

Criação de programa informático em ambiente Matlab® para estudo de estruturas planas pelo Método de Elementos Finitos.

PEDRO PALMEIRA LÁZARO

Outubro de 2015

Criação de programa informático em ambiente
Matlab[®] para estudo de estruturas planas pelo Método
de Elementos Finitos.

Pedro Palmeira Lázaro

Dissertação submetida para a obtenção do grau de Mestre em
Engenharia Mecânica

Instituto Superior de Engenharia do Porto

Departamento de Engenharia Mecânica



16 de Outubro de 2015

Relatório da Unidade Curricular de Dissertação/Projecto/Estágio do 2º ano do Mestrado em
Engenharia Mecânica

Candidato: Pedro Palmeira Lázaro, N°1110979, 1110979@isep.ipp.pt

Orientação Científica: Raul Duarte Salgueiral Gomes Campilho, rds@isep.ipp.pt

Mestrado em Engenharia Mecânica

Departamento de Engenharia Mecânica

Instituto Superior de Engenharia do Porto



16 de Outubro de 2015

À mãe, bebé e Nico.

Agradecimentos

Começo por agradecer ao professor e orientador de projeto, Prof. Raul Duarte Campilho, pela sua disponibilidade em resolver as minhas questões prontamente. Este projeto teve muitos obstáculos, mas o professor esteve sempre disponível quando precisei de um empurrão na direção certa. Quando não empurrava, juntava-se a mim e juntos, de algum modo, conseguíamos ultrapassar mais um obstáculo.

Agradeço também aos meus pais, que sempre insistiram que eu terminasse este percurso e tentaram que não me desviasse deste.

Um especial obrigado à minha namorada, por me ter apoiado nas alturas que faltava a vontade, sempre a meu lado.

Por último, devo agradecer a todos os programadores, que ao longo dos anos foram ajudando responder a dúvidas e problemas nos fóruns *online*. Sem o contributo deles, este trabalho nunca estaria concluído com a qualidade aqui mostrada.

Resumo

Os modelos a ser analisados pelo Método de Elementos Finitos são cada vez mais complexos e, nos tempos que correm, seria impensável realizar tais análises sem um apoio computadorizado. Existe para esta finalidade uma vasta gama de programas que permitem realizar tarefas que passam pelo desenho de estruturas, análise estática de cargas, análise dinâmica e vibrações, visualização do comportamento físico (deformações) em tempo real, que permitem a otimização da estrutura.

Sob o pretexto de permitir a qualquer utilizador uma análise de estruturas simples com o Método dos Elementos Finitos, surge esta tese, onde se irá criar de raiz um programa com interface gráfica no ambiente MATLAB[®] para análise de estruturas simples com dois tipos de elemento finito, triangular de deformação constante e quadrangular de deformação linear.

O *software* desenvolvido, verificado por comparação com um *software* comercial dedicado para o efeito, efetua malhagem com elementos bidimensionais triangulares e quadriláteros e resolve modelos arbitrados pelo Método de Elementos Finitos, representando estes resultados visualmente e em formato tabular.

Palavras-Chave

EPT, EPD, estado plano de tensão, estado plano de deformação, método, elementos finitos, placas finas, MATLAB[®], ABAQUS[®] código, interface gráfica, comparação.

Abstract

Models analyzed by the Finite Element Method are increasingly complex and, nowadays, it would be unthinkable to do such analyses without digital computer support. With this purpose, a vast array of programs is available that enable conducting tasks including structural design, static and dynamic loading and vibration analysis, visualization of the deformed shape in real time, which allows the optimization of the structure.

With the idea of allowing any user to perform a simple structural analysis with the Finite Element Method in mind, this thesis is proposed to create a graphical interface in the MATLAB[®] environment to study such structures with two types of finite element, constant strain triangle and linear strain quadrilateral.

The developed software, verified by comparison with a dedicated software for this analysis, meshes the models with bidimensional triangular and quadrilateral elements, solving them using the Finite Element Method, with a visual and table representation.

Keywords

PS, PE, Plain stress, Plane strain, method, finite elements, thin plates, MATLAB[®], ABAQUS[®], code, graphical interface, comparison.

Índice

AGRADECIMENTOS	VII
RESUMO	IX
ABSTRACT	XI
ÍNDICE DE FIGURAS	XV
ÍNDICE DE TABELAS	XVII
NOMENCLATURA	XIX
1 INTRODUÇÃO	1
1.1 CONTEXTUALIZAÇÃO	1
1.2 OBJETIVOS	1
1.3 ORGANIZAÇÃO DO RELATÓRIO	2
2 REVISÃO BIBLIOGRÁFICA	5
2.1 O MÉTODO DE ELEMENTOS FINITOS	5
2.1.1 <i>A evolução do Método de Elementos Finitos</i>	6
2.1.2 <i>Princípios gerais do Método de Elementos Finitos</i>	8
2.1.2.1 Método das Forças	10
2.1.2.2 Método dos Deslocamentos	10
2.1.2.2.1 Sistema de Equações Elementares	12
2.1.2.2.2 Restrições e Carregamentos	15
2.1.2.2.3 Tipos de elementos finitos	17
2.1.3 <i>Potencialidades</i>	20
2.1.4 <i>Aplicações do Método de Elementos Finitos</i>	21
2.2 <i>SOFTWARES DISPONÍVEIS PARA ANÁLISE ESTRUTURAL</i>	24
2.2.1 <i>Softwares comerciais de uso genérico</i>	25
2.2.2 <i>Códigos específicos</i>	29
2.3 <i>SOFTWARES DE COMPUTAÇÃO E VISUALIZAÇÃO GRÁFICA</i>	31
2.3.1 <i>Características gerais dos softwares existentes no mercado</i>	31
2.3.2 <i>Seleção do software a utilizar no desenvolvimento deste trabalho</i>	33
2.3.3 <i>Funcionalidades pretendidas para o software a desenvolver</i>	33
3 FORMULAÇÃO DOS ELEMENTOS FINITOS A IMPLEMENTAR	35
3.1 TEORIA BÁSICA DA ELASTICIDADE	35
3.1.1 <i>Elasticidade convencional</i>	36
3.1.1.1 Relações tensão-deformação	36
3.1.1.1.1 Caso tridimensional	36

3.1.1.1.2	Caso bidimensional	38
3.1.2	<i>Relações deformação-deslocamento</i>	40
3.1.3	<i>Equações de equilíbrio</i>	42
3.2	ELEMENTOS BIDIMENSIONAIS	43
3.2.1	<i>Elemento triangular de deformação constante de 3 nós</i>	44
3.2.1.1	Matriz de rigidez do elemento – aproximação formal	44
3.2.1.2	Matriz de rigidez do elemento – aproximação isoparamétrica	48
3.2.1.3	Carregamentos equivalentes	52
3.2.1.4	Estado de tensão nos elementos	53
3.2.2	<i>Elemento quadrangular de deformação linear de 4 nós</i>	53
3.2.2.1	Matriz de rigidez do elemento – aproximação isoparamétrica	54
3.2.2.2	Integração numérica	59
3.2.2.3	Carregamentos equivalentes	63
3.2.2.4	Estado de tensão nos elementos	63
4	DESENVOLVIMENTO DO SOFTWARE	65
4.1	ESTRUTURA	65
4.2	MÉTODOS DE PROGRAMAÇÃO	66
4.3	BASES DE DADOS	67
4.4	FUNCIONAMENTO DO SOFTWARE	68
4.4.1	<i>Arranque do programa</i>	69
4.4.2	<i>Escolha do tipo de malhagem</i>	69
4.4.2.1	Criação geometria	70
4.4.2.2	Escolha da geometria	73
4.4.3	<i>Condições fronteira e carregamentos</i>	75
4.4.4	<i>Malhagem e propriedades do material</i>	77
4.4.5	<i>Apresentação de resultados</i>	79
4.5	VALIDAÇÃO DO SOFTWARE DESENVOLVIDO	80
4.5.1	<i>Elementos bidimensionais triangulares</i>	81
4.5.2	<i>Elementos bidimensionais quadriláteros</i>	82
5	CONCLUSÕES	85
6	BIBLIOGRAFIA	87
	ANEXOS	93
	ANEXO A - TIPOS DE ELEMENTOS FINITOS E SUAS APLICAÇÕES. [8]	94
	ANEXO B - RESULTADOS DOS TESTES EFETUADOS	95

Índice de Figuras

Figura 1: Problema de treliça convertido para análise utilizando o MEF.	11
Figura 2: Decomposição do modelo em elementos.....	12
Figura 3: Variáveis e equações fundamentais da Mecânica Estrutural [8].....	13
Figura 4: Montagem e resolução para os deslocamentos nodais com o MEF [10].....	16
Figura 5: Elementos unidimensionais [4].....	17
Figura 6: Elementos bi-dimensionais [11].....	18
Figura 7: Elementos axissimétricos triangulares e quadrangulares.....	18
Figura 8: Elementos tri-dimensionais e quantidades de nós [4].....	19
Figura 9: Painel de aço sob ação do vento analisado pelo MEF. [16]	21
Figura 10: Simulação com recurso ao MEF para calcular a carga térmica de uma câmara de combustão. [17].....	22
Figura 11: Comportamento de uma câmara de pressurização em análise de MEF. [18] ...	22
Figura 12: Campos electromagnéticos entre bobinas de indução com o MEF [19].....	23
Figura 13: Canal micro-fluídico com escoamento não-newtoniano. [20].....	23
Figura 14: Esquema de funcionamento do <i>software</i> desenvolvido.	66
Figura 15: Escolha do tipo de malha.	70
Figura 16: Janela de criação de geometria.....	71
Figura 17: Demonstração da introdução de coordenadas.....	71
Figura 18: Introdução de coordenadas.....	72
Figura 19: Resultado visual da introdução de coordenadas.	72
Figura 20: Aviso quando pressionar o botão “Anterior”.	73
Figura 21: Janela de escolha de geometria.	73
Figura 22: Quadriláteros disponíveis para análise.....	74
Figura 23: Atribuição de restrições nas linhas de fronteira.....	75

Figura 24: Representação dos diferentes tipos de restrição.	76
Figura 25: Painel de seleção da direção da força a aplicar.....	76
Figura 26: Aplicação de força num ponto da área de visualização.....	77
Figura 27 a) Propriedades da malha para análise com elementos triangulares e b) Propriedades da malha para análise com elementos quadrangulares.....	78
Figura 28: Painel de Propriedades do Material.	78
Figura 29: Área de visualização com resultados pós-computação.....	79
Figura 30: Janela de resultados da análise.	80
Figura 31: Correlação entre resultados obtidos em MATLAB® e ABAQUS®.....	82
Figura 32 a) Figura com 384 elementos em MATLAB® e b) Figura com 367 elementos em ABAQUS®.....	82
Figura 33: Correlação entre resultados obtidos em MATLAB® e ABAQUS®.....	83
Figura 34 a) Figura com 54 elementos em MATLAB® e b) Figura com 48 elementos em ABAQUS®.....	84

Índice de Tabelas

Tabela 1: Comparação de resultados entre o *software* desenvolvido e o ABAQUS® 81

Tabela 2: Comparação de resultados entre o *software* desenvolvido e o ABAQUS® 83

Nomenclatura

Caracteres Romanos

A – Vetor de deslocamentos global

a_e – campo de deslocamentos

B – Matriz que relaciona as deformações com as variáveis nodais

D – Matriz das propriedades do material

E – Módulo de Young

F e f_e – Vetor de forças nodais

f_x – Força por unidade de volume segundo eixo x

F_x – Força segundo eixo x

f_y – Força por unidade de volume segundo eixo y

F_y – Força segundo eixo y

f_z – Força por unidade de volume segundo eixo z

F_z – Força segundo eixo z

G – Matriz auxiliar para determinação da matriz de deformação

G – Módulo de elasticidade transversal

J – Matriz de transformação

k – Matriz de rigidez local

K e k_e – Matriz de rigidez global

M_x, M_y e M_z – Momentos no eixo x, y e z , respetivamente

N_1, N_2, \dots, N_n – Funções de interpolação em coordenadas locais

u, v, w – Deslocamentos segundo o eixo x, y e z , respectivamente

w_i – Peso de Gauss

X, Y e Z – Forças efetivas no elemento segundo eixo x, y e z , respetivamente

Caracteres Gregos

a_1, a_2, \dots, a_n – parâmetros para a criação das funções de forma

γ_{xy} – Distorção no plano xy

γ_{xz} – Distorção no plano xz

γ_{yz} – Distorção no plano yz

ε – Vetor de deformações

ε_x – Deformação segundo eixo x

ε_y – Deformação segundo eixo y

ε_z – Deformação segundo eixo z

ξ e η – Coordenadas naturais

σ_{ced} – Tensão de cedência

σ_{max} – Tensão máxima

σ_{VM} – Tensão equivalente de von Mises

σ_x – Tensão segundo eixo x

σ_y – Tensão segundo eixo y

σ_z – Tensão segundo eixo z

τ_{xy} – Tensão de corte no plano xy

τ_{xz} – Tensão de corte no plano xz

τ_{yz} – Tensão de corte no plano yz

ν – coeficiente de Poisson

Δ – área do triângulo

Abreviaturas

CAD – Computer Aided Design

CAP – Computer Aided Planning

CNC – Computer Numerical Control

EPD – Estado Plano de Deformação

EPT – Estado Plano de Tensão

GUIDE – Graphical User Interface Design Development Environment

MEF – Método de Elementos Finitos

MRP – Material Requesting Planning

1 Introdução

1.1 Contextualização

Para o projeto de qualquer estrutura existente (edifícios, pontes, veículos, máquinas, etc.) é necessário conhecer as condições de carga, geometria e comportamento de todas as suas partes, assim como respeitar as normativas em vigor nos países nos quais a estrutura será aplicada. A primeira parte de qualquer projeto nesta área passa pela fase da análise estrutural, onde são calculadas todas as interações e efeitos de cargas sobre as estruturas físicas e os seus componentes de maneira a verificar a aptidão da estrutura para o seu uso.

