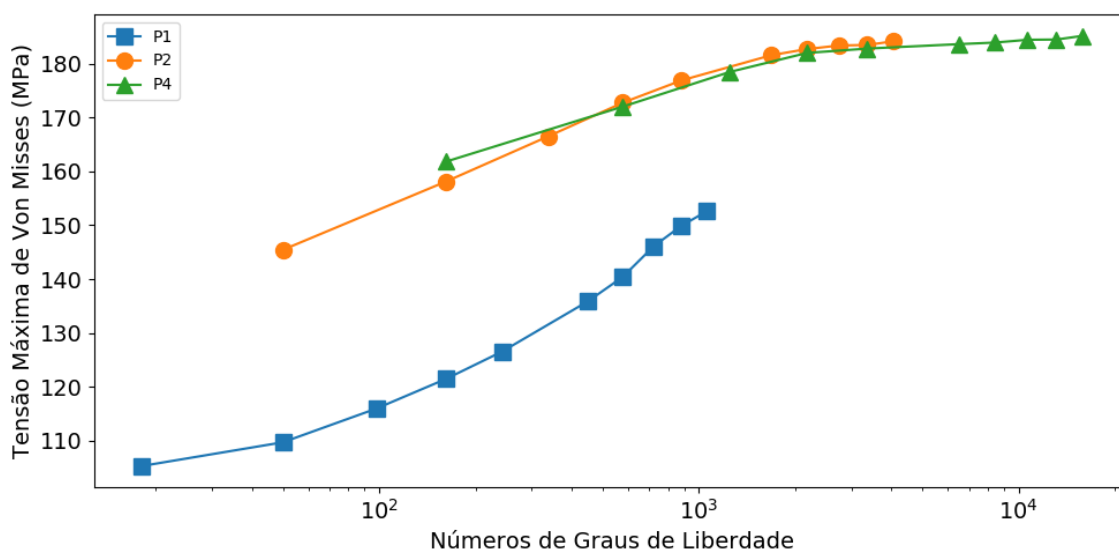
Aplicadas as condições de contorno e o carregamento correspondente, o algoritmo provê os valores de deslocamento e, após o tratamento desses, obtêm-se os valores de

energia de deformação e de tensão de Von Mises e relacionará, para todas as situações, com os graus de liberdade totais.

Vale salientar que para esse último, os valores são recuperados apenas nos pontos de integração numérica do elemento e nas demais posições são interpolados. Isso implica que cada elemento de alta ordem, possui um valor máximo de tensão e a curva de convergência de tensão máxima de Von Mises corresponde ao maior valor do conjunto de tensões dos elementos de cada malha.

Nesse sentido, o comportamento monotônico apresentado pelas curvas de tensão em relação para o primeiro estudo de caso, Figura 4.3, é o esperado pois, ao passo que aumenta o número de graus de liberdade, independente da ordem de interpolação, converge para o valor de 182 MPa.

**Figura 4.3 – Convergência da tensão máxima de Von Mises para o problema da placa quadrada.**



Fonte – Autoria Própria

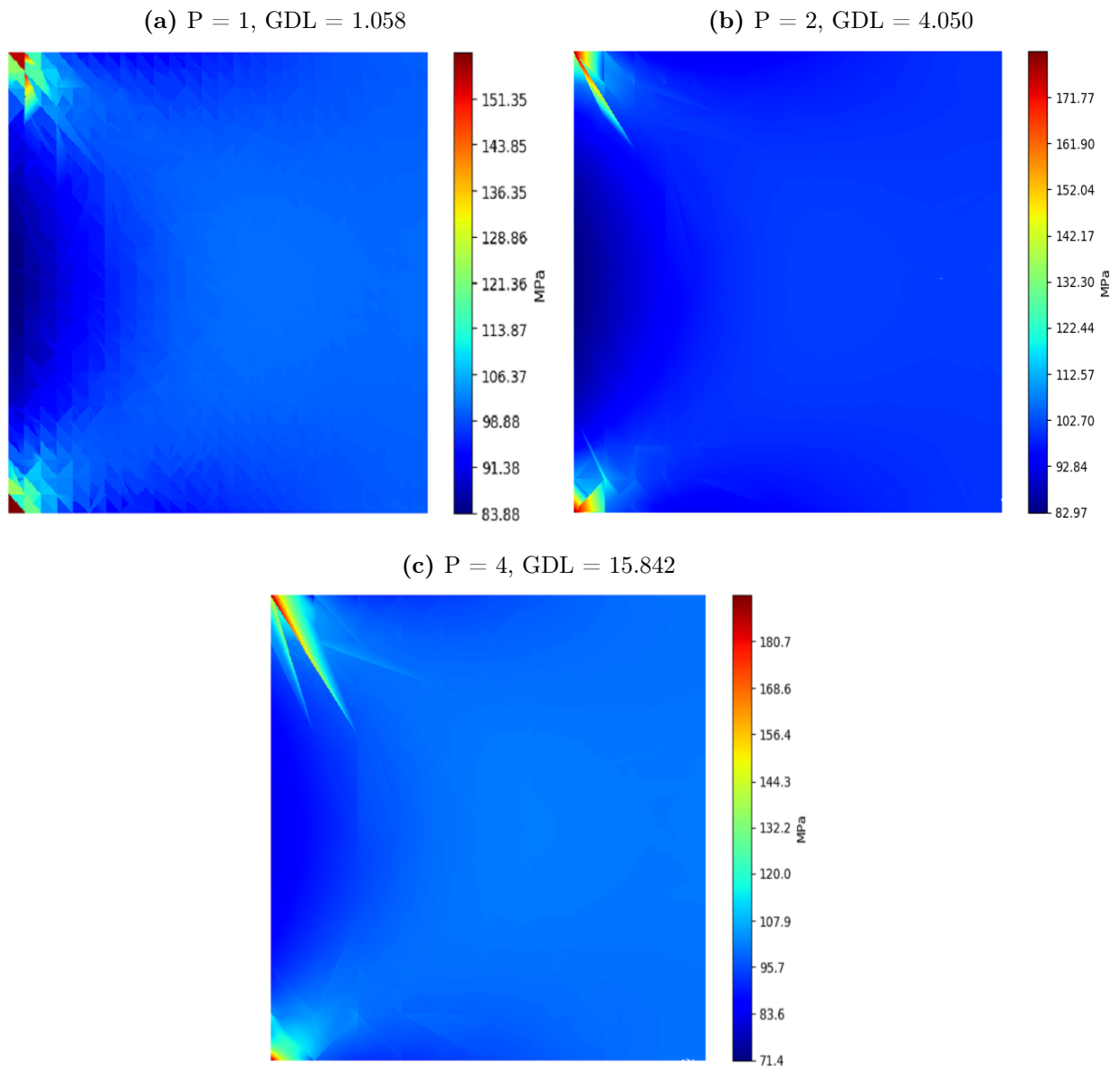
Apesar da convergência de tensão, é possível verificar algumas diferenças de comportamento dadas as ordens de interpolação. Por exemplo, com um número de 250 GDL, a tensão máxima de Von Mises para  $P = 1$  possui um erro relativo de 33,5% em relação ao valor de convergência enquanto para a ordem  $P = 2$  e 4 os erros são, respectivamente, 13,2% e 11,1%. Ao aumentar em 75% o número de GDL, aproximadamente 1.000, o erro relativo de 19,8% para os elementos lineares é superior em referência a situação anterior para as ordens 2 e 4.

Consequente, a distribuição de tensão de Von Mises ao longo da geometria, a Figura 4.4 apresenta os campos de tensão para cada ordem  $P$  de aproximação utilizada com uma malha fixa de 968 elementos. A exemplo, para  $P = 1$  devido as características do triângulo de deformação constante (CST), a tensão é constante ao longo do elemento,

tornando-se visível a mudança de tensão de um elemento adjacente a outro.

Por outro lado, para  $P = 2$  e  $4$ , existem variações de tensões ao longo do elemento devido a variação dos valores da matriz de elasticidade  $[B]$  em cada ponto de integração e desse modo, reduzindo o efeito do gradiente de tensão de um elemento a outro. Nesse sentido, a suavidade dos campos de tensão é melhorada a medida que aumenta o grau das funções polinomiais de interpolação mesmo mantendo a malha fixa.

**Figura 4.4 – Campos de tensões de Von Mises para o problema da placa quadrada. Utilizando uma malha fixa de 968 elementos. A suavidade do campo aumenta com o grau de interpolação empregado.**



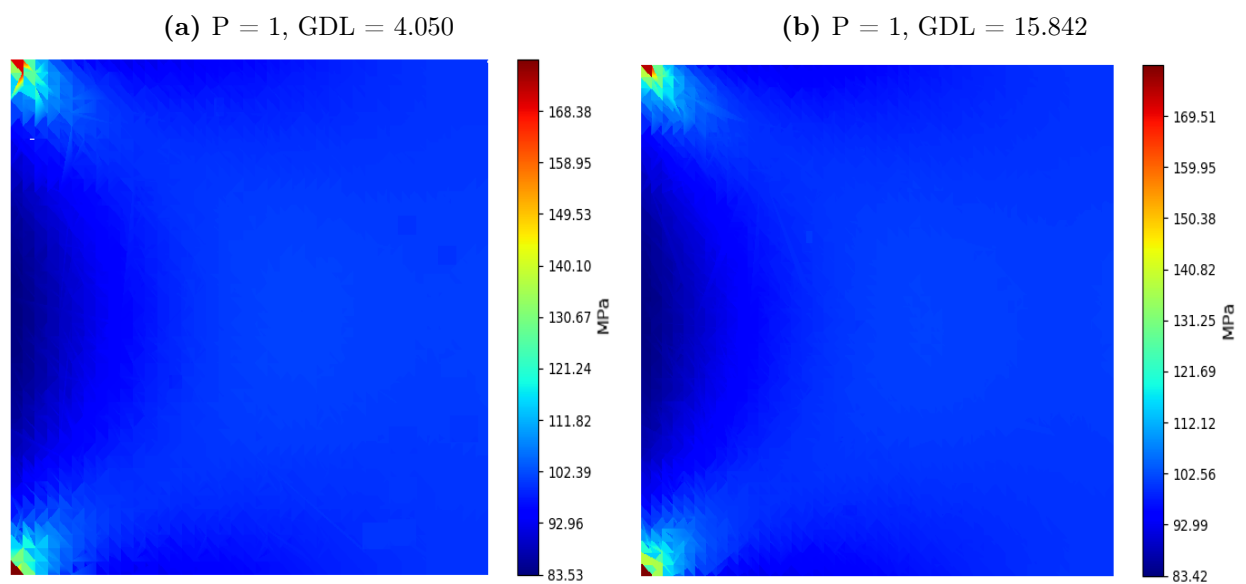
Fonte – Autoria Própria

Com o propósito de reduzir o gradiente de tensão e comparar graficamente com os resultados de  $P = 2$  e  $4$  da situação anterior, o problema da placa quadrada foi solucionado

utilizando elementos lineares mas, agora com 4.050 e 15.842 graus de liberdade, conforme Figura 4.5.

Nota-se que a redução do gradiente foi alcançada aumentando os GDL do problema, todavia, de acordo com a legenda das imagens, as tensões máximas ainda não se aproximam do valor de convergência, corroborando a ideia de que além do número de graus de liberdade, a ordem das funções de forma refletem nos resultados.

**Figura 4.5 – Campos de tensões de Von Mises para o problema da placa quadrada utilizando o refinamento -h**



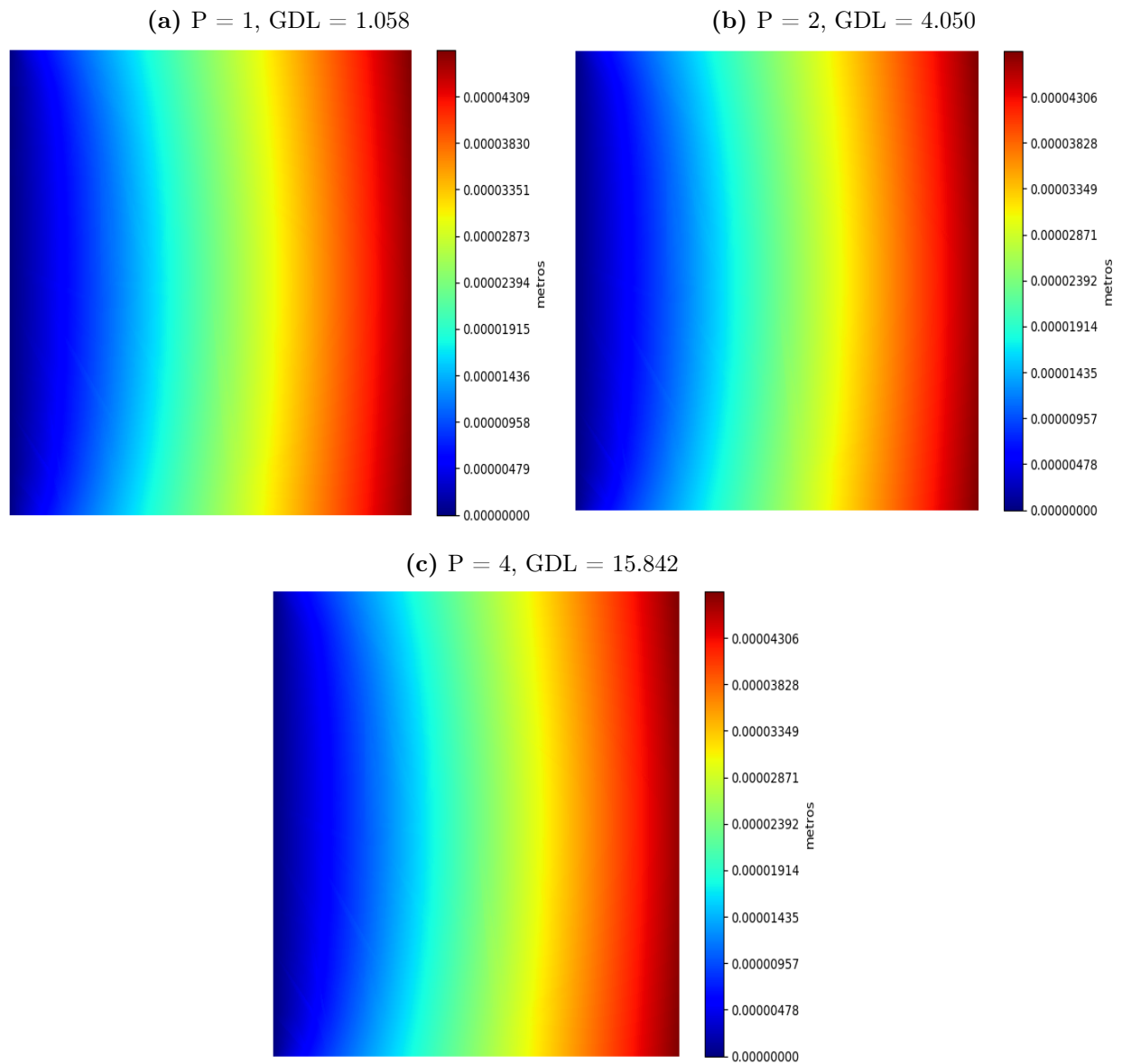
Fonte – Autoria Própria

Quanto a concentração de tensão observada próximo ao engaste (lateral esquerda) em todas as situações apresentadas, pode-se atribuir ao fato dessa porção de material (engaste) estar condicionada a não se deslocar enquanto a porção logo em seguida não tem essa restrição e, como consequência, ocasiona a redução de seção transversal pelo efeito Poisson.