Como com qualquer tipo de projeto, é importante ter à disposição ferramentas que permitam a validação e apoio ao cálculo. E, com o passar dos anos, nenhuma outra ferramenta se mostrou tão fiável e multidisciplinar quanto o Método de Elementos Finitos. Os modelos a ser analisados pelo Método de Elementos Finitos são cada vez mais complexos e, nos tempos que correm, seria impensável realizar tais análises sem um apoio computadorizado. Existe para esta finalidade uma vasta gama de programas que permitem realizar tarefas que passam pelo desenho de estruturas, análise estática de cargas, análise dinâmica e vibrações, visualização do comportamento físico (deformações) em tempo real, que permitem a otimização da estrutura. Porém, estes programas demonstram uma certa complexidade durante a introdução dos parâmetros, levando muitas vezes a resultados errados. Assim sendo, é essencial para o projetista ter uma ferramenta fiável e simples de usar que possa ser usada para fins de projeto de estruturas e otimização.

1.2 Objetivos

Sob o pretexto de permitir a qualquer utilizador uma análise de estruturas simples com o Método dos Elementos Finitos, surge esta tese onde se irá criar de raiz um programa com

interface gráfica no ambiente Matlab® para análise de estruturas simples com dois tipos de elemento finito, triangular de deformação constante e quadrangular de deformação linear. O desenvolvimento deste programa tem como objetivo os seguintes pontos:

- Apresentar deformações e tensões;
- Resolução de sistemas de geometria arbitrária;
- Apresentação dos resultados em forma gráfica e tabular;
- Inserção gráfica das condições fronteira e esforços aplicados;
- Seleção da aproximação bidimensional a efetuar (EPT ou EPD);
- Possibilidade de gravação do trabalho efetuado.

Como sub-objetivos de desenvolvimento, é requerido para o programa os seguintes pontos:

- Escolha linguística do programa;
- Exportação para Excel;
- Cálculo do fator de segurança;
- Cálculo da tensão de von Mises;
- Validação do código por comparação com *software* comercial.

1.3 Organização do relatório

A organização desta tese e efetuada da seguinte forma:

- Introdução: Neste capítulo é feita uma abordagem inicial à presente tese, fazendo uma contextualização ao problema atual com *softwares* de computação. De seguida, retratam-se os objetivos a atingir pela presente tese, assim como a organização da mesma.
- Revisão Bibliográfica: Neste capítulo é feita uma abordagem ao Método de Elementos finitos, descrevendo a sua origem e princípios gerais, assim como potencialidades e aplicações. São também descritos alguns *softwares* disponíveis para análise estrutural, assim como *softwares* de computação e visualização gráfica.
- Formulação dos elementos finitos: Neste capítulo aborda-se a Teoria básica da elasticidade com mais detalhe, iniciando o tema da elasticidade convencional para casos unidimensionais, bidimensionais e tridimensionais. Posteriormente aprofundam-se os conhecimentos sobre os elementos bidimensionais, onde consta o

elemento triangular de deformação constante de três nós e o elemento quadrangular de deformação linear de quatro nós.

- Desenvolvimento do *software*: Neste capítulo é descrito o programa desenvolvido e quais os recursos utilizados para tal, estabelecendo uma estrutura e descrevendo todas as funcionalidades do programa. Posteriormente apresentam-se os resultados obtidos pelo programa, assim como a validação do *software* desenvolvido para elementos bidimensionais triangulares e quadriláteros.
- Conclusões: Neste capítulo são apresentadas as conclusões, assim com indicadas algumas considerações que possam ser tomadas como referência futura.

2 Revisão Bibliográfica

Neste capítulo é feita uma abordagem ao Método de Elementos finitos, descrevendo a sua origem e princípios gerais, assim como potencialidades e aplicações. São também descritos alguns *softwares* disponíveis para análise estrutural, assim como *softwares* de computação e visualização gráfica.

2.1 O Método de Elementos Finitos

Em diversas áreas de Engenharia utiliza-se uma vasta gama de materiais para inúmeras construções e aplicações. Destes materiais, sejam placas ou não, destacam-se inovações tecnológicas como estruturas leves de alta resistência, próteses humanas ou componentes de aeronaves.

Para o desenvolvimento destas peças são necessários uma grande quantidade de testes preliminares com protótipos, prevendo desgaste e possíveis falhas no fabrico e fadiga do material. Tais testes exigem um enorme custo financeiro.

Por estes e outros motivos, foi necessário desenvolver um método numérico capaz de prever o comportamento de estruturas complexas. A partir da compartimentação (ou discretização) de um problema complexo em vários problemas mais simples, eis que surge o Método de Elementos Finitos (MEF). Aplicado à mecânica estrutural, este método tem como finalidade a determinação do estado de tensão e de deformação de um elemento ou estrutura sujeita a ações exteriores, para se obter uma aproximação de um modelo matemático que representa o problema físico o melhor possível. Contudo, o MEF não poderá fornecer mais informações que as fornecidas pelo modelo matemático [1].

Após esta utilização inicial em problemas de elasticidade, o MEF foi-se rapidamente estendendo a outros domínios como o da transferência de calor e da mecânica dos fluidos,

do electromagnetismo, das vibrações mecânicas e acústicas, da visão por computador, da computação gráfica, da realidade virtual (nomeadamente em simulações cirúrgicas), entre outros[2].

2.1.1 A evolução do Método de Elementos Finitos

O *Método dos Elementos Finitos* tem origem no século XVIII quando Gauss propõe o uso de funções de aproximação para a solução de problemas matemáticos[3]. Estas funções eram baseadas na expectativa do comportamento da função que se deseja determinar de forma aproximada, enquanto o processo numérico se encarregava de encontrar alguns parâmetros que ajustassem a função de aproximação à solução do problema. No entanto, existia o problema de se definir uma função apropriada, pois para problemas complexos a escolha desta aproximação requeria uma noção clara da solução do problema. Só mais tarde, no século XX se volta a desenvolver o método.

Logan [4] escreve a seguinte cronologia:

O desenvolvimento moderno do MEF começou nos anos 40 na área de engenharia estrutural com os trabalhos de Hrennikoff em 1941 e McHenry em 1943, que usaram uma grelha de elementos lineares (unidimensionais, como barras e vigas) para a solução das tensões em sólidos contínuos. Num artigo publicado em 1943 mas não reconhecido por muitos anos, Courant propôs estabelecer a solução de tensões numa forma variável. Posteriormente, como método para obter soluções numéricas aproximadas, Courant introduziu funções de forma numa subregião triangular do qual é feita a região total do sistema. Em 1947, Levy desenvolveu o método das flexibilidades, ou método das forças, e em 1953 o seu trabalho sugeria que outro método (da rigidez, ou método dos deslocamentos) poderia ser uma alternativa prometedora para analisar estruturas aeronáuticas hiperestáticas. No entanto, as equações eram laborosas de serem resolvidas à mão, e conseqüentemente o método só se tornou popular com a chegada dos computadores digitais.

Em 1954, Argyris e Kelsey desenvolveram métodos de análise estrutural matricial utilizando princípios energéticos. Este desenvolvimento ilustrou a importância que métodos de energia iriam ter no MEF.

O primeiro tratamento de elementos bidimensionais foi realizado por Turner et al, em 1956. Estes derivaram matrizes de rigidez para elementos de treliça, elementos de viga e elementos bidimensionais triangulares e retangulares em estado plano de tensão (EPT) e esboçaram o procedimento conhecido como o método da matriz de rigidez para obtenção da matriz de rigidez estrutural total. Com o desenvolvimento dos computadores digitais no início dos anos 50, o trabalho de Turner et al. incitou posteriores desenvolvimentos de equações expressas matricialmente no MEF. O termo *elemento finito* foi introduzida por Clough em 1960 quando ambos elementos triangulares e retangulares foram utilizados para a análise de problemas em EPT.

Uma matriz de rigidez de um elemento de placa retangular em flexão foi desenvolvida por Melosh em 1961. Este foi seguido pelo desenvolvimento de matriz de rigidez para elementos de casca curva para cascas assimétricas e recipientes de pressão por Grafton e Strome em 1963.

A extensão do MEF para problemas tridimensionais com o desenvolvimento duma matriz de rigidez tetraédrica foi realizada por Martin em 1961, Gallagher et al. em 1962 e por Melosh em 1963. Elementos tridimensionais adicionais foram estudados por Argyris em 1964. O caso especial de sólidos axissimétricos foi considerado por Clough, Rashid e Wilson in 1965.

A maioria do trabalho em elementos finitos até os inícios dos anos 60 lidam com pequenas deformações e deslocamentos, comportamento elástico do material e carregamentos estáticos. Contudo, a análise térmica e de grandes defleções foram consideradas por Turner et al. em 1960 e a análise dos materiais não-lineares por Gallagher et al. em 1962, enquanto problemas de encurvadura foram inicialmente abordados por Gallagher e Padlog em 1963. Zienkiewicz et al. estendeu o método a problemas de visco-elasticidade em 1968.

Em 1965, Archer considerou a análise dinâmica no desenvolvimento da matriz consistente de massa, que é aplicável à análise de sistemas de distribuição de massa, como as barras e vigas em análise estrutural.

Melosh apercebeu-se, em 1963, que o MEF poderia ser estabelecido em termos de formulação variacional, ao que começou a ser utilizado para resolver aplicações não estruturais. Problemas de campo, como a determinação da torção de um veio, fluxo de

flúido e condução térmica foram resolvidos por Zienkiewicz e Cheung em 1965, Martin em 1968 e Wilson e Nickel em 1966.

Uma posterior extensão do método foi possibilitado pela adaptação de métodos dos resíduos ponderados, primeiro por Szabo e Leeem 1969 para derivar as equações de elasticidade previamente conhecidas, utilizadas em análise estrutural, e mais tarde por Zienkiewicz e Parekh em 1970 para problemas de campo transiente. Foi então reconhecido que quando as formulações diretas e variacionais são difíceis ou impossíveis de utilizar, o método dos resíduos ponderados poderá ser apropriado. Por exemplo, em 1977, Lyness et al. aplicou o método dos resíduos ponderados na determinação de campo magnético.

Em 1976, Belytschko estudou os problemas associados com o comportamento não-linear dinâmico de sistemas com grandes deslocamentos e melhorou técnicas numéricas para resolver o sistema de equações resultante.

Com o aumento da capacidade de processamento dos computadores para resolver sistemas de equações começam a surgir os primeiros *softwares* dedicados à resolução do MEF, inicialmente com uma interface de utilizador muito limitada. Ainda assim, o MEF passa a ser mais fácil de aplicar e surge noutras áreas em que pode ser aplicado. Contudo, resolver problemas com facilidade depende do utilizador, ou, pelas palavras de Cook (1989), " [...] embora o método dos elementos finitos possa fazer um bom engenheiro melhor, ele pode fazer um engenheiro fraco mais perigoso." Não obstante, poucos dos utilizadores realmente precisarão de entender o método para extrair dele um melhor desempenho e confiabilidade.

Atualmente, o MEF restringe-se à representação de uma aproximação dum modelo matemático que representa o mais fielmente possível o problema físico[1].

2.1.2 Princípios gerais do Método de Elementos Finitos

Como referido no capítulo 2.1.1, o MEF surge na indústria aeroespacial para responder à necessidade de resolver sistemas de equações algébricas em termos de parâmetros previamente arbitrados, transformando um domínio complexo em vários sub-domínios, podendo calcular com maior facilidade e precisão o comportamento do domínio[5].

O MEF torna-se assim uma poderosa ferramenta numérica para a solução de problemas que exigem um grande esforço de cálculo e de prototipagem.