Quanto aos deslocamento totais,  $\sqrt{u^2 + v^2}$ , os campos de deslocamentos associados às ordens em análise, Figura 4.6, nota-se que não há nenhuma região de descontinuidade dos deslocamentos, variando suavemente ao longo da geometria, desde a região de engaste, deslocamento nulo, até a aresta onde o carregamento está submetido, com o deslocamento máximo, independente da ordem de interpolação.

Do mesmo modo que os campos de tensões, o problema foi solucionado com elementos lineares para o mesmo número de graus de liberdade de  $P = 2$  e 4 da situação anterior e, observou-se nenhuma mudança gráfica na suavidade do campo, devido aos deslocamentos se apresentarem na ordem de  $10^{-5}$  e as diferenças na ordem de  $10^{-11}$ .

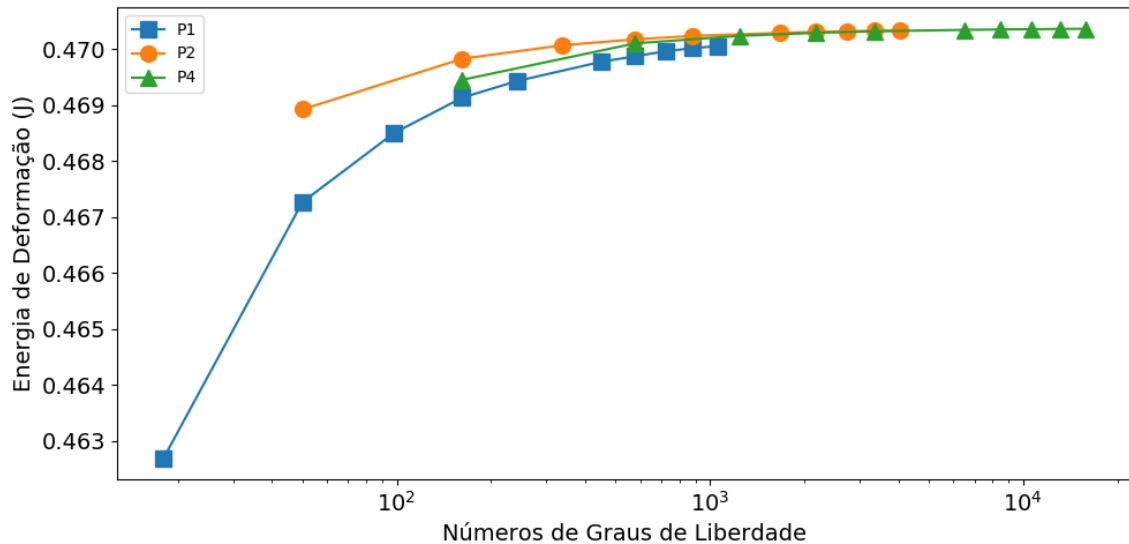
Figura 4.6 – Campos de deslocamentos totais para o problema da placa quadrada. A suavidade é percebida desde a ordem 1, pois o comportamento de  $\{u\}$  é linear ao longo do elemento.



Fonte – Autoria Própria

A energia de deformação associada ao problema,  $\{u\}^T [K] \{u\}$ , Figura 4.7, tem o comportamento monotônico e, ao passo que se acresce a ordem de interpolação, a convergência é acelerada, sendo que para  $P = 2$  e  $4$ , é quase imediato. Em comparação, nota-se que, com aproximadamente 500 GDL nas ordens  $P = 2$  e  $4$ , a energia de deformação se aproximou do valor de convergência, 0,4704 J, enquanto para os elementos lineares necessitou-se aproximadamente 1.000 graus de liberdade para assumir o valor de 0,4698 J.

Figura 4.7 – Energia de deformação (J) para o problema da placa quadrada. A taxa de convergência da energia de deformação aumenta ao passo que se eleva o grau dos polinômios de interpolação



Fonte – Autoria Própria

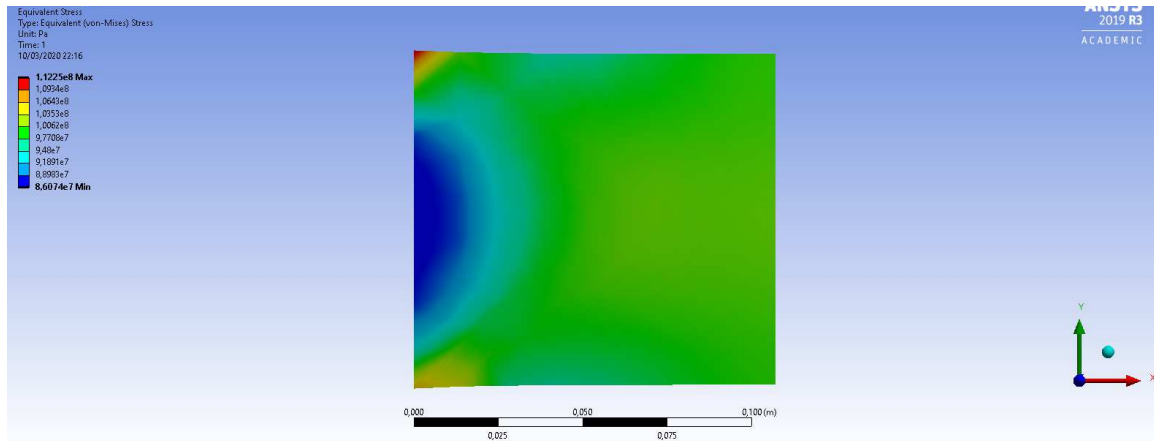
A fim de comparar os resultados do algoritmo, o mesmo problema foi solucionado no software Ansys. Utilizou-se elementos triangulares lineares e as configurações padrão do Ansys para gerar a malha, pois nesta versão do software, há limitações de refinamento para um número elevado de elementos.

Os campos de tensão de Von Mises e deslocamento totais associados ao problema dadas as condições da Tabela 4.1 são apresentados na Figura 4.8. Observa-se a semelhança gráfica das soluções de tensão e deslocamento com as Figuras 4.4 e 4.6, apesar de diferirem os mapas de cores.

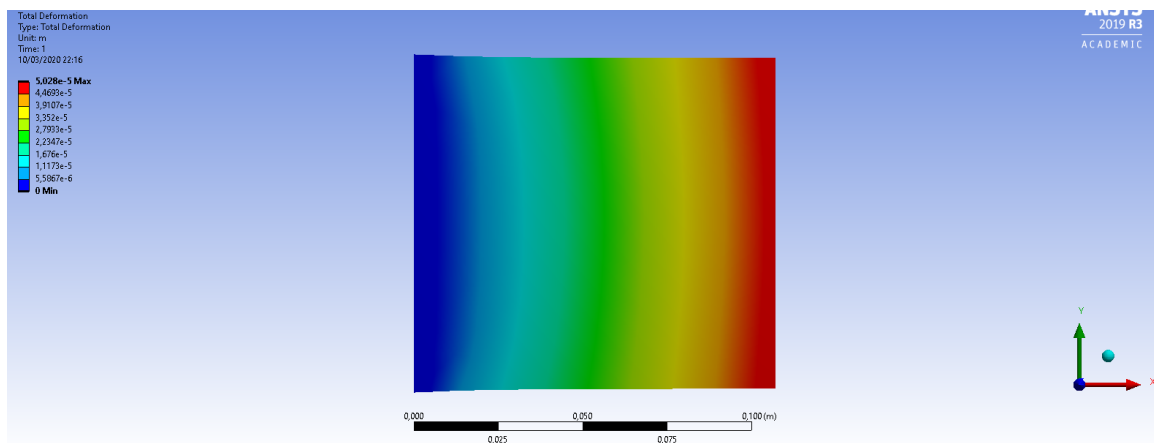
Em seguida, foi estabelecida uma malha com as características mais próximas a malha obtida pelo software, resultando em uma malha de 124 elementos e 90 nós. Também foram estabelecidas mais duas malhas, de 289 e 1.089 nós, e solucionadas unicamente para elementos lineares a fim de estabelecer um comparativo com o mesmo número de graus de liberdade para os resultados anteriores das funções de alta ordem.

Figura 4.8 – Campos de tensões de Von Mises e de deslocamento total para o problema da placa quadrada resolvido no software Ansys 2019 R3 versão estudantil. Malha utilizada contou com 90 nós e 146 elementos

(a) Campo de Tensões



(b) Campo de Deslocamentos



Fonte – Autoria Própria

Pela Tabela 4.2, comparando os resultados, observa-se que mesmo com 10% a menos de graus de liberdade, o algoritmo apresentou um erro relativo de 33,3% em relação ao valor de convergência de tensão máxima de Von Mises, enquanto o software comercial apresentou um erro relativo de 38,3%. Para as ordens superiores, o erro relativo é reduzido ao passo que em  $P = 4$ , o erro é praticamente nulo.

Como apontado anteriormente, uma das dificuldades da convergência da tensão máxima de Von Mises para elementos lineares reside no número reduzido de graus de liberdade comparado as ordens superiores e, nessa situação, ao solucionar o problema com o mesmo número de GDL das ordens  $P = 2$  e 4, houve uma melhora dos resultados de tensão, com a redução do erro relativo para 24,1% e 11,5% respectivamente, mas ainda

superiores aos erros das ordens  $P = 2$  e  $4$  da malha fixa.

**Tabela 4.2 – Comparativo dos resultados do software e o algoritmo para o problema da placa quadrada**

	Ansys	P1	P1	P1	P2	P4
Tensão máxima (MPa)	112,25	121,44	138,13	161,72	157,81	182,06
Deslocamento máximo (m)	5,028e-5	4,73e-5	4,73e-5	4,72e-5	4,72e-5	4,72e-5
Número de GDL	180	162	578	2.178	578	2.178
Erro Relativo (%)	38,3	33,3	24,1	11,1	13,3	0,03

**Fonte – Autoria Própria**

Por outro lado, além do número de graus de liberdade, a distribuição dos nós e o tamanho dos elementos empregados na discretização da geometria podem influenciar na solução, mesmo em uma malha fixa.

Como as maiores tensões estão localizadas na região próxima ao engaste, verificado pelos resultados anteriores, para uma malha fixa de 121 nós e 200 elementos, realizou a distribuição dos nós pela geometria aumentando a densidade de pontos próximo ao engaste, conforme os casos da Tabela 4.3 e ilustrado pela Figura 4.9.

A redistribuição dos pontos ao longo da geometria visa diminuir o tamanho dos elementos garantindo a redução dos gradientes de tensão na região de concentração de tensão ao passo que se aproxima dos valores de convergência da tensão máxima de Von Mises.

Em outras duas situações, foram geradas, respectivamente, malhas de 441 e 1.681 nós e resolvidas para as funções de forma lineares, visando comparar os resultados com as funções de ordem superiores na malha fixa.

**Tabela 4.3 – Distribuição de pontos, em porcentagem, ao longo da geometria**

	0 até L/2	L/2 até L
Caso 1	60%	40%
Caso 2	70%	30%
Caso 3	80%	20%
Caso 4	90%	10%

**Fonte – Autoria Própria**

Essa distribuição dos nós é feita a fim de reduzir o tamanho dos elementos próximo do engaste, resultando em um menor gradiente de tensão nesta região com intuito de minimizar a propagação de erro da solução.



**Tabela 4.4 – Tensão máxima de Von Mises obtida para cada caso de distribuição de pontos apresentados na Tabela 4.3.**

GDL	Grau de Interpolação	Caso 1	Caso 2	Caso 3	Caso 4
242	1	130,44	133,72	136,49	138,88
882	1	141,15	144,55	147,23	149,64
3.362	1	168,66	171,90	174,49	176,47
882	2	171,91	175,99	179,41	182,35
3.362	4	180,23	181,79	182,70	183,15

**Fonte – Autoria Própria**

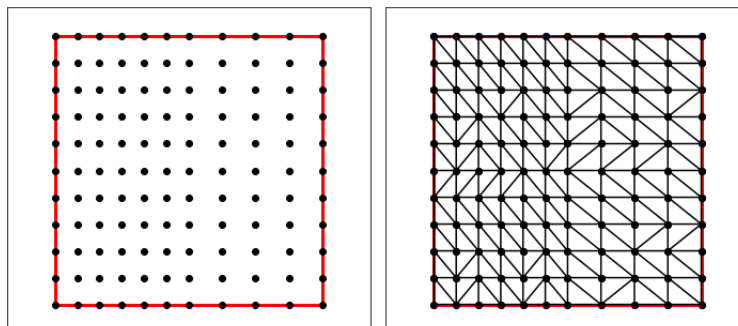
Os resultados apresentados na Tabela 4.4 indicam que a redução do tamanho dos elementos nas regiões dotadas de concentrações de tensão melhora a sensibilidade da solução, com menores erros relativos em relação ao valor de convergência para o caso 4, em todas as ordens de interpolação.

Nas situações onde se elevou o número de graus de liberdade para as funções de forma lineares, os erros relativos foram reduzidos ao valor mínimo de 3% comparado ao valor de convergência, mas ainda maiores que os valores para as ordens superiores, que alcançaram erros relativos aproximadamente zero nos casos 3 e 4.

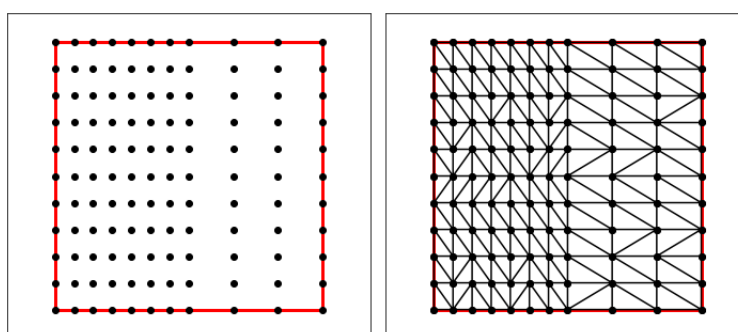
Dessa forma, além do refino da malha e do aumento da ordem  $P$  de interpolação, o modo que é realizado a distribuição dos nós ao longo da geometria também influem na sensibilidade da solução.