A principal desvantagem está relacionada com a necessidade de programas de computador e facilidades computacionais. Para além deste requisito, o MEF necessita de uma grande quantidade de memória para a solução de sistemas de equação de elevada dimensão. Na verdade, essa desvantagem foi motivo de preocupação durante algumas décadas. No entanto, com o avanço da tecnologia e a redução do preço dos equipamentos como a sua globalização, nomeadamente dos computadores atuais, esta desvantagem é minimizada.

O princípio fundamental do MEF é que “toda função contínua, seja ela de temperatura, pressão ou deslocamento, pode ser aproximada por um modelo composto de um conjunto de funções contínuas (dentro de um certo intervalo) definidas sobre um número finito de subdomínios” [2].

A situação mais comum é quando se desconhece o valor da função contínua e se pretende saber o seu valor em certos pontos dentro de uma determinada região. Em geral, os métodos numéricos reduzem o problema contínuo num problema discreto que pode então ser resolvido na forma de um sistema de equações algébricas.

Pretende-se assim resolver matematicamente um sistema, pressupondo que este pode ser descrito por um número finito de variáveis, tendo em conta as seguintes etapas:

- Idealização do sistema considerando a contribuição dos vários elementos constituintes;
- Estabelecimento de equilíbrio em cada elemento em termos de variáveis de interesse;
- Contabilização da contribuição de todos os elementos, tendo em conta as ligações entre elementos para as variáveis do problema;
- Cálculo da resposta, tendo em conta a solução do sistema de equações, obtendo-se as variáveis do problema. Podem obter-se outras informações para além destas variáveis para cada elemento[1].

Tradicionalmente existem duas aproximações na resolução de problemas de mecânica das estruturas, o método das forças e o método dos deslocamentos. Ambas as aproximações resultam em incógnitas diferentes na sua análise e diferentes matrizes associadas à sua

formulação. Posteriormente foram desenvolvidos modelos de equilíbrio, métodos híbridos e mistos. De todos estes métodos, aquele que apresenta uma maior simplicidade e, conseqüentemente, uma maior versatilidade é o método dos deslocamentos [6]. Este método adapta-se melhor à programação automática do que o método das forças, pois ao contrário do que acontece no método das forças, em que apenas algumas restrições são introduzidas, todos os deslocamentos são restringidos para obter a estrutura isostática[4].

2.1.2.1 Método das Forças

O método das forças foi concebido para realizar a análise de estruturas hiperestáticas recorrendo exclusivamente aos conhecimentos necessários para caracterizar o comportamento de estruturas estaticamente determinadas. Neste método explora-se a metodologia de substituir a estrutura a analisar por uma estrutura isostática equivalente denominada de estrutura-base, estudada a nível estático e cinemático, obrigando a estrutura-base a apresentar distribuições de esforços e de deformações idênticas às que se desenvolvem na estrutura hiperestática[7].

2.1.2.2 Método dos Deslocamentos

A aplicação deste método inicia-se com a definição da matriz de rigidez dos elementos, assemblando posteriormente estas matrizes numa matriz de rigidez global. De seguida, definem-se as condições fronteira, que são transformações algébricas dos carregamentos aplicados no sistema e reações do mesmo. É assim obtida uma solução completa incluindo os deslocamentos nodais e as reações do sistema. O método dos deslocamentos pode ser aplicado em ambas estruturas isostáticas e hiperestáticas. Este método será o mais adequado quando o grau de indeterminação estático é elevado em estruturas hiperestáticas, conforme demonstrado no trabalho de Logan [4].

A construção do modelo discreto do objeto contínuo, denominado por “domínio”, para aplicação do MEF pode ser visualizado na Figura 1, onde se representam os conceitos de discretização de domínio e é apresentado um problema comum de hiperestaticidade, o da treliça.

Para poder estudar este problema pelo método dos deslocamentos é necessário decompô-lo em partes mais simples. Começa-se por identificar os pontos de ligação das barras da treliça. Estes pontos são chamados pontos nodais, ou nós. Entre estes nós existe uma barra de interligação, denominada “elemento”. Estes elementos estão conectados pelos nós comuns e juntos aproximam a forma do “domínio”, que será o caso a ser estudado.

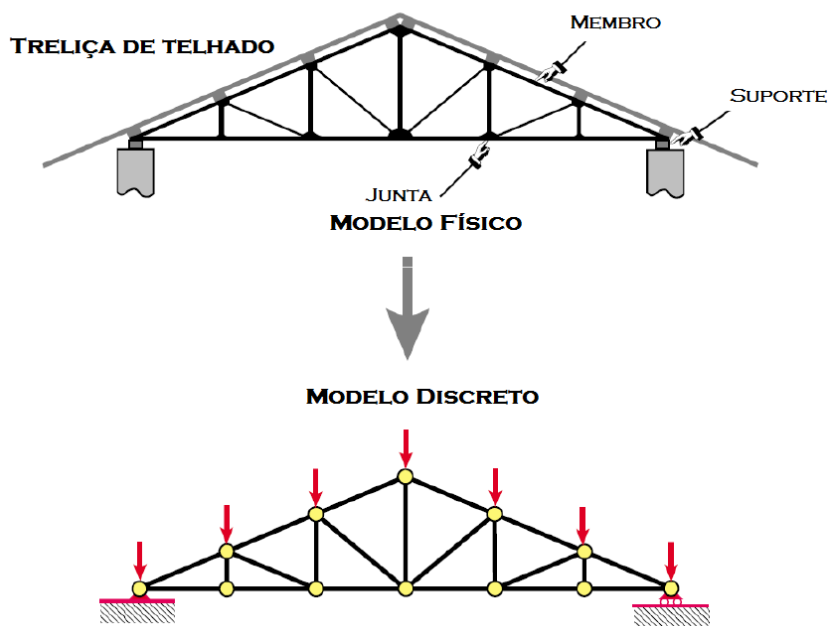


Figura 1: Problema de treliça convertido para análise utilizando o MEF.

Na aplicação do método é necessário individualizar (Figura 2) e caracterizar corretamente cada elemento segundo as suas propriedades mecânicas e geométricas, de forma a aplicar as relações tensão-deformação e deformação-deslocamento necessárias para a formação da matriz de rigidez $[k]$.

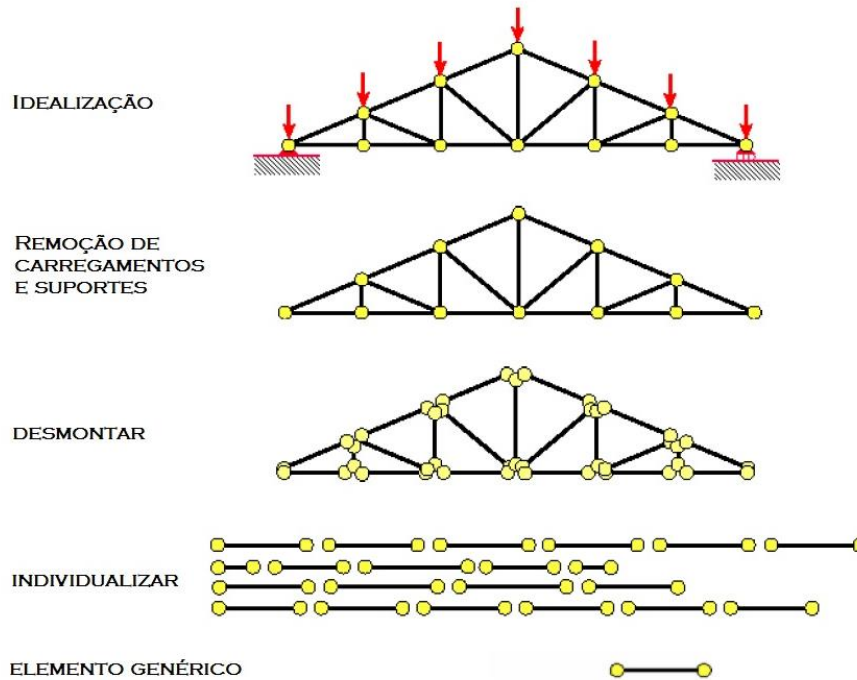


Figura 2: Decomposição do modelo em elementos.

2.1.2.2.1 Sistema de Equações Elementares

Assumindo que o material de que é constituído o modelo do MEF é homogéneo e tem um comportamento linear elástico, as forças nodais dependem de forma direta e proporcional aos deslocamentos dos nós, associados ao desenvolvimento de deformações. Estas dependências são traduzidas por equações a verificar em cada ponto e instante de uma estrutura. As equações em questão são de equilíbrio, constitutivas e de compatibilidade, conforme demonstrado na Figura 3.

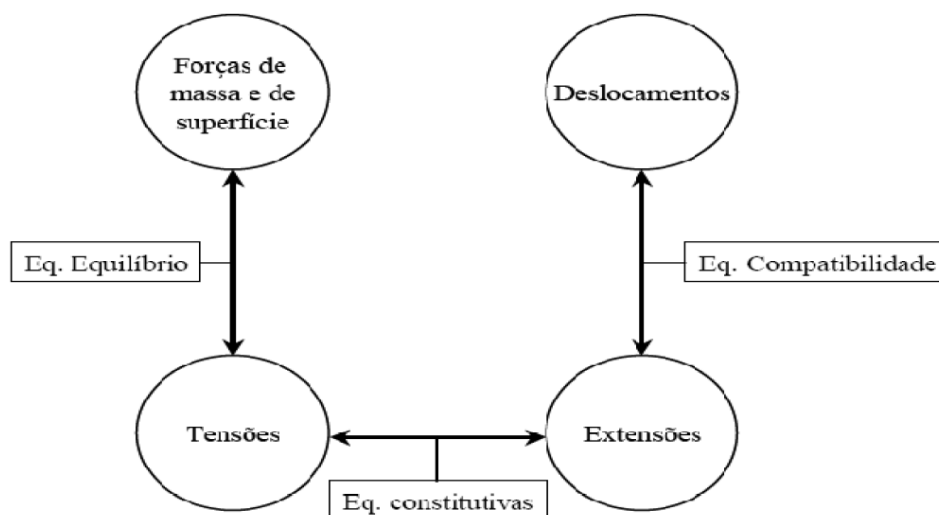


Figura 3: Variáveis e equações fundamentais da Mecânica Estrutural [8].

As relações tensão-deformação e deformação-deslocamento são necessárias para derivar as equações em cada elemento finito. No caso de deformação unidimensional, diga-se no eixo x , tem-se a relação deformação-deslocamento para pequenas deformações dada por:

$$\varepsilon_x = \frac{du}{dx}. \quad (1)$$

As tensões deverão estar relacionadas com as deformações pela lei tensão/deformação, ou lei de Hooke, conhecida também por equação constitutiva. A equação tensão/deformação, ou lei de Hooke, é dada por:

$$\sigma_x = E\varepsilon_x, \quad (2)$$

onde σ_x é a tensão na direção x e E é o módulo de elasticidade.

A relação entre o vetor de forças nodais equivalentes e o vetor de deslocamentos nodais, que se assume ser de proporcionalidade direta, pode ser expressa através da relação matricial apresentada na equação (3).

$$\begin{bmatrix} k_{11} & k_{12} & k_{13} & \cdots & k_{1n} \\ k_{21} & k_{22} & k_{23} & \cdots & k_{2n} \\ k_{31} & k_{32} & k_{33} & \cdots & k_{3n} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ k_{n1} & k_{n2} & k_{n3} & \cdots & k_{nn} \end{bmatrix} \begin{Bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_n \end{Bmatrix} = \begin{Bmatrix} f_1 \\ f_2 \\ f_3 \\ \vdots \\ f_n \end{Bmatrix} \quad (3)$$

Onde n representa o número total de graus de liberdade do elemento finito. Colocando a equação(3) numa forma mais compacta, obtem-se a equação (4).

$$\mathbf{k} \cdot \mathbf{a} = \mathbf{f} \quad (4)$$

Onde, \mathbf{k} designa a matriz de rigidez elementar, \mathbf{a} o vetor de deslocamentos e \mathbf{f} o vetor de forças nodais equivalentes.

Determinadas todas as matrizes de rigidez elementares para o problema em estudo, torna-se necessário agrupá-las por forma a construir a matriz de rigidez global do problema. Esta operação é designada por assemblagem. Uma operação semelhante tem de ser efetuada com os vetores de forças nodais equivalentes.