Figura 4.9 – Distribuição dos pontos na geometria com aumento da densidade de nós na região próxima ao engaste. Malha de 200 elementos e 121 nós

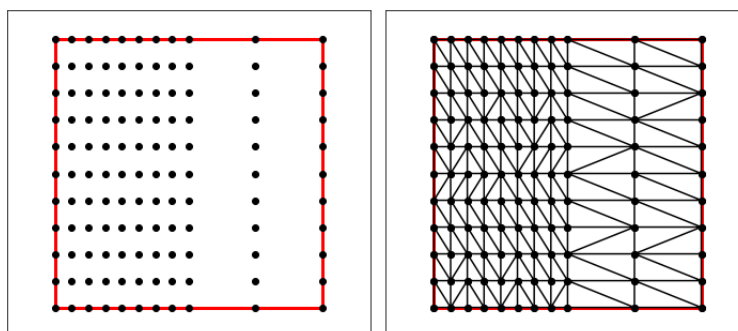
(a) Caso 1: 60% – 40%



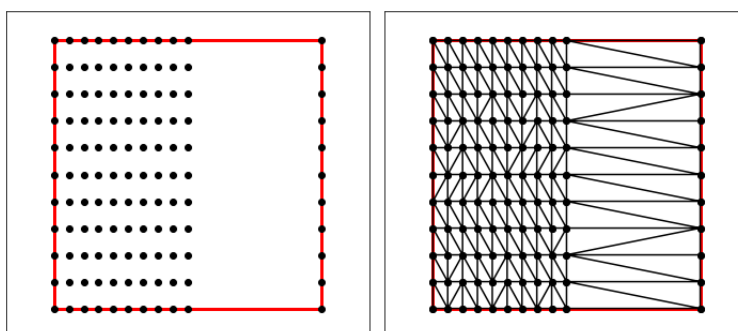
(b) Caso 2: 70% – 30%



(c) Caso 3: 80% – 20%



(d) Caso 4: 90% – 10%



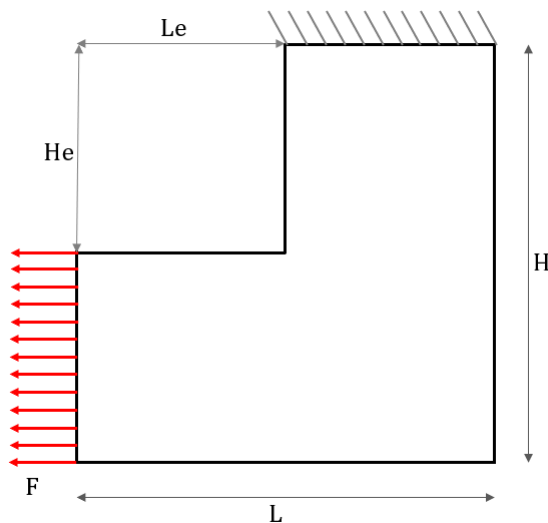
Fonte – Autoria Própria

## 4.2 Estudo de Caso 2: Placa Retangular em Formato de L

Em muitas situações práticas, estruturas podem ser submetidas aos mais variados carregamentos em direções diferentes aos seus apoios.

Neste segundo caso, o problema proposto visa verificar o comportamento do algoritmo para geometria em formato em "L" engastada na aresta superior e carregamento axial na lateral esquerda, apresentada na Figura 4.10.

**Figura 4.10 – Placa em formato de "L" com carregamento axial**



Fonte – Autoria Própria

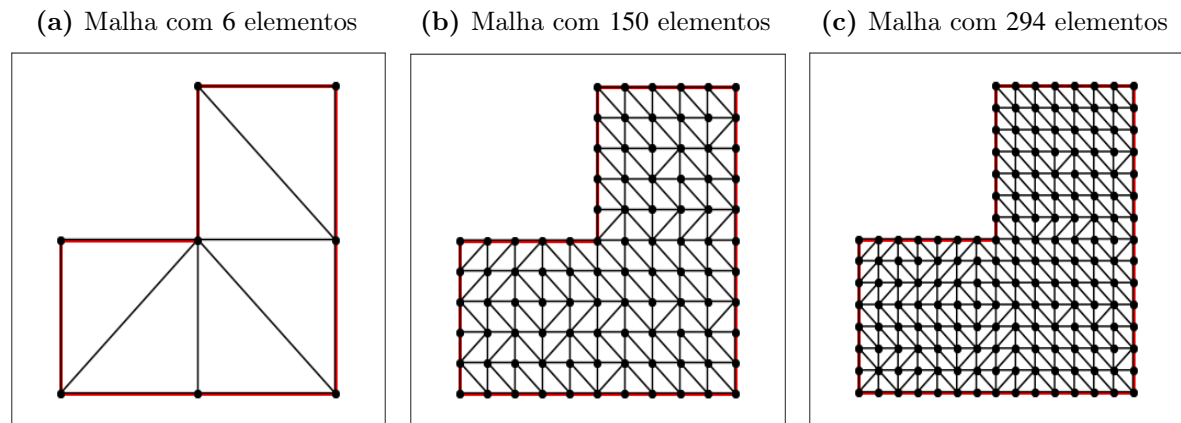
As propriedades geométricas e mecânicas associadas ao problema proposto estão dispostas na Tabela 4.5.

**Tabela 4.5 – Propriedades geométricas e mecânicas da placa retangular em formato de "L"**

	L	H	Le	He	E	$\nu$	t	F
Valor	100	100	50	50	210	0,3	1	30
Unidades	mm	mm	mm	mm	GPa	-	mm	N/mm

Fonte – Autoria Própria

Para construir a malha desta geometria, optou-se em distribuir os pontos igualmente espaçados nas direções horizontal e vertical. Quanto ao refino da malha, utilizou-se do mesmo recurso e, na Figura 4.11, são apresentadas 3 das malhas empregadas no processo de solução.

**Figura 4.11 – Placa em formato de "L" discretizada por triângulos**

Fonte – Autoria Própria

Da mesma maneira que apresentado na Seção 4.1, as condições de contorno são aplicadas impondo aos deslocamentos associados aos graus de liberdade fixos os valores iguais a zero nas direções  $u$  e  $v$  dos nós que compõem a aresta engastada para elementos de alta ordem, e o carregamento pela solução da equação 3.19b.

Aplicadas as condições de contorno e o carregamento, a solução é pós processada similarmente ao estudo de caso da seção 4.1 e, a Figura 4.12 apresenta as curvas de convergência de tensão máxima de Von Mises para o problema da placa em formato de "L".

Observa-se um comportamento monotônico para todas as ordens e a estabilização acelerada para ordens superiores à linear com menores números de graus de liberdade, convergindo para um valor de tensão máxima de Von Mises de 333 MPa.

Para as ordens de interpolação  $P = 2$  e 4, com aproximadamente 450 GDL, os erros relativos da tensão máxima de Von Mises são próximos de 1% e se mantiveram na mesma faixa de valores com o acréscimo de GDL, enquanto para os elementos lineares necessitou-se cerca de 1.000 graus de liberdade para erro relativo similar.

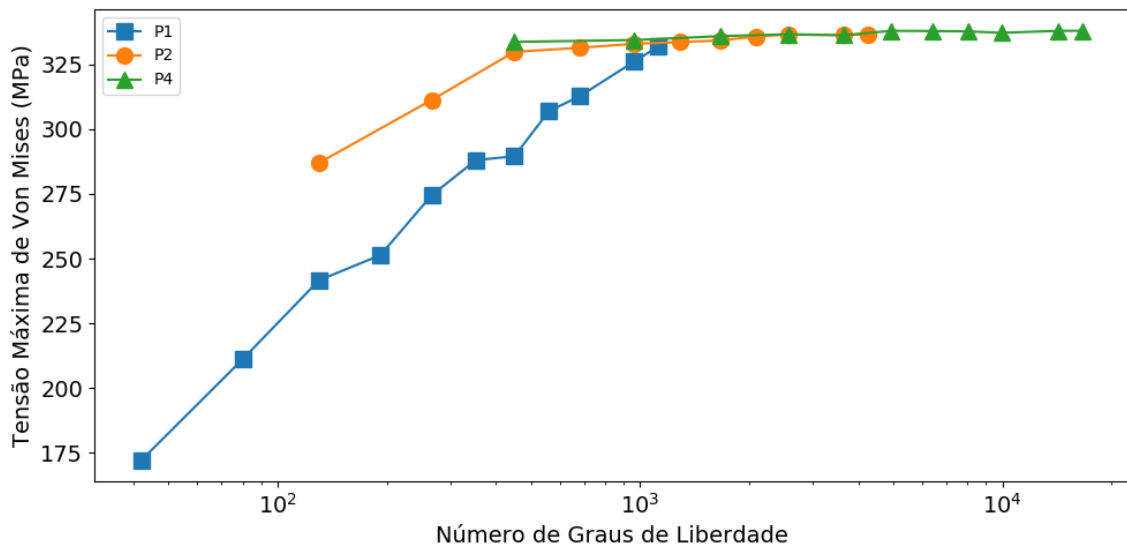
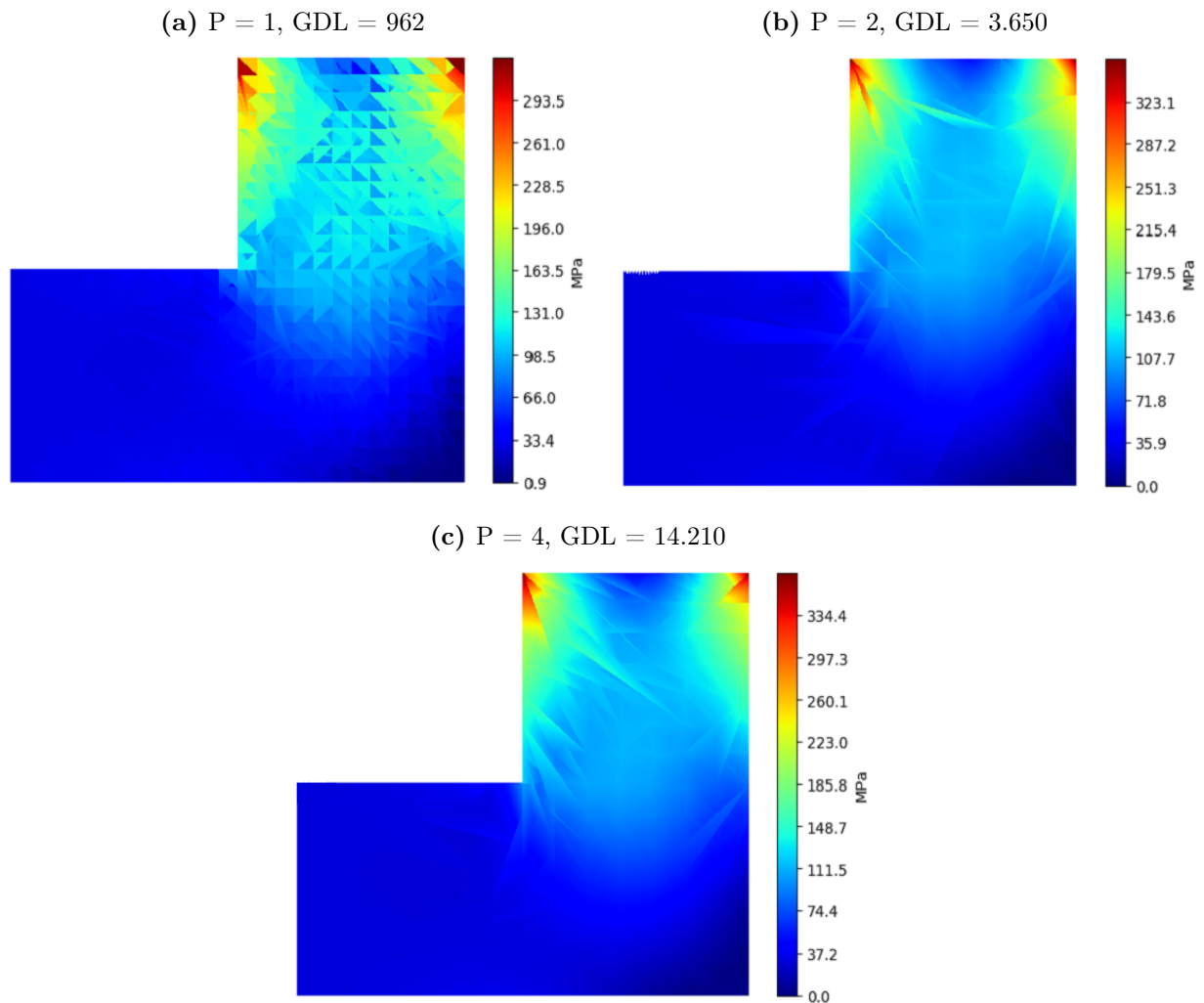
**Figura 4.12 – Convergência da tensão máxima de Von Mises para o problema da placa em "L"**

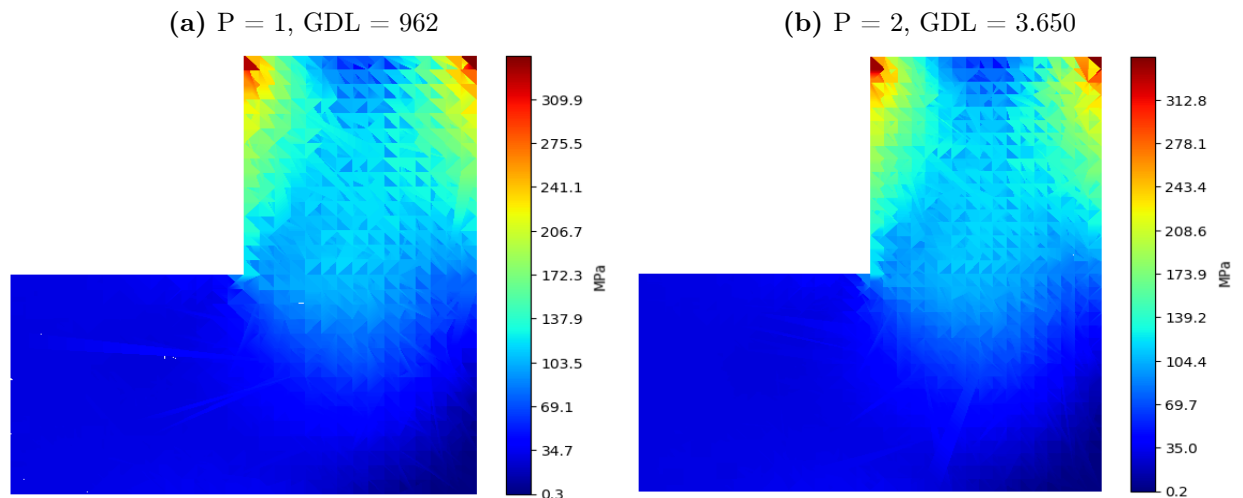
Figura 4.13 – Campos de tensões de Von Mises para o problema da placa em "L". Em todas as situações utilizou-se malha fixa com 864 elementos



Fonte – Autoria Própria

Com o intuito de comparar nas condições em que os elementos lineares possuam o mesmo número de graus de liberdade que os elementos de ordem superior, realizou-se o refinamento  $-h$  da malha e os campos de tensões de Von Mises de saída são apresentados na Figura 4.14. Ainda que o número de GDL se equipare aos dos elementos de ordem  $P = 2$  e  $4$ , ainda é possível notar a presença de gradientes de tensão, ainda que em níveis menores. Esta dificuldade de suavizar os campos é em detrimento dos valores constantes das derivadas da matriz de elasticidade  $[B]$  em cada elemento.