Findo o processo de construção da matriz de rigidez $[\mathbf{k}]$, cada elemento é assemblado numa matriz de rigidez global $[\mathbf{K}]$. Para tal, é construída uma matriz de assemblagem como demonstrada pela matriz representada em (5).

$$\begin{bmatrix} 1 & | & i \\ 2 & | & j \\ 3 & | & k \\ \vdots & | & \vdots \\ n & | & r \end{bmatrix} \quad (5)$$

Onde a primeira coluna contém a numeração local dos n graus de liberdade do elemento finito e a segunda coluna a correspondência destes com a numeração global dos graus de liberdade da malha doMEF. Esta matriz contribui para a construção do sistema de equações apresentado em (6).

$$\begin{matrix} & 1 & \dots & i & j & \dots & N \\ \begin{matrix} 1 \\ \vdots \\ i \\ j \\ \vdots \\ N \end{matrix} & \left[\begin{matrix} & & & & & & \\ & & & & & & \\ & \dots & & k_{11}^e & k_{12}^e & \dots & \\ & \dots & & k_{21}^e & k_{22}^e & \dots & \\ & & & \vdots & \vdots & & \end{matrix} \right] \begin{matrix} \left\{ \begin{matrix} \vdots \\ d_i \\ d_j \\ \vdots \end{matrix} \right\} \\ \\ \end{matrix} = \begin{matrix} \left\{ \begin{matrix} \vdots \\ f_1^e \\ f_2^e \\ \vdots \end{matrix} \right\} \\ \\ \end{matrix}
 \end{matrix} \quad (6)$$

Onde N representa o número total de graus de liberdade da malha do MEF, podendo ser escrito na forma genérica apresentada em (7).

$$\mathbf{K} \cdot \mathbf{A} = \mathbf{F}. \quad (7)$$

Findo o processo de construção matriz de rigidez global $[\mathbf{K}]$, procede-se à aplicação dos carregamentos e restrições nas matrizes \mathbf{A} e \mathbf{F} , respetivamente.

2.1.2.2.2 Restrições e Carregamentos

No MEF, todos os tipos de carregamento (cargas distribuídas, forças pontuais e momentos, forças gravitacionais, etc.) são convertidos em forças pontuais atuando nos nós [9].

O vetor \mathbf{F} é constituído pelos carregamentos que atuam no interior e/ou na fronteira de cada elemento finito da malha do MEF. Os carregamentos concentrados e distribuídos atuantes em cada elemento finito são considerados no cálculo dos vetores de forças nodais equivalentes. Os carregamentos aplicados nos nós da malha são adicionados diretamente ao vetor de forças nodais globais.

As restrições traduzem-se na anulação do deslocamento num dado nó, consoante o tipo de restrição e os graus de liberdade presentes no sistema. Isto verifica-se no vetor representado em (8).

$$\{\mathbf{A}\} = \left\{ \begin{matrix} a_1 \\ a_2 \\ \vdots \\ a_{n \times GL} \end{matrix} \right\} \quad (8)$$

Onde, a é o deslocamento, n a quantidade de nós existentes e GL o número de graus de liberdade do sistema.

Finalmente, obedecendo a (7) procede-se à resolução do sistema para os deslocamentos de cada nó, representada na Figura 4.

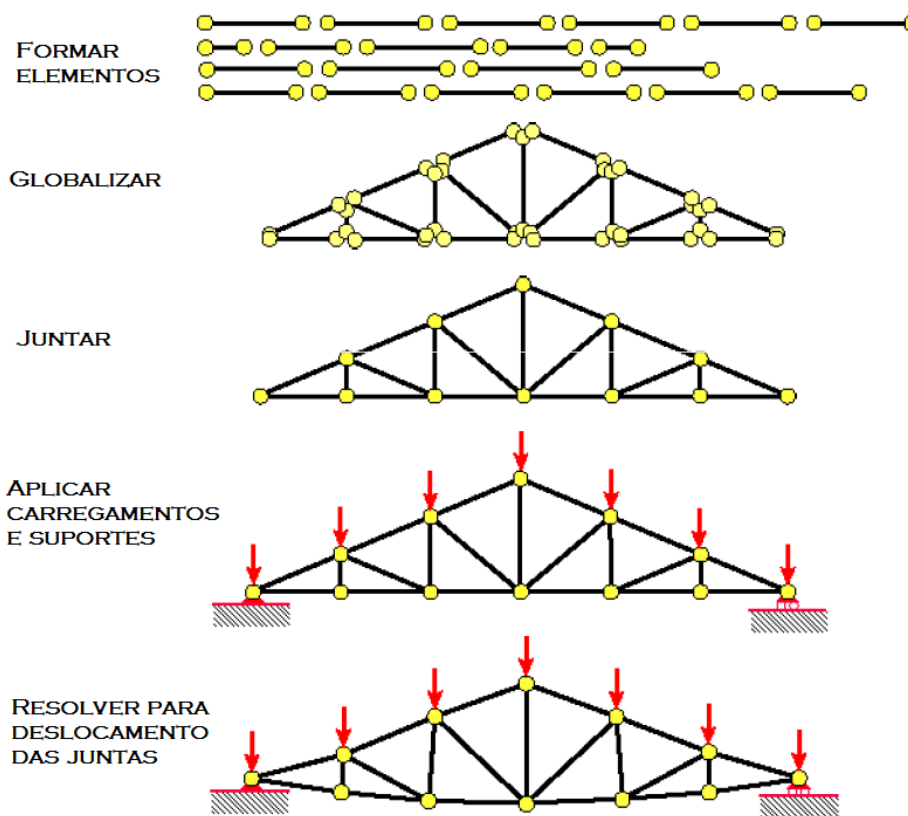


Figura 4: Montagem e resolução para os deslocamentos nodais com o MEF[10].

Após resolução do sistema de equações em função dos deslocamentos, conhecendo agora integralmente o vetor \mathbf{A} , obtem-se o valor das reações reaplicando (7). Por fim, calculam-se as tensões e deformações para cada elemento através das equações (1) e (2).

Para estruturar este capítulo, divide-se o procedimento descrito por passos:

- Dividir a estrutura em elementos discretos (discretização);
- Selecionar um modelo de deslocamento adequado ou uma interpolação propícia;
- Derivar as matrizes de rigidez do elemento e vetores de carregamento;
- Assemblar matrizes de rigidez para obter equações de equilíbrio gerais;

- Resolver o sistema de equações em função dos deslocamentos desconhecidos;
- Obter as reações resolvendo o sistema de equações em função dos deslocamentos;
- Computar tensões e deformações por elemento.

2.1.2.2.3 Tipos de elementos finitos

A escolha do tipo de elemento utilizado numa análise de MEF depende da estrutura física do corpo sob condições de carregamento reais e da aproximação que se pretende obter do comportamento real. É necessária a ponderação apropriada na utilização de elementos uni, bi e tridimensionais. Os elementos mais utilizados podem ser visualizados no Anexo A.

Elementos Unidimensionais

Os elementos unidimensionais consistem em elementos de barra ou de viga. Estes elementos têm uma área transversal, mas são usualmente representados por elementos lineares. Estes elementos são usados regularmente para modelar treliças e estruturas reticuladas contínuas. O elemento de linha mais simples, denominado elemento linear, é constituído por dois nós, um em cada extremidade, e exibe um comportamento linear (Figura 5-a). Se possuir mais pontos nodais pode ser um elemento quadrático, cúbico, entre outros dependendo do número de nós. Este torna-se num elemento de ordem elevada podendo exibir curvatura (Figura 5-b).

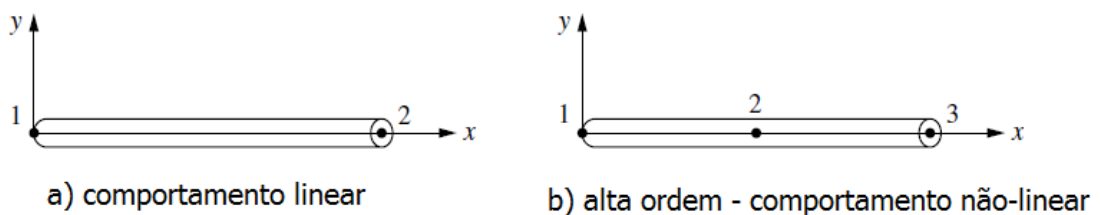


Figura 5: Elementos unidimensionais [4].

Elementos bidimensionais

Nos elementos bidimensionais os carregamentos são aplicados no próprio plano, em EPT ou em estado plano de deformação (EPD). Os elementos bidimensionais mais comuns são os elementos triangulares e quadriláteros, possuindo três e quatro pontos nodais respectivamente (Figura 6-a). No caso de EPD, a componente de deformação e as distorções associadas à direção perpendicular a esse plano são consideradas nulas. Também é possível o uso de elementos com nós adicionais ao longo de suas arestas, denominados por elementos de ordem elevada, mais complexos e que permitem representações polinomiais quadráticas (Figura 6-b) [4].

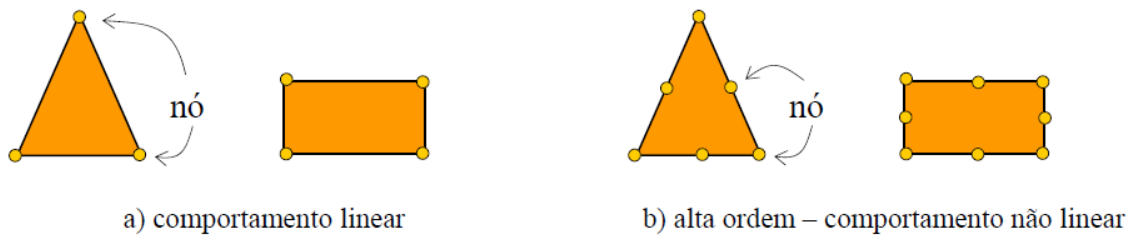


Figura 6: Elementos bi-dimensionais [11].

Elementos axissimétricos

O elemento axissimétrico é construído revolucionando um triângulo ou um quadrilátero à volta de um eixo fixo localizado no plano do elemento a 360° (Figura 7). Este elemento pode ser utilizado quando as condições fronteira e os carregamentos do problema são axissimétricos [4].

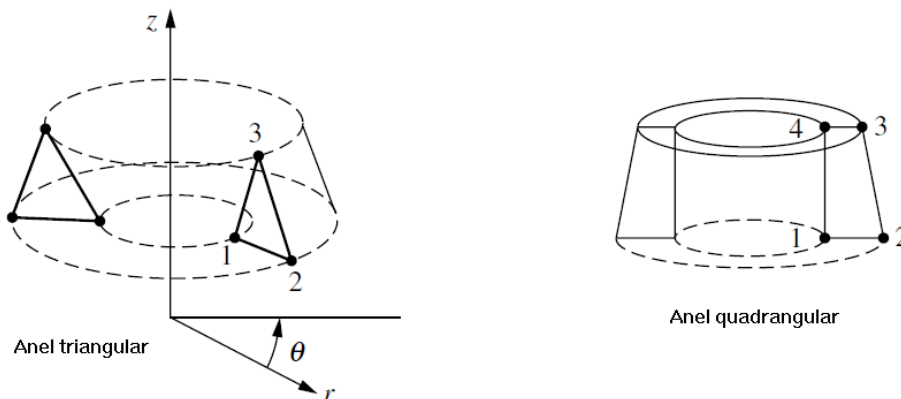


Figura 7: Elementos axissimétricos triangulares e quadrangulares

Elementos de placa e casca

Os elementos de placa e casca modelam corpos de espessura fina. Os elementos de placa destinam-se a modelar placas planas com carregamentos fora do plano, resultando na flexão e respetiva rotação segundo 2 eixos, e também na deflexão transversa. As cascas são definidas pela sua espessura e superfície média, representada por uma superfície curva no espaço. As cascas suportam principalmente esforços de membrana (no plano), mas também resistem a esforços transversos de flexão como os elementos de placa. Em termos de geometria, podem exibir curvatura em um ou dois eixos, adquirir forma semelhante aos elementos bidimensionais simples ou de ordem elevada, representada na Figura 6, tendo como principal diferença o número de graus de liberdade por nó [12].

Elementos tridimensionais

Na prática, a generalidade dos problemas de engenharia encontrados são tridimensionais. Recorre-se a simplificações que permitem analisar o meio tridimensional através de uma abordagem bidimensional. Contudo, em muitos casos não é possível ou desejável adotar as simplificações do EPT ou outras.

A utilização de elementos finitos tridimensionais proporciona uma maior abrangência de possibilidades na modelação de meios contínuos que não são fornecidas pelos elementos finitos bidimensionais. Os elementos tridimensionais mais utilizados são os elementos tetraédricos e hexaédricos, constituídos apenas pelos nós dos vértices e arestas lineares. Os elementos de ordem elevada com nós a meio das arestas possuem superfícies curvas (Figura 8) [4].

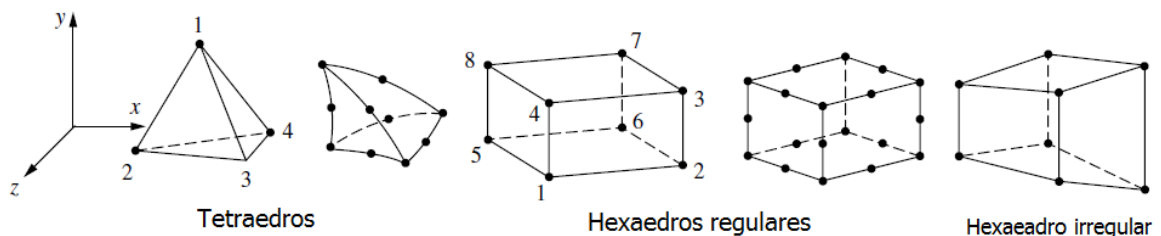


Figura 8: Elementos tri-dimensionais e quantidades de nós [4].