**Figura 4.14 – Campos de tensões de Von Mises para o problema da placa em "L". Refinamento -h**



Fonte – Autoria Própria

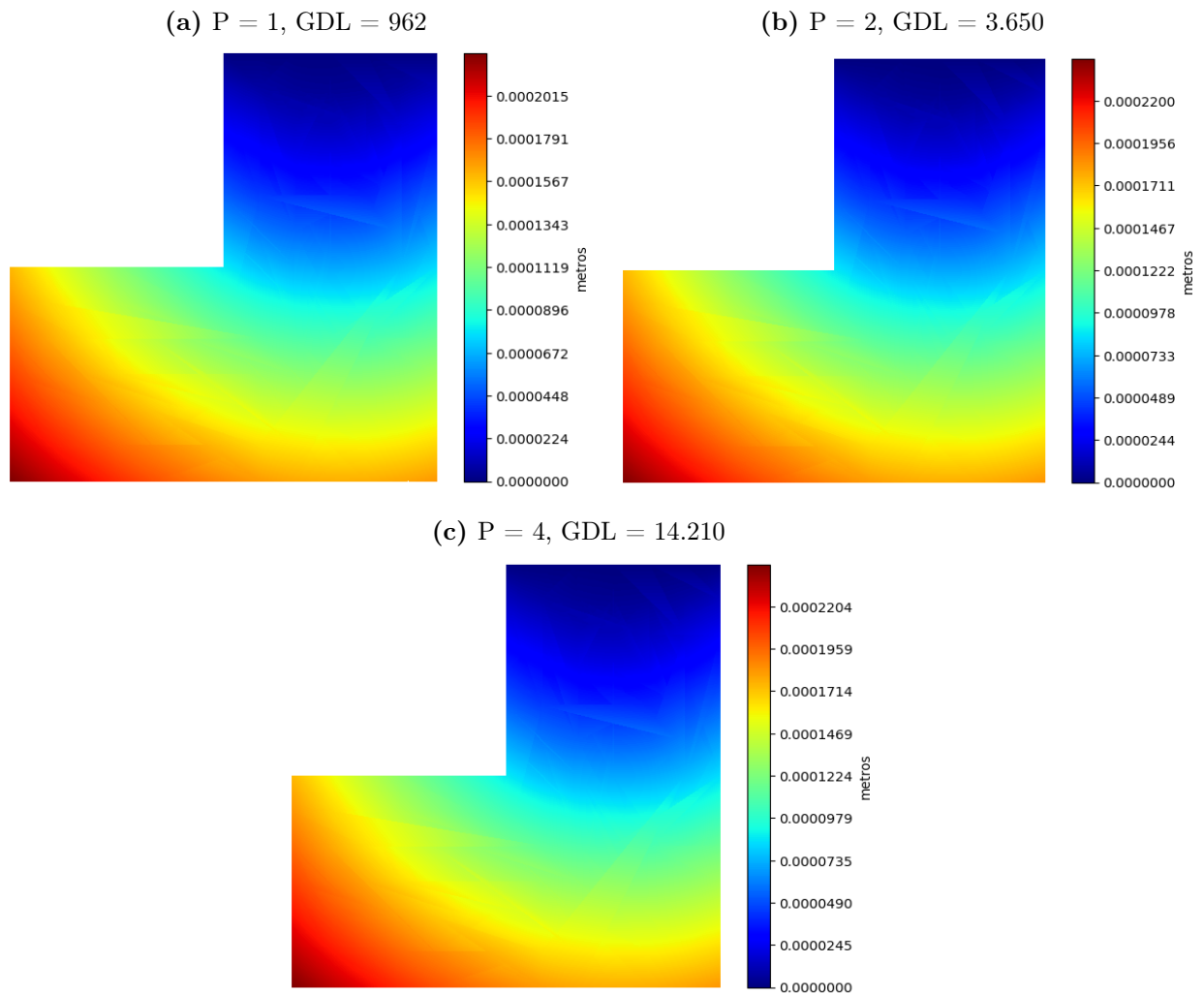
Verifica-se ainda que as concentrações de tensões estão localizadas na região próxima ao engaste e o comportamento é o esperado, pois existe uma porção de material fixo e a porção logo em seguida tende a se movimentar, reduzindo a seção transversal pelo efeito Poisson.

Na região central da placa, há a presença de um canto vivo apresentando uma concentração de tensão que, embora não seja da mesma magnitude das tensões próximas ao engaste, pode influir no comportamento da estrutura.

Por outro lado, os deslocamentos totais para o problema,  $\sqrt{u^2 + v^2}$ , são apresentados por meio dos campos de deslocamento, Figura 4.15. O comportamento apresentado não demonstra gradientes expressivos de deslocamento, com o deslocamento mínimo coincidente com o apoio fixo (lateral superior) e deslocamento máximo na porção mais afastada do apoio fixo, com maior liberdade para movimentar-se, na região que contém o carregamento (lateral esquerda).

Além disso, nota-se que, diferentemente da tensão de Von Mises, os deslocamentos são menos sensíveis a alteração de ordem de interpolação e ao refino da malha, apresentando diferenças da ordem de  $10^{-9}$ .

Figura 4.15 – Campos de deslocamentos totais para o problema da placa em "L". Em todas as situações utilizou-se malha fixa com 864 elementos



Fonte – Autoria Própria

Quanto a energia de deformação associada ao problema, Figura 4.16, o comportamento monotônico apresentado é o esperado e notadamente, a estabilização ocorre quase que imediatamente para ordens superiores à linear. Ao passo que, para  $P = 2$  e 4 necessitam, respectivamente, 282 e 462 graus de liberdade para se estabilizarem em uma energia de deformação de 0,201 J, os elementos lineares apenas alcançam um valor de 0,198 J a partir de 1.000 GDL.



Figura 4.16 – Convergência da Energia de Deformação (J) para o problema da placa em "L".

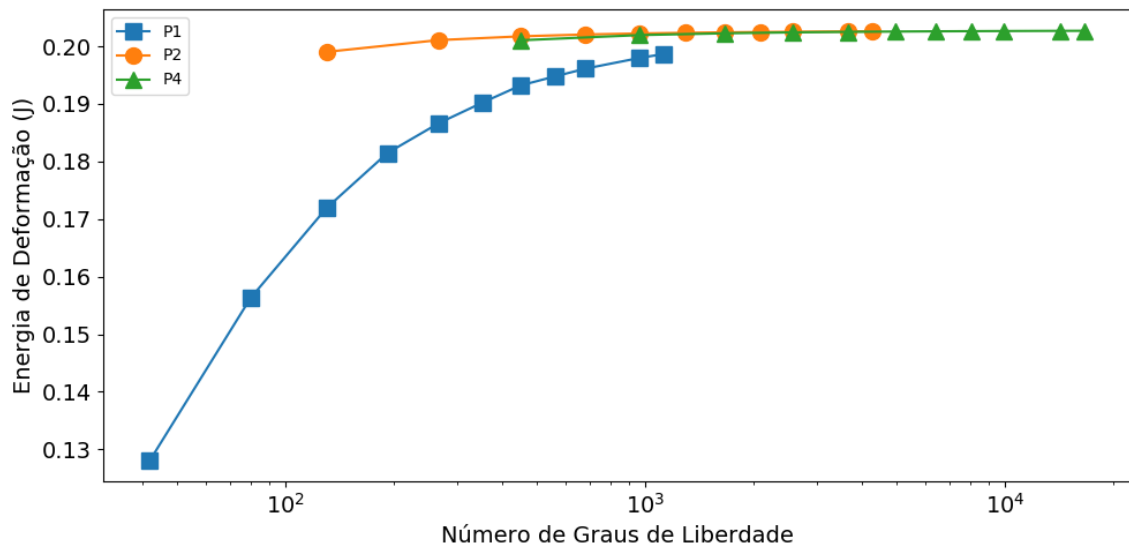
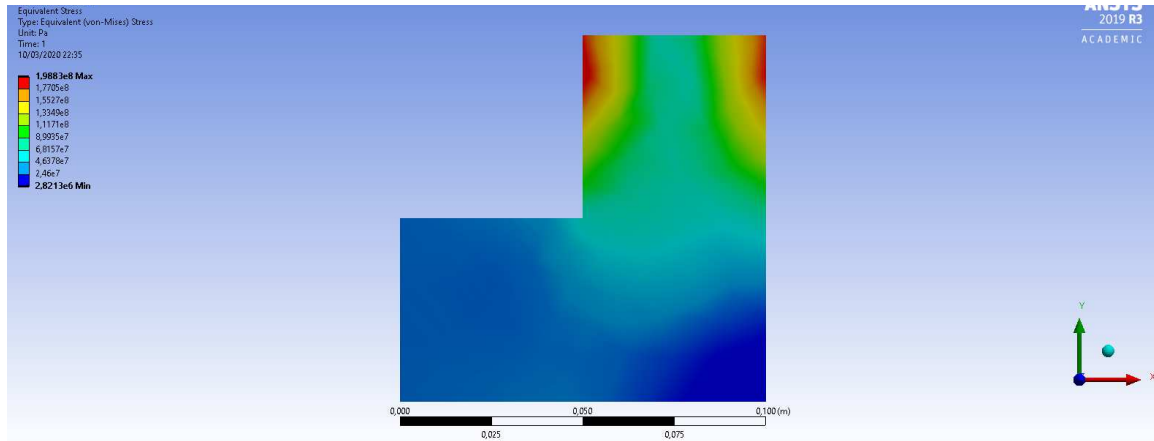
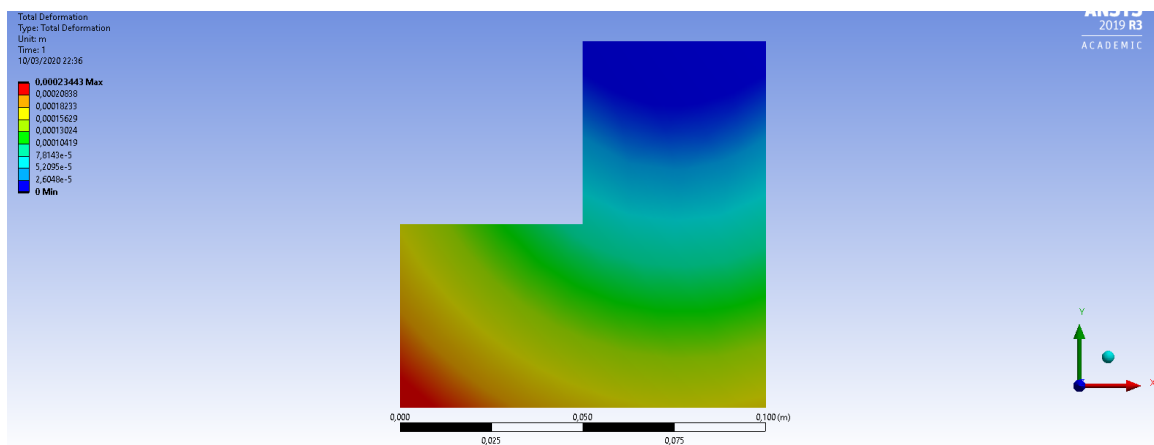


Figura 4.17 – Campos de tensões e de deslocamentos totais para o problema da placa em "L" resolvido no software Ansys 2019 R3 versão estudantil. A malha utilizada contou com 83 nós e 126 elementos

(a) Campo de Tensões



(b) Campo de Deslocamentos



Fonte – Autoria Própria

No algoritmo, a fim de tornar mais próximo às condições de simulação do Ansys, construiu-se uma malha com 150 elementos. Também, duas outras malhas foram geradas, 341 e 1.281 nós respectivamente, e solucionadas com elementos lineares com o intuito de comparar com os valores obtidos pelas funções de alta ordem.

Na Tabela 4.6, são comparados os valores de tensão de Von Mises e de deslocamentos totais obtidos pelo algoritmo e o software.