2.1.3 Potencialidades

O MEF consiste num método numérico para solução de um sistema de equações matemáticas que descrevem fenómenos físicos em engenharia. As suas potencialidades incluem, entre outros, os seguintes pontos:

- Representação bastante aproximada de geometrias complexas;
- Análise de estruturas de grau hiperestático elevado;
- Análise de problemas de engenharia estáticos e dinâmicos;
- Inclusão de materiais com propriedades dissimilares;
- Representação do resultado final relativamente simples;
- Captura de efeitos localizados como concentração de tensão em zonas de entalhe;
- Pode ser utilizado para obter o valor numérico de problemas de equação diferenciais;
- Simulação de processos de fabricação de produto, utilizando campos da engenharia diversos;
- A utilização de um sistema CAE (*Computer Aided Engineering* ou Engenharia Auxiliada por Computador) pode reduzir significativamente a quantidade de recursos em prototipagem de teste, pois reduz a quantidade de soluções inviáveis a partir dos resultados obtidos e, conseqüentemente, o tempo e custo de concepção de produto[1] [13] [14].

Assim, existem muitas razões para se instalar um sistema computacional no auxílio de projecto:

- **Aumentar a capacidade do projectista/engenheiro:** Conseguido por permitir ao projetista visualizar o produto final, reduzindo o tempo necessário em sintetizar, analisar e documentar o projecto. O aumento de produtividade traduz-se não só em custos mais baixos de projecto, mas também em prazos menores para a sua conclusão;

- **Melhorar a qualidade do projecto:** Os sistemas CAD (*Computer Aided Design*) com módulo de MEF permitem uma análise mais completa (da concepção ao dimensionamento final do produto) e propiciam um maior número de alternativas para investigação em pouco tempo, originando assim menores erros dimensionais de projecto;

- **Criar bases de dados:** No processo de criação de um produto desenvolvido em CAD, é automaticamente gerada uma base de dados com informações geométricas que posteriormente poderão ser utilizadas para produção através de CNC (*Computer Numerical Control*). Ao preencher a documentação necessária do projeto do produto (especificação de materiais, lista de componentes, dimensões do produto, notas de desenho, numeração das peças, etc.) é também gerada uma base de dados utilizada por actividades de suporte em produção tais como: CAP (*Computer Aided Planning*), MRP (*Material Requesting Planning*) etc. [15].

2.1.4 Aplicações do Método de Elementos Finitos

O MEF é um método numérico aplicado não só à resolução das equações diferenciais da Teoria da Elasticidade, mas também à resolução dos mais diversos problemas existentes em todas as áreas da Engenharia.

Nos problemas da Teoria da Elasticidade as variáveis físicas são os deslocamentos, as extensões e tensões, relacionadas pelas equações de equilíbrio, compatibilidade e constitutivas. Na Figura 9 pode-se verificar o resultado de uma análise estrutural, neste caso um painel de aço, cuja paleta de cores à direita da figura indica a intensidade da deformação causada pela ação do vento.

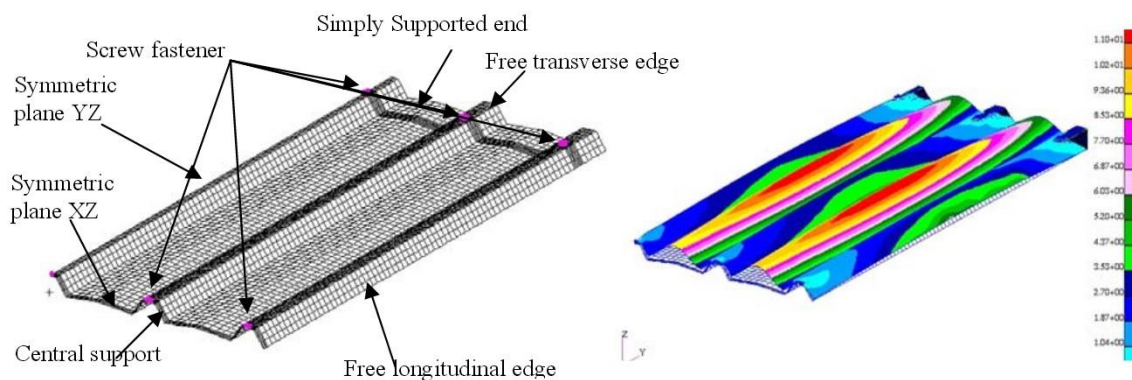


Figura 9: Painel de aço sob ação do vento analisado pelo MEF. [16]

Nos problemas de condução de calor, onde a variável utilizada é a temperatura, o MEF permite estudar a distribuição da temperatura no interior de uma estrutura para, por exemplo, definir a acção térmica a considerar no estudo de um estado de tensão.

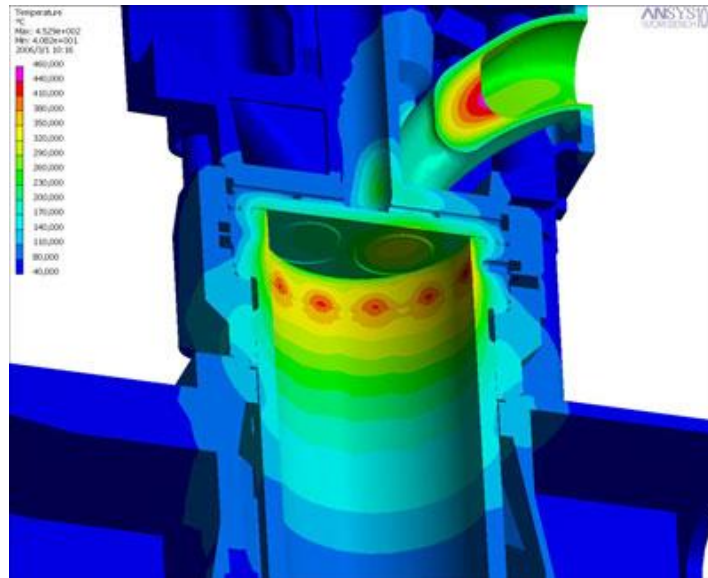


Figura 10: Simulação com recurso ao MEF para calcular a carga térmica de uma câmara de combustão. [17]

O interesse dos estudos na área continuam com a análise de fenómenos não apenas estruturais. Referem-se como exemplo a análise termoestrutural, onde o cálculo de tensões térmicas é integrado com o cálculo do transiente de temperatura (Figura 10); a interação fluido-estrutura (Figura 11), a análise de hidroelasticidade e aeroelasticidade, e problemas multicampo em geral, tais como acústica da aeroelasticidade.

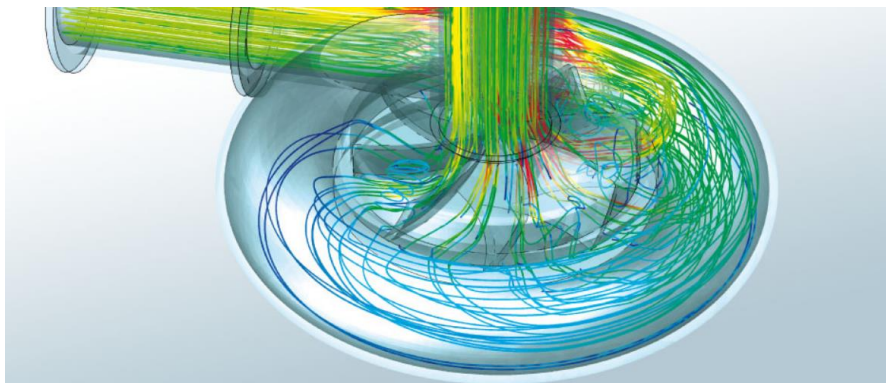


Figura 11: Comportamento de uma câmara de pressurização em análise de MEF. [18]

É ainda possível através do MEF formular o comportamento de campos electromagnéticos a partir das equações de Maxwell, como se pode verificar entre as duas bobinas de indução na Figura 12.

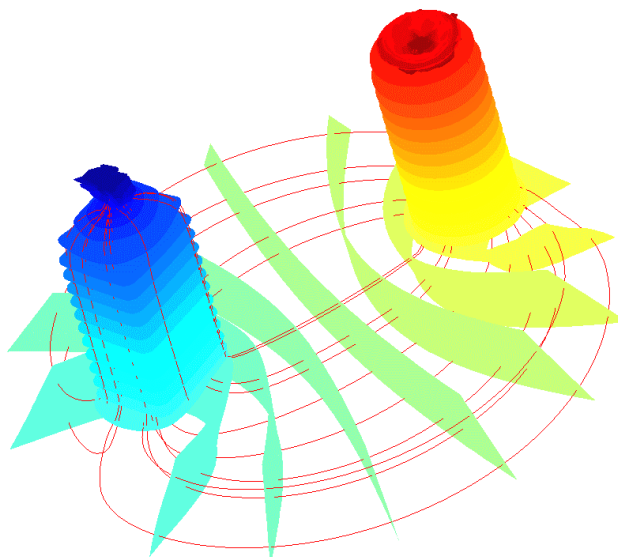


Figura 12: Campos electromagnéticos entre bobinas de indução com o MEF[19].

Nos problemas de escoamento em meios porosos, a variável utilizada é o potencial hidráulico de um determinado solo e as condições fronteira correspondem à imposição de pressões hidráulicas ou caudais (Figura 13) [8].

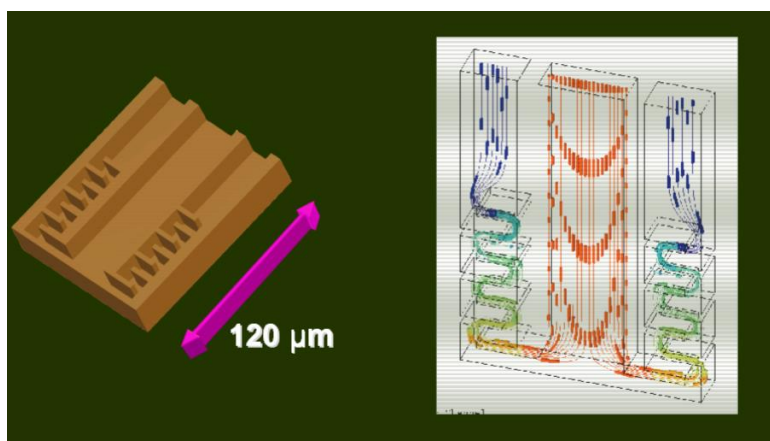


Figura 13: Canal micro-fluídico com escoamento não-newtoniano. [20]

2.2 Softwares disponíveis para análise estrutural

Como referido anteriormente, a utilização do MEF tem como grande desvantagem o elevado número de sistemas de equações algébricas geradas para resolver cada problema. Isto é em parte ultrapassado com a evolução dos computadores e a sua capacidade de processamento, permitindo resolver estes sistemas que seriam morosos e impraticáveis numa fração de tempo.

Posto isto, a solução para qualquer análise de problemas utilizando o MEF em *softwares* comerciais abrange os seguintes passos [14]:

Pré processamento:

Neste passo, a geometria, propriedades do material, carregamentos e condições fronteira são fornecidos pelo utilizador. Módulos de geração de malha integrados criam malhas de elementos finitos com um *input* mínimo do utilizador no tipo de elemento e densidade da malha a ser utilizada. O utilizador pode visualizar os dados inseridos assim como a malha de elementos finitos, gerada programaticamente para inspeção visual e verificação de validade.

Análise numérica:

O *software* gera automaticamente as matrizes características do elemento (matriz de rigidez) e vetores característicos (vetor de carregamentos), faz a sua assemblagem para gerar o sistema de equações, implementa as condições fronteira especificadas e resolve as equações para encontrar os valores nodais da variável de campo (vetor de deslocamentos) e reações desconhecidas (vetor de carregamentos). Por fim, o *software* calcula as resultantes para os elementos (componentes de tensão e deformação).

Pós processamento:

A solução para o problema, como deslocamentos nodais e tensões no elemento, pode ser mostrada em formato numérico ou gráfico (com gráficos bi ou tridimensionais da variação de tensão ou deformada). O utilizador pode escolher o modo de visualização dos resultados.

Completados estes passos, é de extrema importância verificar os resultados obtidos por um *software* de análise MEF. Normalmente isto faz-se testando um problema simples com solução teórica conhecida.