**Tabela 4.6 – Comparativo dos resultados do software e o algoritmo para o problema da placa em formato "L"**

	Ansys	P1	P1	P1	P2	P4
Tensão máxima (MPa)	198,83	251,33	308,78	329,12	331,39	333,66
Deslocamento máximo (m)	2,34e-4	2,24e-4	2,24e-4	2,24e-4	2,25e-4	2,25e-4
Número de GDL	166	192	682	2.562	682	2.562
Erro Relativo (%)	40,3	24,5	7,27	1,20	0,48	0,20

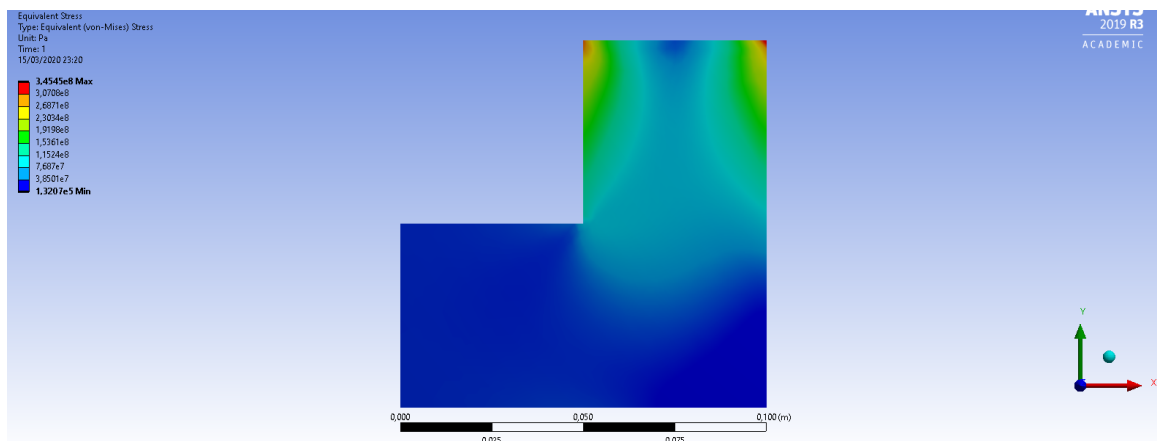
Fonte – Autoria Própria

Nota-se que em todas as situações analisadas o algoritmo alcançou um valor de tensão de Von Mises mais próximo do valor de convergência, com erro relativo de 0,48% e 0,2% para as ordens quadrática e quártica, respectivamente. Para as ordens lineares, na situação de malha fixa, o erro relativo registrado foi de 24,5%, enquanto nas malhas refinadas, 341 e 1.281 nós, os erros foram de 7,27% e 1,2%. Mesmo com o refinamento empregado, a situação de menor erro relativo para os elementos lineares, os resultados são inferiores comparado as ordem superiores.

Por outro lado, observa-se que a distribuição de tensão registrada pelo software é abaixo dos valores obtidos pelo algoritmo em toda a extensão da geometria, Figura 4.17. Uma possível razão deste fato pode estar relacionado ao erro geométrico da discretização do domínio na solução do problema que, possivelmente é abaixo do número necessário para obter uma solução satisfatória.

Nesse sentido, a fim de verificar essa hipótese, realizou um refinamento da malha no software reexecutando o problema e o resultado obtido para o campo de tensões é demonstrado na Figura 4.18.

**Figura 4.18 – Campo de tensões de Von Mises para o problema da placa em "L" com refinamento da malha solucionado no software Ansys 2019 R3 versão estudantil. A malha utilizada contou com 3.318 GDL e 3.168 elementos**



Fonte – Autoria Própria

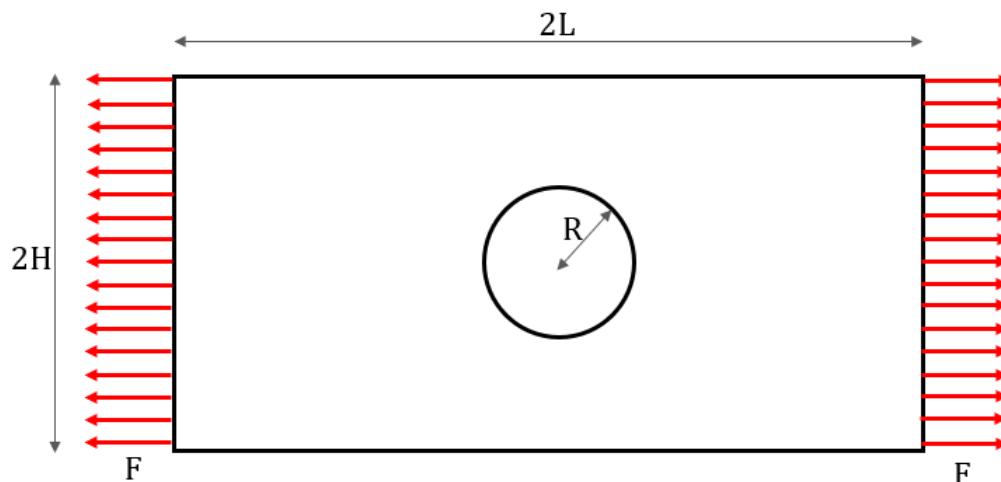
Observa-se que os valores de tensão do software se aproximam dos valores obtido pelo algoritmo, contudo, requerem cerca de 386% e 29,5% GDL a mais do que as ordens de aproximação  $P = 2$  e 4 respectivamente.

### 4.3 Estudo de caso 3: Placa retangular com furo central

Em problemas reais, as descontinuidades geométricas são as principais causas de concentrações de tensão em um corpo. Nos pontos de concentração de tensão, os algoritmos de elementos finitos podem ser confrontados com situações de singularidade de tensões - pontos da malha que não há convergência da tensão mesmo com refino da malha.

Nesta condição, o caso proposto visa verificar a robustez do algoritmo em decorrência da redução da seção transversal de uma geometria pela inserção de um furo.

**Figura 4.19 – Placa retangular com furo central com carregamento axial**



Fonte – Adaptado de SIMSOLID (2015)

Na situação analisada, considere uma placa com furo central com carregamento axial, Figura 4.19, e as propriedades geométricas e mecânicas dispostas na Tabela 4.7.

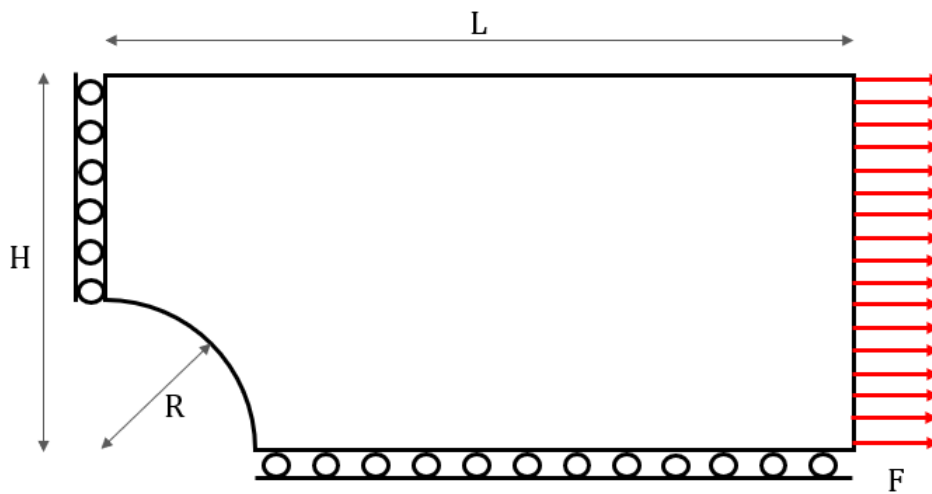
**Tabela 4.7 – Propriedades geométricas e mecânicas da placa com furo central.**

	L	H	R	E	$\nu$	t	F
Valor	100	50	20	210	0,3	1	100
Unidades	mm	mm	mm	GPa	-	mm	N/mm

Fonte – SIMSOLID (2015)

Devido as condições de simetria e para efeito de análise, será considerado apenas 1/4 da seção, como mostrado na Figura 4.20.

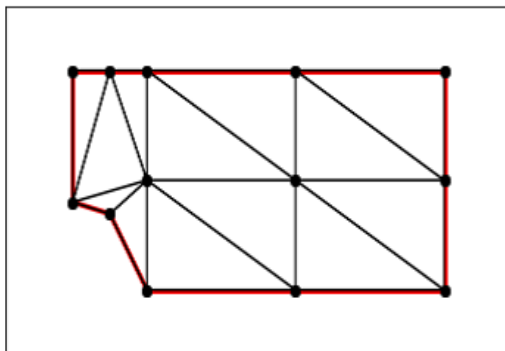
Figura 4.20 – Seção analisada da placa retangular com furo central com carregamento axial



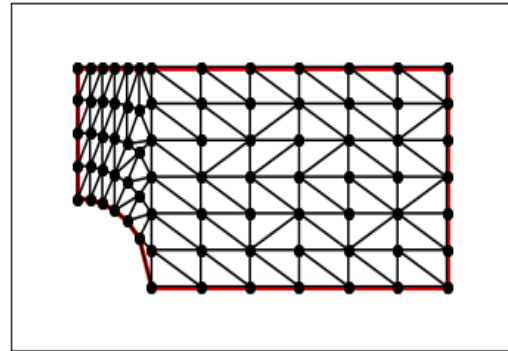
Fonte – Adaptado de SIMSOLID (2015)

Figura 4.21 – Placa retangular com furo central discretizada por triângulos

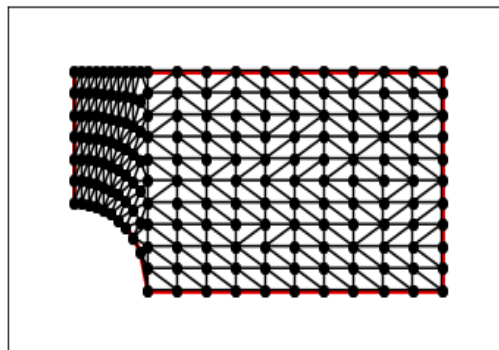
(a) Malha com 13 elementos



(b) Malha com 122 elementos



(c) Malha com 324 elementos



Fonte – Autoria Própria

A estratégia utilizada para a construção da malha foi melhorar a resolução da região circular por meio do aumento da densidade de pontos, pois devido a redução da

seção transversão, espera-se que ocorra a incidência de concentrações de tensões nessa faixa da geometria.

Para o refino, empregou-se o mesmo princípio de reduzir o tamanho dos elementos na região circular e algumas das malhas resultantes do processo de construção estão dispostas na Figura 4.21

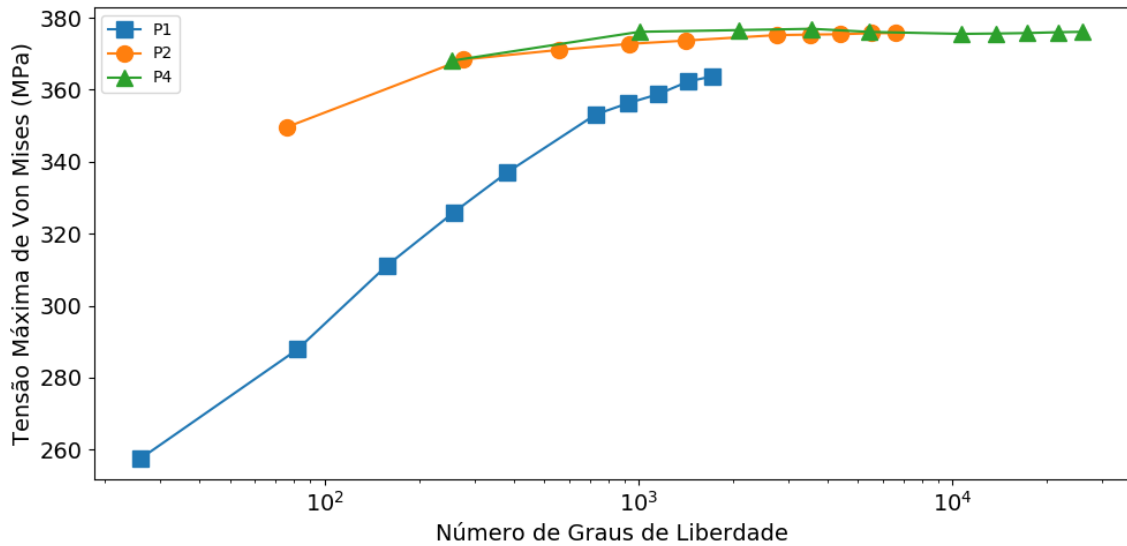
Em geometrias com contornos complexos, a exemplo da região circular, a biblioteca *Triangle* possui limitações na geração de uma malha fiel a geometria original para uma baixa densidade de pontos, Figura 4.21a, mas a medida que se acresce o número de nós nos contornos, Figuras 4.21b e 4.21c, o erro geométrico é minimizado.

Diferentemente dos outros estudos de caso, a presente situação é referente a análise de apenas um quarto da geometria total. Nesse caso, atentou-se em aplicar ao modelo as condições de contorno de apoio deslizante nas laterais inferior e esquerda, isto é, a imposição dos deslocamentos na direção  $u$  da lateral esquerda e os deslocamentos da direção  $v$  da lateral inferior, respectivamente, iguais a zero.

Após aplicar ao modelo às condições de contorno, o carregamento e solucioná-lo, realiza-se o tratamento das respostas conforme descrito nos estudos de casos anteriores, obtendo as curvas de convergência de tensão máxima de Von Mises, os campos de tensão de Von Mises e campos de deslocamentos totais.

Em relação as curvas de convergência, em todas as situações apresentadas na Figura 4.22, a tensão máxima converge para a solução analítica, 372,2 MPa, apontada no manual de validação de problemas de referências de software comercial (SIMSOLID, 2015). Observa-se ainda que, para as ordens quadrática e quártica, a estabilização é alcançada a partir de 250 graus de liberdade enquanto para a ordem linear, mesmo na última situação, 1.718 GDL e tensão máxima de 363,7 MPa, há um erro relativo de 2%.

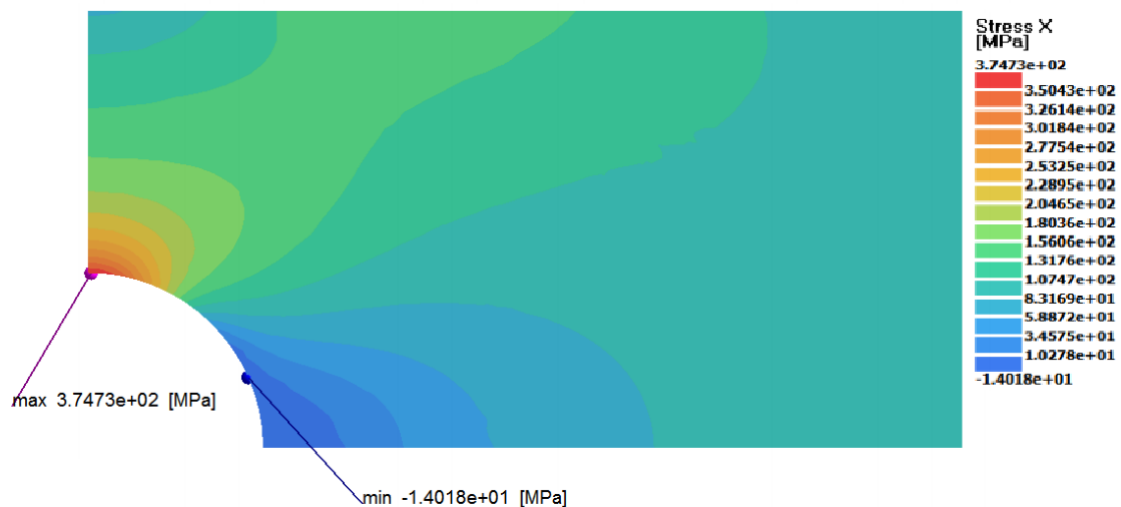
Figura 4.22 – Convergência da tensão máxima de Von Mises para o problema da placa circular com furo central. A taxa de convergência aumenta com o grau das funções interpoladoras utilizadas.



Fonte – Autoria Própria

Quanto as concentrações de tensões, conforme a justificativa indicada durante a construção da malha, o ponto que possui as maiores tensões coincide com a região que há redução da seção transversal da placa e corroborada pelo manual de validação e a simulação pelo do software SIMSOLID (2015), Figura 4.23.

Figura 4.23 – Campo de tensões de Von Mises obtido para o teste VS08 - Placa com um furo - do manual de validação



Fonte – SIMSOLID (2015)

A fim de ilustrar a distribuição de tensão segundo o critério de Von Mises ao longo da geometria e compatibilizar com o manual de validação, a Figura 4.24 apresenta

os campos de tensão para as ordens de interpolação  $P = 1, 2$  e  $4$  gerados pelo algoritmo.

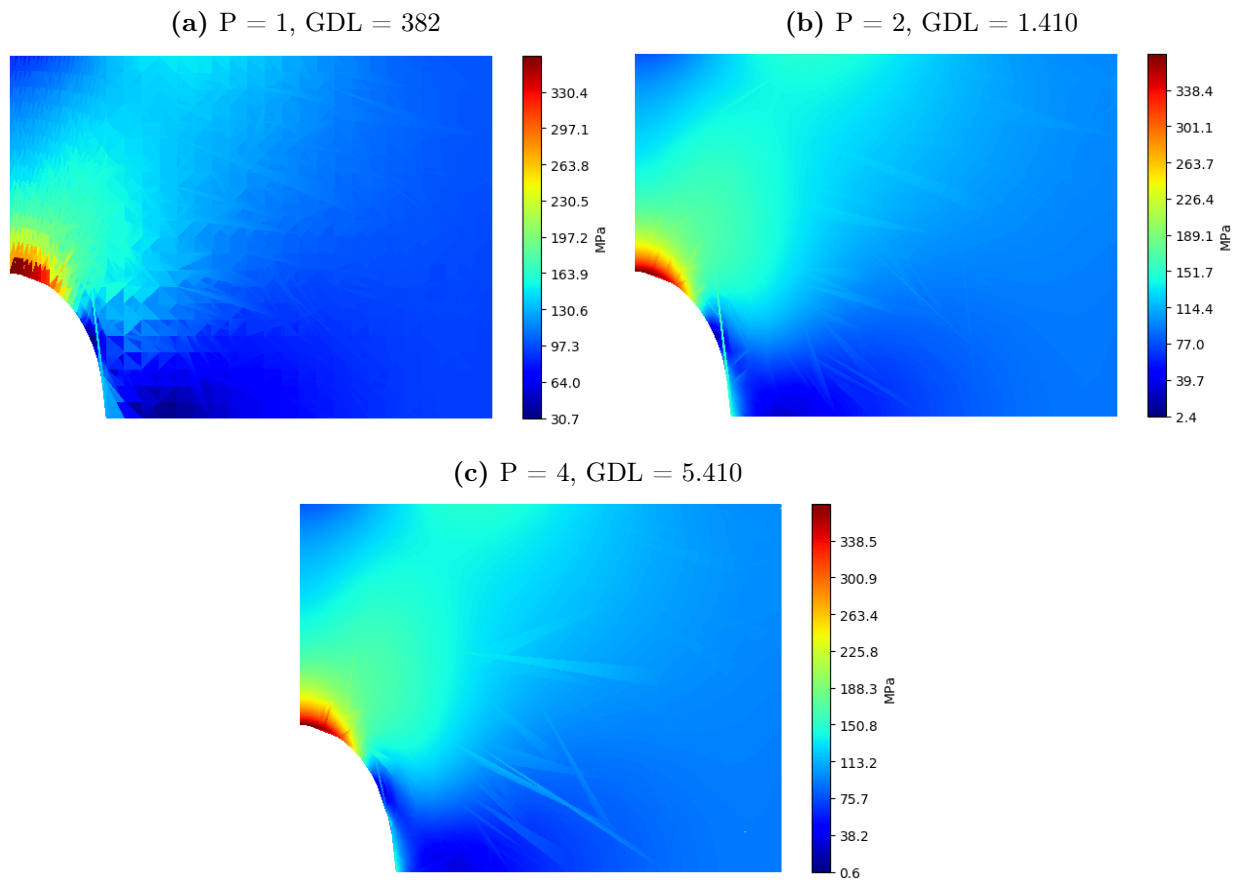
Quanto a suavidade do campo de tensão de Von Mises, os resultados obtidos são semelhantes a simulação do software SimSolid, e se tornando mais homogêneo a medida que aumenta a ordem de interpolação das funções de forma, onde a tensão não é constante ao longo do elemento. Todavia, não é possível fazer outras comparações, pois, o manual não oferece maiores detalhes da malha utilizada, assim como os números de GDLs.

Por outro lado, nota-se diferenças gráficas entre os campos de tensão da Figura 4.24 e como comentado nas outras seções, esse efeito se deve pela presença de gradientes entre um elemento e outro devido aos valores constante de tensão no elemento.

Com o emprego do refinamento  $-h$  e visando criar malhas com o mesmo número de graus de liberdade que as ordens quadrática e quártica da distribuição de tensão anterior, os campos de tensão de Von Mises de saída do algoritmo são mostrados na Figura 4.25. O aumento da suavidade dos campos de tensão de Von Mises propiciado pelo refino  $-h$  é nítido, mas ainda é possível notar que há mudança clara de tensão de um elemento para outro. A minimização desse efeito só seria alcançada reduzindo ainda mais os tamanhos dos elementos ou empregando o uso de funções de alta ordem.

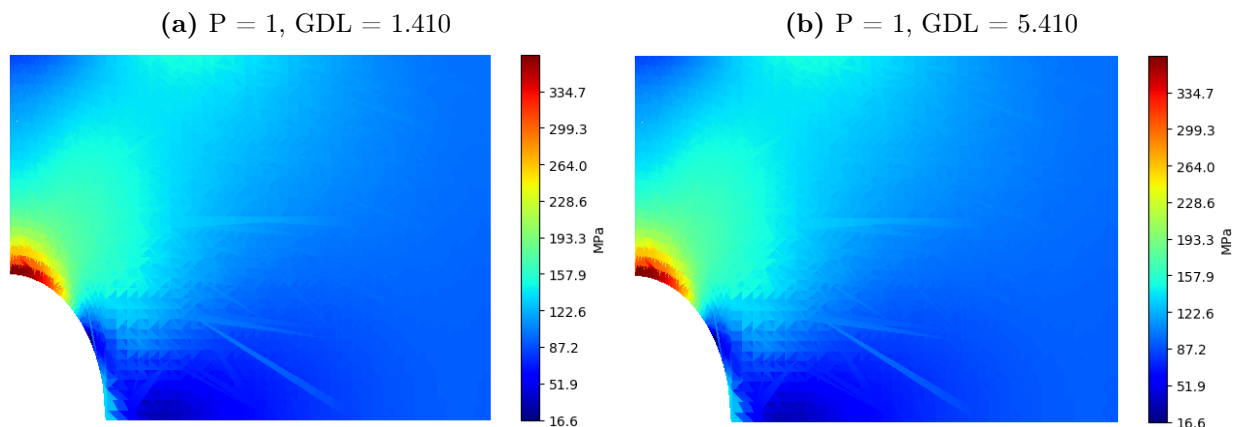


Figura 4.24 – Campos de tensões de Von Mises para a placa retangular com furo e carga axial. A malha para as três situações contém 324 elementos



Fonte – Autoria Própria

Figura 4.25 – Campos de tensões de Von Mises para a placa retangular com furo e carga axial. Refinamento -h

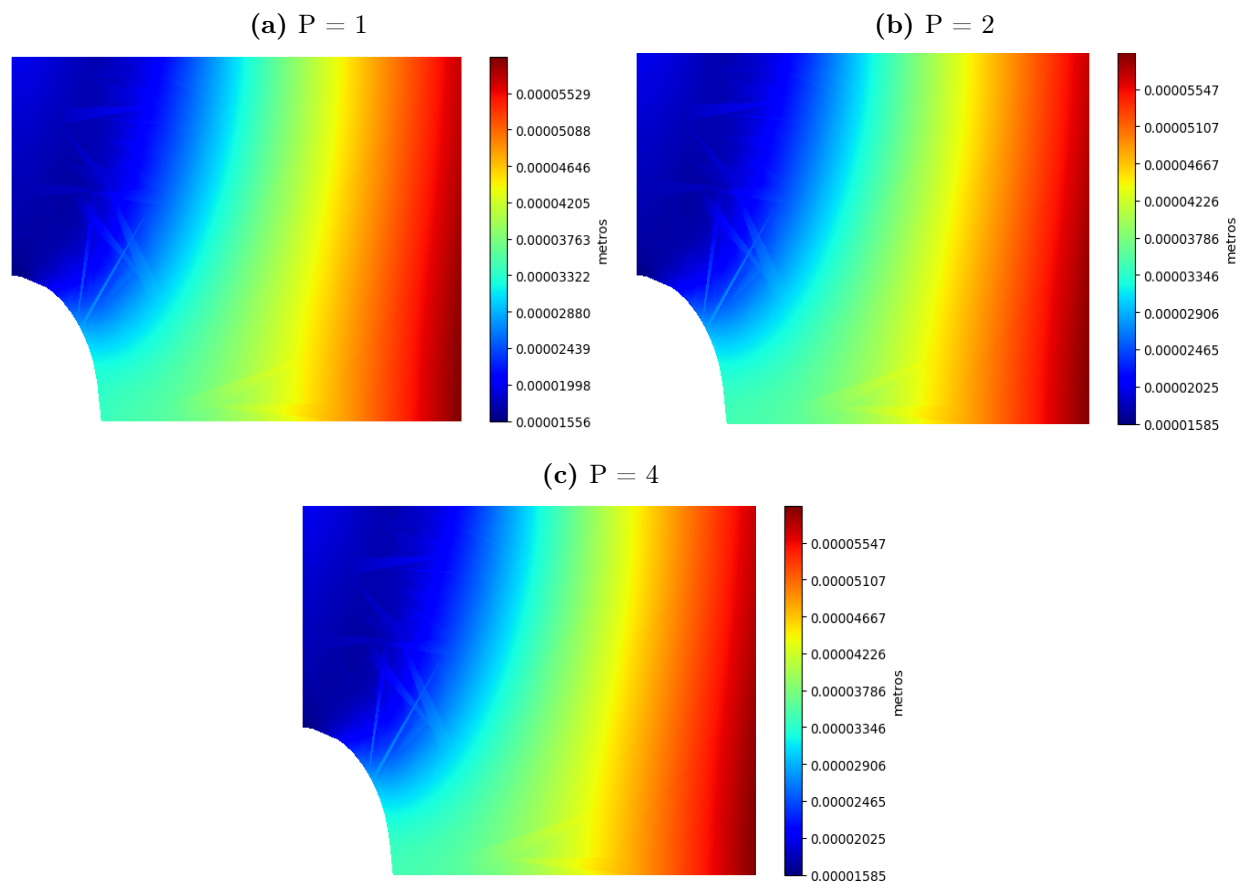


Fonte – Autoria Própria

Os campos de deslocamentos totais obtidos pelo algoritmo, Figura 4.26, são

suaves por toda a extensão da geometria e tem comportamento esperado, com os menores deslocamentos na região de apoio deslizante da lateral esquerda devido as condições de não deslocamento da direção  $u$  e os maiores deslocamentos na porção de material da lateral direita, onde se encontra o carregamento e é a região mais distante de restrições de movimento.

**Figura 4.26 – Campos de deslocamentos totais para a placa retangular com furo e carga axial. A malha para as três situações contém 324 elementos**

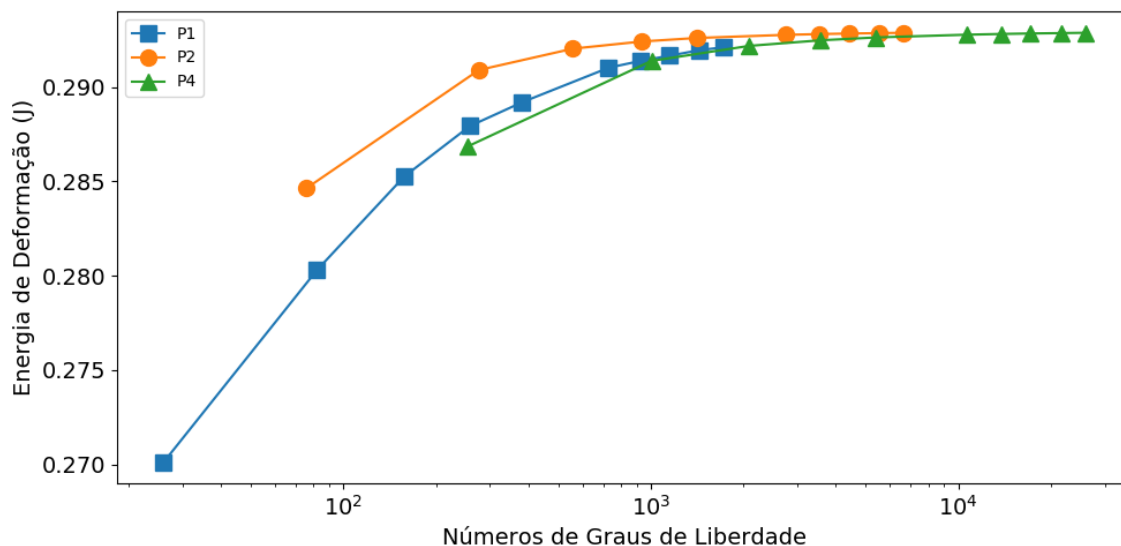


Fonte – Autoria Própria

Já a energia de deformação associada a estrutura,  $\{u\}^T [K] \{u\}$ , nota-se que sua convergência, Figura 4.27, é acelerada ao passo que se aumenta a ordem de interpolação com comportamento monotônico.

Nesta situação, nota-se que as ordens superiores, necessitam de aproximadamente 3.500 graus de liberdade para se estabilizarem em uma energia de deformação de 0,2928 J, enquanto que, para o maior número de GDL do caso linear, 1.718, a o valor assumido pela energia de estabilização é de 0,2921 J.