Atualmente existe uma grande escolha de programas para cálculo e simulação, com licenças que variam entre o *open-source* (código aberto e grátis) até licenças que estão fora do alcance do utilizador pontual. A acessibilidade difere em características tais como:

- Capacidades de desenho assistido por computador (CAD);
- Possibilidade de simulação;
- Comparação de resultados;
- Possibilidade de trabalho multiplataforma (trabalho em rede);
- Disponibilidade de mais de um código para cálculo;
- Facilidade de leitura e criação de vários tipos de ficheiros (compatibilidade);
- Tipo de interface com o utilizador (linha de texto ou interface gráfica);
- Certificação sobre os testes efetuados.

2.2.1 Softwares comerciais de uso genérico

No capítulo anterior refere-se a variedade de programas para cálculo e simulação utilizando o MEF. Referem-se de seguida alguns destes *softwares*:

Nastran

O programa de análise por elementos finitos **NASTRAN** foi desenvolvido nos Estados Unidos pela NASA no final dos anos 60. Escrito primariamente em **FORTRAN** e contendo acima de um milhão de linhas de código *open source*, é hoje em dia utilizado nas indústrias aeroespacial, automotiva e marítima[21]. A sua componente comercial, **MSC Nastran**[®], é uma aplicação para análise estrutural multidisciplinar utilizada por engenheiros para análises estáticas, dinâmicas e térmicas em domínios lineares e não lineares, complementada por uma optimização estrutural automatizada e por tecnologias de análise de fadiga embutidas. Utiliza-se este programa para certificação de sistemas estruturais em termos de resiliência, resistência e vida útil antes de falha, necessários para não comprometer funções estruturais e a segurança[22].

ANSYS®

O ANSYS® é um *software* de CAE desenvolvido no início da década de 1970 nos Estados Unidos. Reconhecido internacionalmente, este *software* que inicialmente enverudou pela análise estrutural e rapidamente se expandiu para outros tipos de análise[23] [24]. Este *software* que permite:

- Possibilidade de trabalho por diagrama de fluxo, permitindo aos resultados de um domínio em um dado modelo, servir como dados de entrada num outro domínio de um outro modelo (efeito dominó);
- Possibilidade de manipulação dos modelos de estudo através de *Drag-and-Drop* de condições de estudo (propriedades físicas externas/vizinhança);
- Possibilidade de baterias de estudos sequenciais a um mesmo modelo;
- Possibilidade de mudança das condições de análise ainda antes de ser concluída a simulação ou análise;
- Possibilidade de importação de modelos CAD de diferentes fontes;
- Cálculo e modelação paramétrica;
- Possibilidade de ajustar a malha de elementos dos modelos em análise segundo os critérios pretendidos (refinamento à medida).

ABAQUS®

Originalmente lançado em 1978, o *software* ABAQUS® utiliza a linguagem de programação *Python open-source* para edição e estrutura. Atualmente comercializado sob a marca SIMULIA®, marca da Dassault Systèmes S.A., este *software* inclui:

-**ABAQUS/Standard**, um programa de MEF para propósitos gerais;

-**ABAQUS/Explicit**, um programa de MEF de dinâmica explícita para resolução de sistemas altamente não lineares com muitos contactos complexos sob carregamentos variáveis;

-**ABAQUS/CAE**, uma interface interativa para construir, analisar e visualizar resultados;

-**ABAQUS/CFD**, uma extensão do módulo **ABAQUS/CAE** para análise de dinâmica de fluidos.

Ambos os módulos **ABAQUS/Standard** e **ABAQUS/Explicit** permitem um grande controlo e detalhe do procedimento de malhagem. [25]

Existem outros módulos para análise pelo MEF como método das partículas, análise eletromagnética, transferência de calor, modelos de rotura, entre outros [26].

Podem ser utilizados dois métodos para construir os modelos – a partir da interface **ABAQUS/CAE** e escrevendo um ficheiro de código. Os ficheiros de código podem ser escritos em qualquer editor de texto. Estes contêm informação incluindo o tipo de problema, geometria da estrutura, propriedades da estrutura, parâmetros de análise e dados de saída requeridos.

O tipo de ficheiro gerado pelo *software* **ABAQUS/CAE** é compatível com **Solidworks®**, **Pro/ENGINEER®** e **CATIA®**, podendo ser acedidos por estes *softwares* sem necessidade de conversão [27].

ADINA®

O desenvolvimento do **ADINA®** começa em 1974 nos Estados Unidos. Posteriormente, em 1986 é fundada a empresa **ADINA R & D, Inc.**, que o desenvolve até aos dias de hoje. Este *software* comercial é utilizado pela indústria e para fins académicos para resolver problemas estruturais, de fluidos, transmissão de calor e electromagnéticos. Pode também ser utilizado para resolver problemas multifísicos, incluindo interações estrutura-fluido e problemas termo-mecânicos [28].

As principais características da Interface Gráfica **ADINA®** incluem:

- A geometria do modelo pode ser criada diretamente a partir da interface gráfica, importada de vários sistemas CAD que suportem Parasolid (como por exemplo **NX®**, **Solid Edge®** e **SolidWorks®**), ou importada através da interface IGES;
- As propriedades físicas, cargas e condições de contorno podem ser atribuídas diretamente sobre a geometria do modelo. Malhas de MEF podem ser assim modificadas sem afetar a definição do modelo;
- A geração de malha totalmente automática pode ser utilizada em geometria geral com controlo flexível sobre a distribuição do tamanho de elemento. Pode ser utilizado malhagem mapeada em geometrias mais simples;

- Suporta malhagem de elementos unidimensionais, bidimensionais e tridimensionais, onde as capacidades em 3D incluem malhagem mapeada com elementos tetraédricos, de bloco e de cunha, malhagem livre que gera elementos tetraédricos, ou uma mistura de ambos processos;
- O ficheiro de sessão mantém a informação dos dados introduzidos e seleção durante a criação do modelo. O modelo pode ser totalmente recreado a partir do ficheiro de sessão. Pode também ser criado um modelo diferente modificando o ficheiro de sessão [29].

Autodesk Inventor®

Inicialmente lançado em 1999, este *software* é um rival direto dos *softwares* comerciais **SolidWorks®** e **Solid Edge®**. Utilizando o seu próprio código de modelação geométrica, *ShapeManager*, e cada vez com maior utilização na indústria, as características deste programa são:

- Modelação em CAD;
- Dispõe de biblioteca com características e propriedades de materiais;
- Programação dos modelos de modo a agirem como os seus modelos reais;
- Disponibilidade de um grande número de elementos finitos para simulação (barras, placas, tetraedros, axisimétricos, etc.);
- Possibilidade de importação de modelos CAD de diferentes fontes;
- Vocacionado para modelos de grande porte;
- Cálculo de modelos estáticos, dinâmicos e transientes, lineares e não lineares;
- Ferramentas de otimização do modelo durante a simulação;
- Cálculo de resposta acústica e fadiga.

CATIA®

O **CATIA®** é o *software* CAD produzido pela Dassault, que pertence à IBM. Este *software* de CAD é um híbrido, uma vez que permite a modelação de superfícies, assim como de sólidos.

A implementação do **Catia®** tem aumentado bastante, graças a dois factores:

- A indústria aeronáutica e a indústria automóvel têm uma tendência natural de utilizar este *software*, uma vez que as superfícies são superfícies NURBS, permitindo assim descrever as grandes superfícies com elevada precisão;

- A popularidade deste *software* CAD deve-se ao facto de utilizar formatos de ficheiros de difícil leitura em outros programas de CAD. Daí resulta que muitos fornecedores se vêm obrigados a adquirir este programa, nem que seja apenas para conseguir ler os ficheiros dos seus clientes. CAD [15].

Os controlos para análise em elementos finitos incluem uma malhagem altamente customizável, com ferramentas interativas de edição de malha, opção de automação completa e suavização de malha[30].

2.2.2 Códigos específicos

Existe uma grande variedade de *softwares* comerciais disponíveis para análise pelo MEF, como se demonstrou nos exemplos do capítulo anterior. Contudo, nem todo o *software* disponível para este tipo de conteúdo é de vertente comercial. Existem várias bases de dados com programas e códigos que podem ser implementados em diversas plataformas programáticas para este efeito. Como exemplo, o Matlab[®] atinge um nível superior face aos seus congéneres devido à sua extensa base de dados criada pelos utilizadores. Esta característica foi conseguida devido às capacidades de programação modular, cálculo matricial e possibilidade de integração gráfica deste programa.

MESH2D [31]

Criado em 2009 por Darren Engwirda, MESH2D é uma caixa de ferramentas de rotinas de geração de malha 2D que permite a geração automática de malhas triangulares não estruturadas para a geometria 2D em geral. É usado um algoritmo iterativo para otimizar a posição da topologia/vértices da malha para obter triangulações de alta qualidade. Para além das definições totalmente automáticas, o MESH2D permite ao utilizador especificar a informação de dimensionamento de malha, o que permite diferentes níveis de resolução de malha dentro do domínio.

DistMesh2d[32]

Criado por Per-Olof Persson e Gilbert Strang, este é um código elegante com poucas linhas que gera uma malha não estruturada a partir de funções de distância, que definem a geometria a analisar. Usa uma técnica iterativa baseada na analogia de malhas simples e uma estrutura treliçada de molas. Após equilibrar a posição de cada nó através da relação força-deslocamento, recorre à triangulação de Delaunay para reestruturar a malha. Este código gera uma malha de grande qualidade, tendo como desvantagem de ser somente possível definir corpos para malhagem que obedeçam a funções de distância, como círculos, quadrados, segmentos de raio, polígonos e junções dos anteriores.

PolyMesher[33]

Criado por Talischi et al. em 2010, apresenta-se um código robusto de geração de malha com elementos poligonais inspirado no trabalho de Per-Olof Persson e Gilbert Strang, *DistMesh2d*. O *PolyMesher* tem a possibilidade de resolver sistemas pelo MEF utilizando polígonos convexos. Utilizando funções de distância relativos à geometria a analisar e diagramas de Voronoi, este código possibilita a malhagem de geometrias mais complexas sem se deparar com os problemas de instabilidade numérica de malhas de elementos triangulares e quadrangulares.

FEPSA [34]

Desenvolvido por Sandro Campos em 2014, este programa permite definir a estrutura por meio de coordenadas, introdução de forma rápida e clara, propriedades mecânicas dos elementos, condições fronteira e cargas a aplicar. Como resultados devolve ao utilizador as reações, deformações e distribuição de tensões nos elementos quer em forma tabular quer em representação gráfica sobre a estrutura em análise. Existe ainda a possibilidade de importação de dados e exportação dos resultados em ficheiros XLS e XLSX, de maneira a facilitar a gestão de informação.

2.3 *Softwares* de computação e visualização gráfica

Em geral existe um conjunto de características procuradas em programas de computação e visualização gráfica. Do grande leque de *softwares* que existem disponíveis que permitem a criação de códigos para realizar as tarefas de cálculo durante as análises por MEF. Dependendo das pretensões do utilizador, a escolha recai sobre um conjunto das seguintes características:

- Capacidade de cálculo vetorial, matricial, polinomial, etc;
- Rapidez de cálculo;
- Possibilidade de importação exportação de diferentes tipos de ficheiros;
- Capacidade de criação de gráficos (*plots*) de diferentes tipos;
- Possibilidade de interação com outros programas;
- Ser capaz de funcionar em diferentes máquinas e sistemas operativos;
- Possibilidade de criação de interfaces gráficas para os programas a ser criados;
- Tipo de aplicação (aplicação para problemas gerais ou particulares).

2.3.1 Características gerais dos *softwares* existentes no mercado

As principais desvantagens dos *softwares* disponíveis para análise comercial são as seguintes:

- O custo inicial de desenvolvimento do *software* é elevado;
- Os *softwares* comerciais são geralmente para um âmbito elevado de problemas, o que os torna menos eficientes que programas feitos para casos específicos;
- Muitos dos *softwares* são proprietários, o que limita o acesso do utilizador ao programa. Se for necessária uma revisão, é frequente ter de ser o programador a fazer a mesma [4].

Tendo estes fatores em conta e as características dos *softwares* de computação e visualização gráfica anteriormente referidos, de seguida serão enumerados alguns dos *softwares* existentes no mercado que permitem a criação de códigos e programas específicos pelo utilizador, sendo apresentada uma breve descrição sobre os mesmos de seguida.

Matlab®[35]

Programa utilizado mundialmente como ferramenta de cálculo, tratamento de dados e de imagens, com uma enorme comunidade de utilizadores. Disponível na vertente comercial. Linguagem programática semelhante à do Fortran. Algumas das características do programa são:

- Grande número de fontes, tutoriais e guias;
- Referência no que toca a capacidades programáticas e de cálculo;
- Visualização de gráficos em 2D e 3D;
- Grande número de funções matemáticas disponíveis na biblioteca;
- Possibilidade de importação de ficheiros de diferentes tipos;
- Vocacionado para a computação matemática;
- Atualmente existem módulos como o ToolBox NuPAD, que permitem cálculo simbólico;
- Capacidade de criação de janelas programáticas para interface gráfica a partir de código ou a partir da ferramenta interna de criação de interface gráfica, GUIDE (*Graphical User Interface Development Environment*);
- Simulação gráfica de multi-domínios;
- Capacidade de cálculo vetorial, matricial, polinomial, paramétrica, etc.;
- Ferramentas estatísticas de suporte;
- Possibilidade de conversão para linguagem programática C.