**Figura 4.27 – Energia de Deformação (J) para o problema da placa com furo central. A taxa de convergência em função de P demonstra comportamento monotônico, com maiores velocidades para  $P > 1$ .**



Fonte – Autoria Própria

Apesar do aumento da convergência da solução a medida que eleva-se a ordem de interpolação, observa-se que a partir de  $P = 2$ , a solução é satisfatória para a maioria dos casos estudados e o custo computacional é progressivo dado o número de graus de liberdade do problema.

Para comparar os resultados obtidos pelo algoritmo, também se resolveu o problema via Ansys com os parâmetros da Tabela 4.7. O software comercial teve como soluções gráficas os campos de tensão e de deslocamentos apresentados na Figura 4.28.

Em comparação dos campos de tensão provenientes do algoritmo e aos obtidos pelo manual de validação, Figura 4.23 e o software Ansys, Figura 4.28a, observa-se grande semelhança gráfica das soluções, apesar da diferença dos mapas de cores utilizados em cada situação.

Por outro lado, o campo de deslocamentos obtidos pelo software também, possuem grande semelhança gráfica e de valores aos do algoritmo. Nesse sentido, a Tabela 4.8 compila uma série de resultados que visa comparar os resultados do algoritmo e do software Ansys.

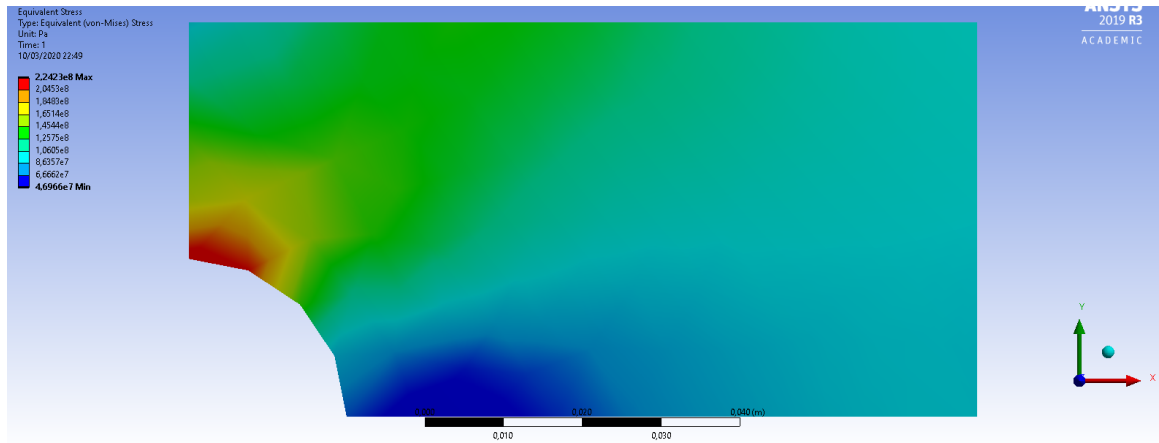
**Tabela 4.8 – Comparativo dos resultados do software e o algoritmo para a placa retangular com furo central**

	Ansys	P1	P1	P1	P2	P4
Tensão máxima (MPa)	224,23	257,6	284,71	323,15	349,66	368,06
Deslocamento máximo (m)	6,1e-5	5,52e-5	5,54e-5	5,6e-5	5,98e-5	5,98e-5
Número de GDL	190	26	76	254	76	254
Erro relativo (%)	39,8	30,9	23,6	13,3	6,18	1,2

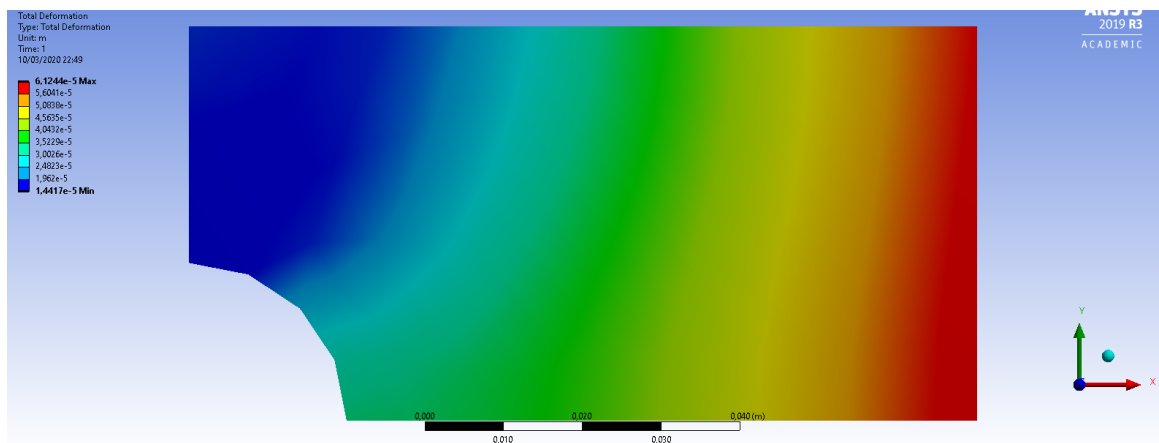
Fonte – Autoria Própria

Figura 4.28 – Campos de tensões de Von Mises e deslocamentos totais para o problema da placa retangular com furo em uma malha de 95 nós e 135 elementos. Resultados obtidos pelo software Ansys 2019 R3 versão estudantil

(a) Campo de Tensões



(b) Campo de Deslocamentos



Fonte – Autoria Própria

Observa-se pelos resultados apresentados na Tabela 4.8 que, para a tensão máxima de Von Mises, com um número 60% inferior de graus de liberdades para  $P = 2$  e um número 36,7% superior de GDL para  $P = 4$  comparado ao Ansys, as soluções para as funções de alta ordem tem um erro relativo ao resultado analítico de 6,18% e 1,2%, respectivamente, enquanto o Ansys traz erro relativo de 39,8% para a tensão máxima.

Nas situações em que se realizou o refinamento -h, o erro relativo reduziu para 23,6% e 13,3% respectivamente, mas ainda assim, são maiores que as soluções para as ordens  $P = 2$  e 4. Quanto aos deslocamentos máximos totais, não foi notado sensibilidade dos resultados em relação a variação de ordem de interpolação e refinamento.

As diferenças de tensão e deslocamento apresentadas podem ter diversas razões, a

exemplo da disposição dos nós que ocasionam erros geométricos. Ademais, a abordagem utilizada na criação do modelo, o método de solução e até mesmo, a versão do software podem ter influência nos resultados.

Nesse sentido, o modelo do Método dos Elementos Finitos de Alta Ordem possibilitou a resolver problemas de geometrias submetidas a um Estado Plano de Tensão. Nota-se que os resultados obtidos se equiparam aos resultados do software Ansys e, em algumas situações, foram melhores.

O algoritmo também provê as curvas do resíduo do problema,  $\{u\}^T [K] \{u\} - \{u\}^T \{F\}$ , e em todos os estudos de caso as curvas apresentadas foram da ordem de  $10^{-15}$ , próximo da precisão de máquina, satisfazendo o objetivo do trabalho.

Além disso, nota-se que ao subir a ordem de interpolação de  $P = 2$  para  $P = 4$ , apesar da contribuição nítida das funções de alta ordem aos resultados, o custo computacional exigido não é na mesma proporção, uma vez que o número de GDL cresce em taxas não lineares e o algoritmo não foi construído visando eficiência computacional.

# Capítulo 5

## Conclusão

Um modelo do Método de Elementos Finitos de Alta Ordem utilizando bases modais e hierárquicas na construção das funções de forma foi implementado. O modelo alcançou resultados de tensão e deslocamento que, em comparação com o software comercial Ansys, foram satisfatórios.

Destaca-se que o algoritmo foi capaz de solucionar geometrias submetidas a um Estado Plano de Tensão, discretizando-nas por meio de elementos não estruturados, com apoio da biblioteca *Triangle* da linguagem de programação Python. As funções de interpolação de alta ordem demonstraram-se importantes no processo de solução, uma vez que permitiram obter resultados próximos aos do software, sem a necessidade de um refino acentuado das malhas estudadas.

Entretanto, verificou-se que ao passo que cresce a ordem das funções de interpolação, a contribuição das funções de alta ordem não é proporcional ao custo computacional exigido, se tornando proibitivo para geometrias complexas ou, que exijam malhas refinadas uma vez que a implementação do código para funções de interpolação de alta ordem não foi focada na busca de desempenho computacional.

No trabalho apresentado, há oportunidade de realizar diversas implementações em trabalhos futuros. É pertinente a aplicação de elementos estruturados na discretização de geometrias e comparação com elementos não estruturados, onde também é possível variar as representações geométricas, utilizando elementos isoparamétricos e superparamétricos.

Afim de melhorar o desempenho computacional, a aplicação de métodos de decomposição do domínio é uma alternativa na paralelização de algoritmos, e no trabalho de Bittencourt (2014), apresenta-se o complemento de Schur, método que visa melhorar a esparsidade e o condicionamento numérico para aproximações de alta ordem.

Nesse sentido, o objetivo de desenvolver um algoritmo em Python para análise de problemas em Estado Plano de Tensão usando o Método dos Elementos Finitos de Alta Ordem foi bem sucedido, uma vez que os resultados obtidos e apresentados são satisfatórios.

# Referências

ANSYS. *About Ansys*. 2020. Disponível em: <<https://www.ansys.com/about-ansys>>.

BABUŠKA, I.; GUO, B. The h, p and hp version of the finite element method; basis theory and applications. *Advances in Engineering Software*, Elsevier, v. 15, n. 3-4, p. 159–174, 1992.

BITTENCOURT, M.; VAZQUEZ, M.; VAZQUEZ, T. Construction of shape functions for the h-and p-versions of the fem using tensorial product. *International journal for numerical methods in engineering*, Wiley Online Library, v. 71, n. 5, p. 529–563, 2007.

BITTENCOURT, M. L. Introdução ao método de elementos finitos aplicado à análise estrutural—exemplos com o programa ansys. *Campinas: Unicamp*, 2007.

BITTENCOURT, M. L. *Computational solid mechanics: Variational formulation and high order approximation*. [S.l.]: CRC Press, 2014.

CHANDRUPATLA, T. R. et al. *Introduction to finite elements in engineering*. [S.l.]: Prentice Hall Upper Saddle River, NJ, 2002. v. 2.

COOK, R. D. *Finite element modeling for stress analysis*. [S.l.]: Wiley, 1994.

DHATT, G.; LEFRANÇOIS, E.; TOUZOT, G. *Finite element method*. [S.l.]: John Wiley & Sons, 2012.

EDGAR, N.; SURANA, K. On the conditioning number and the selection criteria for p-version approximation functions. *Computers & structures*, Elsevier, v. 60, n. 4, p. 521–530, 1996.

FISH, J.; BELYTSCHKO, T. *A first course in finite elements*. [S.l.]: John Wiley & Sons New York, 2007. v. 1.

FOUNDATION, P. S. *About Python*. 2020. Disponível em: <<https://www.python.org/about/>>.

HIBBELER, R. *Mechanics of Materials*. Prentice Hall, 2011. ISBN 9780136022305. Disponível em: <<https://books.google.com.br/books?id=HikDQgAACAAJ>>.

HUNTER, J. D. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, IEEE COMPUTER SOC, v. 9, n. 3, p. 90–95, 2007.

KARNIADAKIS, G.; SHERWIN, S. *Spectral/hp element methods for computational fluid dynamics*. [S.l.]: Oxford University Press, 2013.

REDDY, J. N. *An introduction to the finite element method*. [S.l.]: McGraw-hill New York, 1993. v. 2.

RUFAT, D. *Triangle*. 2009. Disponível em: <<https://rufat.be/triangle/>>.

SIMSOLID. *Validation Manual*. [S.l.]: INNEO Handleinformation, 2015.

SMITH, I. M.; GRIFFITHS, D. V.; MARGETTS, L. *Programming the finite element method*. [S.l.]: John Wiley & Sons, 2013.



# APÊNDICE A

Código implementado em *Python*

In [1]:

```
## BIBLIOTECAS NECESSÁRIAS
import numpy as np
import sympy as se
from scipy import special as sp
import quadpy

class FEM:
    def __init__(self, degree=1, coordinates=0, connectivity=0, cardinality=0, ndof = 0
, loads=0, boundary=0, poison=.3, young_modulus=210e9, thickness=1e-3, alpha = 1, beta
= 1):
        self.degree = degree
        self.alpha = alpha
        self.beta = beta
        self.coordinates = coordinates
        self.connectivity = connectivity
        self.cardinality = cardinality
        self.ndof = ndof
        self.loads = loads
        self.boundary = boundary
        self.poison = poison
        self.young_modulus = young_modulus
        self.thickness = thickness

    # DEFINE A MATRIZ CONSTITUTIVA DO MATERIAL
    def __constitutive(self):
        A = (self.young_modulus/(1-self.poison**2))
        return A * np.array([[1., self.poison, 0], [self.poison, 1., 0], [0, 0, .5*(1
-self.poison)]])

    # VARIÁVEIS SIMBÓLICAS QUE CORRESPONDEM AS COORDENADAS NATURAIS
    L1, L2, L = se.var('L1 L2 L')

    # FUNÇÃO PARA CRIAR AS FUNÇÕES DE FORMA E SUAS DERIVADAS
    def shape_function(self):

        # GERA O POLINÔMIO DE JACOBI
        def Phi(L, index):
            Pi = sp.jacobi(int(index), self.alpha, self.beta)
            Natu = 0
            for i in range(0,len(Pi.c)):
                Natu = Natu + Pi.c[i]*(2*L-1)**(len(Pi.c)-i-1)
            return Natu

        # GERA AS BASES MODAIS EM CADA COORDENADA NATURAL
        N , N_L1, N_L2, N_L3 = list(), list(), list(), list()
        N_L1.append(1), N_L2.append(1), N_L3.append(1)
        for i in range(0,self.degree-1):
            N_L1.append(se.expand(L1*Phi(L1,i)))
            N_L2.append(se.expand(L2*Phi(L2,i)))
            N_L3.append(se.expand((1- L1 - L2)*Phi(1- L1 - L2,i)))
        N_L1.append(L1), N_L2.append(L2), N_L3.append(1- L1 - L2)

        # GERA AS FUNÇÕES DE FORMA
        # FUNÇÕES DE VÉRTICE
        N.append(N_L1[1]),N.append(N_L2[1]),N.append(N_L3[1])

        # FUNÇÕES DE FORMA ARESTA OCORREM PARA GRAU >= 2
        if self.degree >= 2:
```

```

    for p in range(1,self.degree):
        q = self.degree - p
        N.append(N_L1[p]*N_L2[q])
    for q in range(1,self.degree):
        r = self.degree - q
        N.append(N_L2[q]*N_L3[r])
    for p in range(1,self.degree):
        r = self.degree - p
        N.append(N_L1[p]*N_L3[r])

# FUNÇÕES DE FORMA DE FACE OCORREM PARA GRAU >=3
if self.degree >=3:
    for r in range(0,self.degree):
        for q in range(0, self.degree):
            for p in range(0, self.degree):
                if (p+q+r+3 == self.degree):
                    N.append(N_L1[p+1]*N_L2[q+1]*N_L3[r+1])

# DERIVADAS DAS FUNÇÕES DE FORMA
derivatives = se.zeros(len(N),2)
for i in range(0,len(N)):
    derivatives[i,0], derivatives[i,1] = se.expand(se.diff(N[i],L1)), se.expand
(se.diff(N[i],L2))

    return N, derivatives

# PROCESSO ANÁLOGO PARA GERAR AS FUNÇÕES DE FORMA UNIDIMENSIONAIS PARA CÁLCULO DO V
ETOR CARREGAMENTO
def SF_1d(self):

    # GERA O POLINÔMIO DE JACOBI
    def Phi(L, index):
        Pi = sp.jacobi(int(index), self.alpha, self.beta)
        Natu = 0
        for i in range(0,len(Pi.c)):
            Natu = Natu + Pi.c[i]*(2*L-1)**(len(Pi.c)-i-1)
        return Natu

    # GERA AS BASES MODAIS IMPONDO L2 = 0
    N , N_L1, N_L2, N_L3 = list(), list(), list(), list()
    N_L1.append(1), N_L2.append(1), N_L3.append(1)
    for i in range(0,self.degree-1):
        N_L1.append(se.expand(L1*Phi(L1,i)))
        N_L2.append(se.expand(0*Phi(0,i)))
        N_L3.append(se.expand((1- L1 - 0)*Phi(1- L1 - 0,i)))
    N_L1.append(L1), N_L2.append(0), N_L3.append(1- L1 - 0)

    # GERA AS FUNÇÕES DE FORMA
    # FUNÇÕES DE FORMA DE VÉRTICE
    N.append(N_L1[1]),N.append(N_L2[1]),N.append(N_L3[1])

# FUNÇÕES DE FORMA DE ARESTA OCORREM PARA GRAU >=2
if self.degree >= 2:
    for p in range(1,self.degree):
        q = self.degree - p
        N.append(N_L1[p]*N_L2[q])
    for q in range(1,self.degree):
        r = self.degree - q
        N.append(N_L2[q]*N_L3[r])
    for p in range(1,self.degree):
        r = self.degree - p