FlexPro®[36]

Software criado pela Weisang GmbH, disponível na vertente comercial. As características desta plataforma programática são:

- Linguagem de programação FPScript;
- Possibilidade de criar e visualizar *plots* e GUIs;
- Possui assistentes para cada área de trabalho;
- Guarda todas as variáveis sob um mesmo ficheiro (não dá para modificar);
- O ponto anterior permite a criação de relatórios *multi-data*;
- Cálculo matricial;

- Permite a importação e exportação de vários tipos de ficheiros;
- Permite visualização de vídeos.

Sage [37]

Criado a partir de mais de 100 *softwares open-source*, não pretende copiar o Matlab[®], mas melhorou a linguagem programática. Algumas características desta plataforma são:

- *Shell* interativa e permite manipulação;
- Cálculos básicos de Álgebra;
- *Plot* 2D e 3D;
- Possibilidade de criação de GUIs;
- Cálculo de grupos finitos (não confundir com elementos finitos);
- Permite a instalação de módulos programáticos.

2.3.2 Seleção do software a utilizar no desenvolvimento deste trabalho

O *software* seleccionado para o desenvolvimento deste trabalho é o Matlab[®]. Sendo a plataforma programática requirida para a execução desta tese, o Matlab[®] dispõe de ferramentas de apoio à criação de janelas programáticas para interface com o utilizador, uma grande comunidade existente em fóruns, a acessibilidade da linguagem programática e a capacidade gráfica para o desenho das estruturas em análise.

2.3.3 Funcionalidades pretendidas para o software a desenvolver

Uma vez seleccionado o *software* Matlab[®] como plataforma base na programação e tendo em conta as características exigidas para uma análise por Método de Elementos Finitos (MEF), as características no software a desenvolver serão as seguintes:

- Capacidade de cálculo por elementos finitos;
- Interface gráfica para interação com o utilizador;
- Pré-visualização da estrutura em análise;

- Introdução e visualização da posição das cargas e das reações na pré-visualização da estrutura em análise;
- Capacidade de análise com recurso a elementos triangulares e quadrangulares;
- Possibilidade de introduzir propriedades mecânicas e elásticas para a estrutura;
- Obter gráfico da distribuição de tensões na estrutura em análise por elementos;
- Apresentar os resultados em tabela;
- Capacidade de alterar as características do material e da malha após obtenção de resultados sem necessidade de reiniciar a análise;
- Exportar os resultados, criando um relatório da análise num ficheiro em Excel;
- Possibilidade de gravar o estado da análise e continuar noutra sessão.

3 Formulação dos elementos finitos a implementar

Neste capítulo aborda-se a Teoria básica da elasticidade com mais detalhe, iniciando o tema da elasticidade convencional para casos unidimensionais, bidimensionais e tridimensionais. Posteriormente aprofundam-se os conhecimentos sobre os elementos bidimensionais, onde consta o elemento triangular de deformação constante de três nós e o elemento quadrangular de deformação linear de quatro nós.

3.1 Teoria básica da elasticidade

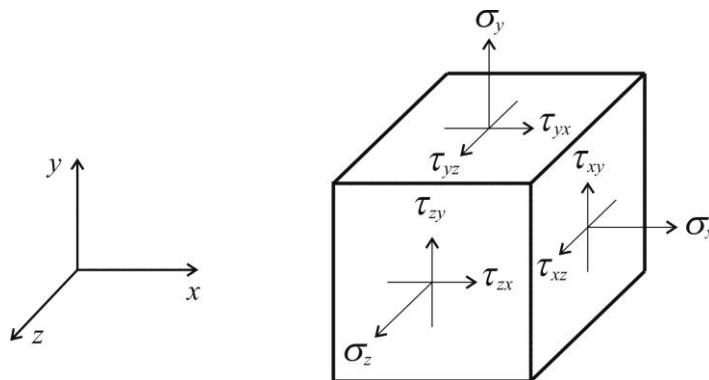
A *Teoria da Elasticidade* lida com a análise linear de tensões e deformações, isto é, considerando uma relação linear entre os esforços aplicados e os deslocamentos na estrutura. Tanto as análises linear como não-linear têm sido alvo de estudo nos últimos 3 séculos, começando por *Hooke*, razão pela qual esta Teoria também é conhecida por *Lei de Hooke* [38]. A *Teoria da Elasticidade* como é apresentada nesta Secção é usada na maioria das análises estruturais, uma vez que sob solicitações normais de serviço as estruturas não sofrem plasticidade nem grandes deslocamentos. Esta Teoria também permite lidar com muitos tipos de fenómenos relevantes à ciência dos materiais, como a determinação de deformações e tensões na proximidade de fendas. No entanto, nesta Secção serão apenas desenvolvidos os tópicos necessários à formulação dos elementos finitos planos ou sólidos, axisimétricos ou de placa, que são abordados nas próximas Secções, focando nas relações tensão-deformação e deformação-deslocamento (necessárias para a obtenção de \mathbf{k}_e).

3.1.1 Elasticidade convencional

O estado de tensão σ e deformação ϵ num qualquer ponto de uma estrutura tridimensional pode ser definido pelas seguintes componentes [39]

$$\sigma = \{\sigma_x \quad \sigma_y \quad \sigma_z \quad \tau_{xy} \quad \tau_{yz} \quad \tau_{zx}\}^T \quad (9)$$

$$\epsilon = \{\epsilon_x \quad \epsilon_y \quad \epsilon_z \quad \gamma_{xy} \quad \gamma_{yz} \quad \gamma_{zx}\}^T. \quad (10)$$



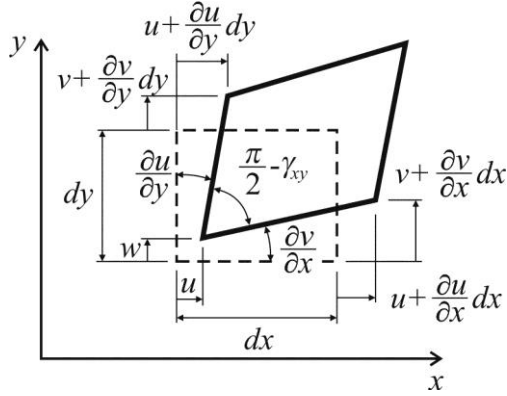
Sob determinadas condições, o estado de tensão e deformação num corpo pode ser simplificado. Como tal, uma estrutura tridimensional pode, então, ser modelada por uma análise bidimensional. Distinguem-se os casos de EPTe EPD. O caso mais simples é o unidimensional.

3.1.1.1 Relações tensão-deformação

Para materiais com comportamento linear elástico, as relações entre as tensões e deformações podem ser definidas a partir da *Lei de Hooke*[40]. Para materiais isotrópicos, as duas propriedades a considerar são o módulo de Young (E) e o coeficiente de *Poisson* (ν).

3.1.1.1.1 Caso tridimensional

Considerando um cubo elementar de um sólido, a *Lei de Hooke* permite escrever



$$\varepsilon_x = \frac{\sigma_x}{E} - \nu \frac{\sigma_y}{E} - \nu \frac{\sigma_z}{E}, \quad \varepsilon_y = \frac{\sigma_y}{E} - \nu \frac{\sigma_x}{E} - \nu \frac{\sigma_z}{E}, \quad \varepsilon_z = \frac{\sigma_z}{E} - \nu \frac{\sigma_x}{E} - \nu \frac{\sigma_y}{E} \quad (11)$$

$$\gamma_{yz} = \frac{\tau_{yz}}{G}, \quad \gamma_{xz} = \frac{\tau_{xz}}{G}, \quad \gamma_{xy} = \frac{\tau_{xy}}{G}.$$

O módulo de elasticidade transversal, G , é dado por

$$G = \frac{E}{2(1+\nu)}. \quad (12)$$

A partir de (11) é possível escrever

$$\varepsilon_x + \varepsilon_y + \varepsilon_z = \frac{(1-2\nu)}{E} (\sigma_x + \sigma_y + \sigma_z). \quad (13)$$

Substituindo $(\sigma_y + \sigma_z)$ e assim sucessivamente nas expressões (11) obtém-se a relação entre σ e ε

$$\sigma = \mathbf{D}\varepsilon, \quad (14)$$

onde \mathbf{D} representa a matriz das propriedades do material, simétrica e de ordem 6, dada por

$$\mathbf{D} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & 0,5-\nu & 0 & 0 \\ 0 & 0 & 0 & 0 & 0,5-\nu & 0 \\ 0 & 0 & 0 & 0 & 0 & 0,5-\nu \end{bmatrix}. \quad (15)$$

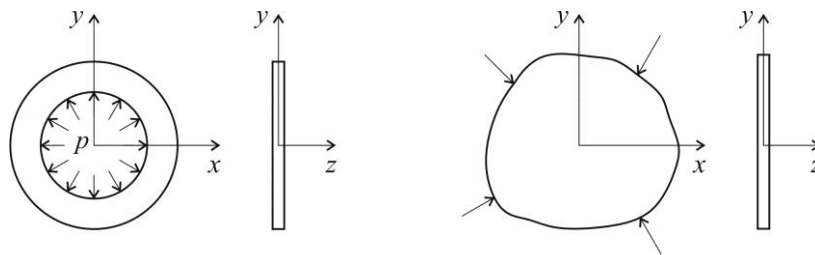
O procedimento de obtenção de \mathbf{D} é apresentado em detalhe em [4].

3.1.1.1.2 Caso bidimensional

Para a elasticidade bidimensional, as relações apresentadas podem ser simplificadas. Os problemas bidimensionais tipicamente podem ser simplificados para uma das duas seguintes formas: EPT ou EPD[9].

- Estado plano de tensão

Aplicável a corpos planos de espessura fina e constante, com carregamento no plano do corpo (plano xy). Considera-se por aproximação que a tensão normal segundo z é nula ao longo da espessura do componente, apesar da respetiva deformação não o ser necessariamente.



Esforços aplicados e aplicações possíveis da condição de estado plano de tensão

Observa-se $\sigma_z = \tau_{yz} = \tau_{zx} = 0$ e $\varepsilon_z \neq 0$. Como tal, as expressões apresentadas em (11) tomam a forma

$$\varepsilon_x = \frac{\sigma_x}{E} - \nu \frac{\sigma_y}{E}, \quad \varepsilon_y = \frac{\sigma_y}{E} - \nu \frac{\sigma_x}{E}, \quad \varepsilon_z = -\frac{\nu}{E}(\sigma_x + \sigma_y), \quad \gamma_{xy} = \frac{2(1+\nu)}{E} \tau_{xy}. \quad (16)$$

Resolvendo em ordem a σ obtém-se a relação (14), resultando

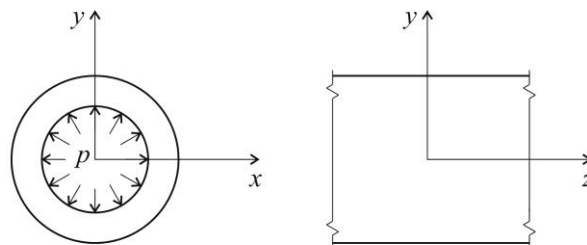
$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & (1-\nu)/2 \end{bmatrix} \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix}, \quad (17)$$

onde a matriz \mathbf{D} assume a forma

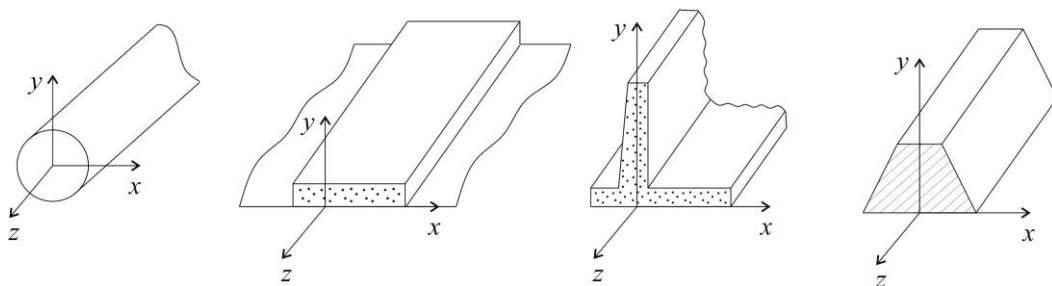
$$\mathbf{D} = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & (1-\nu)/2 \end{bmatrix}. \quad (18)$$

– Estado plano de deformação

Adequado para corpos planos de espessura elevada e secção reta constante ao longo da espessura, com carregamento tipicamente lateral (direções x e y) e com deformações nulas ou consideradas negligenciáveis na direção longitudinal (direção z). Nestas condições, com as exceções das extremidades da estrutura, o estado de tensão pode ser considerado constante em qualquer secção longitudinal, considerando nulo o deslocamento na direção longitudinal z .