```

```

        N.append(N_L1[p]*N_L3[r])

# FUNÇÕES DE FORMA DE FACE OCORREM PARA GRAU >=3
if self.degree >=3:
    for r in range(0,self.degree):
        for q in range(0, self.degree):
            for p in range(0, self.degree):
                if (p+q+r+3 == self.degree):
                    N.append(N_L1[p+1]*N_L2[q+1]*N_L3[r+1])

# DERIVADAS DAS FUNÇÕES DE FORMA
N_1d = list(i for i in N if i !=0.0)
derivatives_1d = se.zeros(len(N_1d),1)

for i in range(0,len(N_1d)):
    derivatives_1d[i,0] = se.expand(se.diff(N_1d[i],L1))

return N_1d, derivatives_1d

# SOLUCIONA A MATRIZ RIGIDEZ DO PROBLEMA / SEM CONDIÇÕES DE CONTORNO
def __stiffness(self):
    material = FEM.__constitutive(self)
    SF, DER = FEM.shape_function(self)
    # GERA A MATRIZ RIGIDEZ MXM COM M =NÚMERO TOTAL DE GRAUS DE LIBERDADE
    K = np.zeros((self.ndof,self.ndof))

    # COORDENADAS E PESOS PARA A INTEGRAÇÃO NUMÉRICA
    x_ξ = quadpy.line_segment.gauss_lobatto(2*self.degree + 1, a = 1, b = 0).points
    w_ξ = quadpy.line_segment.gauss_lobatto(2*self.degree + 1, a = 1, b = 0).weight

    x_η = quadpy.line_segment.gauss_lobatto(2*self.degree + 1, a = 0, b = 0).points
    w_η = quadpy.line_segment.gauss_lobatto(2*self.degree + 1, a = 0, b = 0).weight

    # LOOP NO NÚMERO DOS ELEMENTOS
    for i in range(0, len(self.connectivity)):
        # PROCURA AS COORDENADAS DO ELEMENTO EM QUESTÃO
        x1, y1 = self.coordinates[self.connectivity[i][0]][0], self.coordinates[self
f.connectivity[i][0]][1]
        x2, y2 = self.coordinates[self.connectivity[i][1]][0], self.coordinates[self
f.connectivity[i][1]][1]
        x3, y3 = self.coordinates[self.connectivity[i][2]][0], self.coordinates[self
f.connectivity[i][2]][1]

        # SOLUCIONA O DETERMINANTE E A MATRIZ JACOBIANA INVERSA / TRANSFORMAÇÃO SUB
PARAMÉTRICA
        detJ = (x1-x3)*(y2-y3)-(x2-x3)*(y1-y3)
        IJac = (1/detJ)* np.array([[ (y2-y3), (y3-y1) ], [ (x3-x2), (x1-x3) ]])
        ij11, ij12, ij21, ij22 = IJac[0][0], IJac[0][1], IJac[1][0], IJac[1][1]

        DSF = se.Lambdify([L1,L2], DER)
        L1n = lambda ξ,η: .5*(1+ξ)
        L2n = lambda ξ,η: .25*(1-ξ)*(1+η)

        # INICIA A MATRIZ DE RIGIDEZ E ÁREA ELEMENTAR
        K_elem, area = np.zeros((2*len(SF),2*len(SF))), 0

        # RESOLVE A MATRIZ DE ELASTICIDADE E A MATRIZ DE RIGIDEZ ELEMENTAR EM CADA
PONTO DE INTEGRAÇÃO
        for j in range(0,len(x_ξ)):
            for k in range(0,len(x_η)):

```

```

L1A, L2A = L1n(x_ξ[j],x_η[k]), L2n(x_ξ[j],x_η[k])
B_elem = np.zeros((3,2*len(SF)))
for l in range(0,len(SF)):
    dndx= ij11*DSF(L1A,L2A)[1][0] + ij12*DSF(L1A,L2A)[1][1]
    dndy= ij21*DSF(L1A,L2A)[1][0] + ij22*DSF(L1A,L2A)[1][1]
    B_elem[0][2*1], B_elem[0][2*1+1] = dndx, 0
    B_elem[1][2*1], B_elem[1][2*1+1] = 0, dndy
    B_elem[2][2*1], B_elem[2][2*1+1] = dndy, dndx

    K_elem = K_elem + abs(detJ)*np.dot(np.dot(np.transpose(B_elem),mate
rial),B_elem)*(w_ξ[j]*w_η[k]/8)
    area = area + .5*abs(detJ)

    # ENCONTRA A CARDINALIDADE DO ELEMENTO
    LOCE = (self.cardinality[:,i]).astype(int)

    # REALIZA A MONTAGEM DA MATRIZ RIGIDEZ ELEMENTAR NA MATRIZ GLOBAL DE RIGIDE
Z
    for n in range(0, len(LOCE)):
        for m in range(0,len(LOCE)):
            K[LOCE[n]][LOCE[m]] = K[LOCE[n]][LOCE[m]] + K_elem[n][m]
return self.thickness*K

# RESOLVE O VETOR DE CARGA / APENAS CARGAS UNIFORMES
def __distributed_loads(self):
    # COORDENADAS E PESOS PARA A INTEGRAÇÃO NUMÉRICA
    x_ξ = quadpy.line_segment.gauss_lobatto(2*self.degree + 1, a = 1, b = 0).points
    w_ξ = quadpy.line_segment.gauss_lobatto(2*self.degree + 1, a = 1, b = 0).weight
S
    x_η = quadpy.line_segment.gauss_lobatto(2*self.degree + 1, a = 0, b = 0).point
S
    w_η = quadpy.line_segment.gauss_lobatto(2*self.degree + 1, a = 0, b = 0).weight
S

    # INICIA O VETOR DE CARREGAMENTO COM MX1 COM M=NÚMERO DE GRAU DE LIBERDADES
    F = np.zeros((self.ndof,1))
    shape_functions_1d, derivatives_1d = FEM.SF_1d(self)

    # LOOP PARA CALCULAR O CARREGAMENTO
    for i in range(0,len(self.loads)):
        # SE O CARREGAMENTO É PARALELO A DIREÇÃO X
        if self.loads[i][1] == 1:
            N_1d = se.zeros(np.size(self.cardinality,0),1)
            element, index_a, index_b = self.connectivity[self.loads[i][0]].tolist
            self.loads[i][2],self.loads[i][3]
            node_1, node_2, = int(element.index(index_a)), int(element.index(ind
ex_b))

            sum_index = node_1 + node_2
            # MONTA A MATRIZ DE FUNÇÕES DE FORMA BASEADO NA ARESTA DO ELEMENTO QUE
CONTÉM O CARREGAMENTO
            if sum_index == 1:
                N_1d[2*node_1], N_1d[2*node_2] = shape_functions_1d[0], shape_funct
ions_1d[1]

                for j in range(0,self.degree-1):
                    N_1d[6+2*j] = shape_functions_1d[j+2]
            if sum_index == 3:
                N_1d[2*node_1], N_1d[2*node_2] = shape_functions_1d[0], shape_funct
ions_1d[1]

                for j in range(0,self.degree-1):
                    N_1d[6+2*(self.degree-1)+2*j] = shape_functions_1d[j+2]

```

```

        if sum_index == 2:
            N_1d[2*node_1], N_1d[2*node_2] = shape_functions_1d[0], shape_funct
ions_1d[1]

            for j in range(0,self.degree-1):
                N_1d[6+4*(self.degree-1)+2*j] = shape_functions_1d[j+2]

            # CALCULA O DETERMINANTE, INICIA O VETOR DE CARGA DE LINHA ELEMENTAR E
            DETERMINA A CARGA UNIFORME
            det_J = .5*abs(-self.coordinates[index_a][1]+self.coordinates[index_b][
1])

            line_load = np.zeros((np.size(self.cardinality,0), 1))
            value_load = self.loads[i][4]
            N_n1d = se.Lambdify(L1,N_1d)

            # CALCULA O VETOR DE CARGA DE LINHA
            for k in range(0, len(x_η)):
                line_load = line_load + w_η[k]*det_J*value_load*np.array(N_n1d(.5*(
1+x_η[k]))) .astype(np.float64)

            # PROCURA A CARDINALIDADE DO ELEMENTO
            LOCE = (self.cardinality[:,self.loads[i][0]]).astype(int)

            # REALIZA A MONTAGEM DO VETOR DE CARGA DE LINHA NO VETOR GLOBAL DE CARR
EGAMENTO

            for k in range(0,len(LOCE)):
                F[LOCE[k]] = F[LOCE[k]] + line_load[k]

            # SE O CARREGAMENTO É PARALELO A DIREÇÃO Y
            elif self.loads[i][1] == 2:
                N_1d = se.zeros(np.size(self.cardinality,0),1)
                element, index_a, index_b = self.connectivity[self.loads[i][0]].tolist
(,),self.loads[i][2],self.loads[i][3]
                node_1, node_2, = int(element.index(index_a)), int(element.index(ind
ex_b))

                sum_index = node_1 + node_2

            # MONTA A MATRIZ DE FUNÇÕES DE FORMA BASEADO NA ARESTA DO ELEMENTO QUE
            CONTÉM O CARREGAMENTO
            if sum_index == 1:
                N_1d[2*node_1+1], N_1d[2*node_2+1] = shape_functions_1d[0], shape_f
unctions_1d[1]

                for j in range(0,self.degree-1):
                    N_1d[6+2*j+1] = shape_functions_1d[j+2]

            if sum_index == 3:
                N_1d[2*node_1+1], N_1d[2*node_2+1] = shape_functions_1d[0], shape_f
unctions_1d[1]

                for j in range(0,self.degree-1):
                    N_1d[6+2*(self.degree-1)+2*j+1] = shape_functions_1d[j+2]

            if sum_index == 2:
                N_1d[2*node_1+1], N_1d[2*node_2+1] = shape_functions_1d[0], shape_f
unctions_1d[1]

                for j in range(0,self.degree-1):
                    N_1d[6+4*(self.degree-1)+2*j+1] = shape_functions_1d[j+2]

            # CALCULA O DETERMINANTE, INICIA O VETOR DE CARGA DE LINHA ELEMENTAR E
            DETERMINA A CARGA UNIFORME
            det_J = .5*abs(-self.coordinates[index_a][0]+self.coordinates[index_b][
0])

            line_load = np.zeros((np.size(self.cardinality,0), 1))
            value_load = self.loads[i][4]
            N_n1d = se.Lambdify(L1,N_1d)

```

```

        # CALCULA O VETOR DE CARGA DE LINHA
        for k in range(0, len(x_ξ)):
            line_load = line_load + w_ξ[k]*det_J*value_load*np.array(N_n1d(.5*(
1+x_ξ[k]))) .astype(np.float64)

        # PROCURA A CARDINALIDADE DO ELEMENTO
        LOCE = (self.cardinality[:,loads[i][0]]).astype(int)

        # REALIZA A MONTAGEM DO VETOR DE CARGA DE LINHA NO VETOR GLOBAL DE CARR
EGAMENTO

        for k in range(0,len(LOCE)):
            F[LOCE[k]] = F[LOCE[k]] + line_load[k]

    return F

# APLICA AS CONDIÇÕES DE CONTORNO NA MATRIZ DE RIGIDEZ E O VETOR DE CARREGAMENTO
def __boundary_conditions(self,K,F):
    for i in range(0,len(self.boundary)):
        for j in range(0,2):
            if self.boundary[i][j+1] == 1:
                idiri = int(2*self.boundary[i][0] + j)
                a = self.boundary[i][3+j]
                for k in range(0, self.ndof):
                    guardk = K[k][idiri]
                    K[idiri][k] = 0
                    K[k][idiri] = 0
                    F[k] = F[k] - guardk*a

                K[idiri][idiri] = 1
                F[idiri][0] = a

    return K,F

# SOLUCIONA O PROBLEMA E DEVOLVE O CAMPO DE DESLOCAMENTOS U = U(u,v)
def solve(self):
    K = FEM.__stiffness(self)
    F = FEM.__distributed_loads(self)
    modified_K, modified_F = FEM.__boundary_conditions(self,K,F)
    displacements = np.dot(np.linalg.inv(modified_K),modified_F)
    return displacements

```