(a) Esforços aplicados em estado plano de deformação



(b) Aplicações possíveis da condição de estado plano de deformação

Observa-se $\varepsilon_z = \gamma_{yz} = \gamma_{zx} = 0$ e $\sigma_z \neq 0$. As expressões (11) permitem escrever

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & 0 \\ \nu & 1-\nu & 0 \\ 0 & 0 & (1-2\nu)/2 \end{bmatrix} \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix}, \quad (19)$$

sendo \mathbf{D} igual a

$$\mathbf{D} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & 0 \\ \nu & 1-\nu & 0 \\ 0 & 0 & (1-2\nu)/2 \end{bmatrix}. \quad (20)$$

Caso unidimensional Considerando a análise unidimensional como simplificação do estado de tensão e deformação generalizado, apenas se consideram σ_x e ε_x , correspondentes ao corpo orientado segundo a direção x . Assim tem-se

$$\sigma_x = E\varepsilon_x. \quad (21)$$

3.1.2 Relações deformação-deslocamento

Tendo em consideração as 6 componentes que permitem definir o estado de tensão de um ponto, expressas em (9), é também possível definir o estado de deformação a partir das 6 componentes definidas em(10) [41].

Caso tridimensional Tomando os deslocamentos nas 3 direções ortogonais x , y e z como sendo u , v e w , respetivamente, são definidas as relações entre as deformações e deslocamentos da seguinte forma

$$\begin{aligned} \varepsilon_x &= \frac{\partial u}{\partial x} + \frac{1}{2} \left[\left(\frac{\partial u}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial x} \right)^2 + \left(\frac{\partial w}{\partial x} \right)^2 \right] & \gamma_{xy} &= \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} + \frac{\partial u}{\partial x} \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \frac{\partial v}{\partial y} + \frac{\partial w}{\partial x} \frac{\partial w}{\partial y} \\ \varepsilon_y &= \frac{\partial v}{\partial y} + \frac{1}{2} \left[\left(\frac{\partial u}{\partial y} \right)^2 + \left(\frac{\partial v}{\partial y} \right)^2 + \left(\frac{\partial w}{\partial y} \right)^2 \right] & \gamma_{yz} &= \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} + \frac{\partial u}{\partial y} \frac{\partial u}{\partial z} + \frac{\partial v}{\partial y} \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \frac{\partial w}{\partial z} \\ \varepsilon_z &= \frac{\partial w}{\partial z} + \frac{1}{2} \left[\left(\frac{\partial u}{\partial z} \right)^2 + \left(\frac{\partial v}{\partial z} \right)^2 + \left(\frac{\partial w}{\partial z} \right)^2 \right] & \gamma_{zx} &= \frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} + \frac{\partial u}{\partial x} \frac{\partial u}{\partial z} + \frac{\partial v}{\partial x} \frac{\partial v}{\partial z} + \frac{\partial w}{\partial x} \frac{\partial w}{\partial z}. \end{aligned} \quad (22)$$

A partir destas relações, sabe-se que as deformações e respetivas tensões são em um grau inferiores relativamente aos deslocamentos, se estes forem representados por funções

polinomiais. Nas expressões (22), as deformações são apresentadas para uma precisão de 2ª ordem (ordem quadrática) dos deslocamentos. Estas expressões podem no entanto ser simplificadas para 1ª ordem, eliminando os termos de 2ª ordem

$$\varepsilon_x = \frac{\partial u}{\partial x}, \quad \varepsilon_y = \frac{\partial v}{\partial y}, \quad \varepsilon_z = \frac{\partial w}{\partial z}, \quad \gamma_{xy} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}, \quad \gamma_{yz} = \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}, \quad \gamma_{zx} = \frac{\partial w}{\partial x} + \frac{\partial u}{\partial z}. \quad (23)$$

Deve-se ter em atenção que as expressões (22) devem ser sempre utilizadas para problemas com grandes deslocamentos ou em formulações que necessitem da consideração de grandes deslocamentos, enquanto as expressões (23) podem ser utilizadas com precisão em problemas que envolvam deslocamentos e flexões tipicamente reduzidos [40]. Neste último caso, pode-se escrever na forma matricial

$$\boldsymbol{\varepsilon} = \mathbf{B}\mathbf{a}_e \quad (24)$$

$$\boldsymbol{\varepsilon} = \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{zx} \end{Bmatrix} = \begin{bmatrix} \partial/\partial x & 0 & 0 \\ 0 & \partial/\partial y & 0 \\ 0 & 0 & \partial/\partial z \\ \partial/\partial y & \partial/\partial x & 0 \\ 0 & \partial/\partial z & \partial/\partial y \\ \partial/\partial z & 0 & \partial/\partial x \end{bmatrix} \begin{Bmatrix} u \\ v \\ w \end{Bmatrix} = \begin{Bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial w}{\partial z} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \\ \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \\ \frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} \end{Bmatrix}. \quad (25)$$

Caso bidimensional Considerando apenas pequenas deformações e rotações, tem-se

$$\varepsilon_x = \frac{\partial u}{\partial x}, \quad \varepsilon_y = \frac{\partial v}{\partial y}, \quad \gamma_{xy} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \quad (26)$$

$$\boldsymbol{\varepsilon} = \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} = \begin{bmatrix} \partial/\partial x & 0 \\ 0 & \partial/\partial y \\ \partial/\partial y & \partial/\partial x \end{bmatrix} \begin{Bmatrix} u \\ v \end{Bmatrix} = \begin{Bmatrix} \partial u/\partial x \\ \partial v/\partial y \\ \partial u/\partial y + \partial v/\partial x \end{Bmatrix}. \quad (27)$$

Caso unidimensional Para problemas unidimensionais como veios ou barras de estruturas articuladas, apenas se considera ε_x (corpo orientado segundo a direção x)

$$\varepsilon_x = \frac{\partial u}{\partial x}. \quad (28)$$

3.1.3 Equações de equilíbrio

A solução exata de um problema de engenharia (neste caso estrutural) é aquela que permite satisfazer as equações de equilíbrio, as condições fronteira e as condições de compatibilidade (estas últimas para garantir que as estruturas deformam continuamente, sem o aparecimento de fendas ou sobreposição de elementos). No entanto, apenas em casos de condições de geometria, carregamentos e condições fronteira muito simples é que se torna possível a obtenção de uma solução exata a partir das equações de equilíbrio da elasticidade. Nos casos mais comuns de engenharia, é necessária a utilização do MEF, que permite obter soluções aproximadas pelo estabelecimento de um sistema de equações que satisfaz, também de forma aproximada, as equações de equilíbrio nos nós dos elementos finitos.

Caso tridimensional As equações de equilíbrio para problemas sólidos estruturais podem ser obtidas por equilíbrio das forças que atuam num elemento infinitesimal nas 3 direções ortogonais do elemento, ou seja

$$\sum F_x = 0 \quad , \quad \sum F_y = 0 \quad , \quad \sum F_z = 0. \quad (29)$$

Considerando o equilíbrio de todas as forças envolvidas chega-se às 3 equações de equilíbrio que devem ser satisfeitas em todos os pontos da estrutura (é possível encontrar a descrição detalhada da dedução destas expressões em [41])

$$\begin{aligned} \frac{\partial \sigma_x}{\partial x} + \frac{\partial \tau_{xy}}{\partial y} + \frac{\partial \tau_{zx}}{\partial z} + f_x &= 0 \\ \frac{\partial \tau_{xy}}{\partial x} + \frac{\partial \sigma_y}{\partial y} + \frac{\partial \tau_{zy}}{\partial z} + f_y &= 0 \\ \frac{\partial \tau_{xz}}{\partial x} + \frac{\partial \tau_{yz}}{\partial y} + \frac{\partial \sigma_z}{\partial z} + f_z &= 0, \end{aligned} \quad (30)$$

onde f_x , f_y e f_z representam as forças por unidade de volume do elemento infinitesimal (como é o caso da força gravítica do elemento). As forças efetivas no elemento (X , Y e Z) são dadas por

$$\begin{aligned} X &= f_x dx dy dz \\ Y &= f_y dx dy dz \\ Z &= f_z dx dy dz. \end{aligned} \quad (31)$$

Aplicando as equações de equilíbrio, relativas ao centroide do elemento, dos momentos segundo os 3 eixos

$$\sum M_x = 0 \quad , \quad \sum M_y = 0 \quad , \quad \sum M_z = 0 \quad (32)$$

obtêm-se as seguintes igualdades (explicação detalhada em [41])

$$\begin{aligned} \tau_{yz} &= \tau_{zy} \\ \tau_{xz} &= \tau_{zx} \\ \tau_{xy} &= \tau_{yx} \end{aligned} \quad (33)$$

Caso bidimensional Por um procedimento semelhante, para casos bidimensionais tem-se que

$$\begin{aligned} \frac{\partial \sigma_x}{\partial x} + \frac{\partial \tau_{xy}}{\partial y} + f_x &= 0 \\ \frac{\partial \tau_{xy}}{\partial x} + \frac{\partial \sigma_y}{\partial y} + f_y &= 0. \end{aligned} \quad (34)$$

Caso unidimensional Efetuando as simplificações necessárias para uma dimensão

$$\frac{\partial \sigma_x}{\partial x} + f_x = 0. \quad (35)$$

3.2 Elementos bidimensionais

Os elementos estruturais sólidos que apresentem espessura constante podem ser analisados por elementos bidimensionais, que pressupõem à partida um valor constante de espessura. De notar que este tipo de elementos não necessita transformação de coordenadas, e cada nó apresenta à partida dois deslocamentos em direções ortogonais. Conforme apresentado na Secção 3.1.1, podem ser utilizadas condições de EPT ou EPD. Estes elementos são representados como corpos geométricos no plano, cujas formas mais comuns são o triângulo e o quadrilátero, e que possuem nas formulações mais simples 1 nó em cada vértice. Cada um dos nós dos elementos bidimensionais tem dois graus de liberdade: deslocamento segundo x , u , e deslocamento segundo y , v . A formulação destes elementos é de forma geral idêntica à apresentada anteriormente, com a determinação de \mathbf{N} , que permite o cálculo de \mathbf{B} e, por conseguinte, \mathbf{k}_e . O procedimento a seguir após determinação

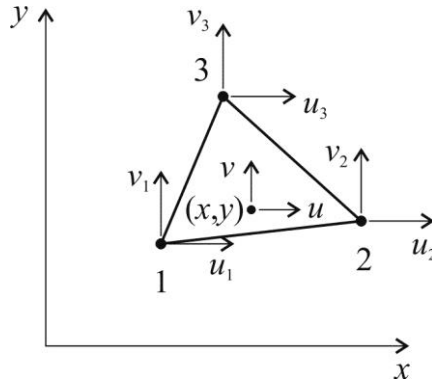
de \mathbf{k}_e é também idêntico, com a montagem do sistema global de equações e determinação das incógnitas em \mathbf{a}_e , por condensação do sistema, e cálculo das variáveis derivadas.

3.2.1 Elemento triangular de deformação constante de 3 nós

A forma triangular é a mais simples de considerar para um elemento sólido bidimensional. Para este elemento, a localização mais óbvia para os nós é nos vértices do elemento, o que conduz a um elemento com 3 nós e 6 variáveis nodais. Praticamente qualquer forma bidimensional pode ser discretizada com este tipo de elemento de forma expedita, embora os elementos não tenham necessariamente o mesmo tamanho e forma. De facto, foi devido a esta vantagem que o elemento triangular de 3 nós foi dos primeiros a serem desenvolvidos. A existência de apenas 3 nós para interpolar as variáveis de campo faz com que as funções de interpolação sejam lineares e que a formulação do elemento seja matematicamente muito simples. Por outro lado, as funções de forma lineares fazem com que as deformações, e correspondentemente as tensões, sejam constantes no interior do elemento. Esta característica pode trazer uma simplificação considerável ao estado de tensão da estrutura em zonas de variação abrupta de tensões, como em pontos de aplicação de cargas concentradas ou variações bruscas de geometria. Para se conseguir uma reprodução minimamente fiel do comportamento dos componentes nestas condições, é necessária a utilização de um número elevado de elementos, o que por sua vez leva ao aumento da dimensão dos sistemas de equações a resolver, e logicamente esforço computacional muito superior. Uma alternativa que permite a redução do número total de elementos sem prejudicar a precisão dos resultados é a utilização de elementos triangulares de ordem superior, por exemplo com 6 nós (3 nós nos vértices e 3 nós a meio das arestas do triângulo), cuja formulação pode ser consultada em [42] ou [4].

3.2.1.1 Matriz de rigidez do elemento – aproximação formal

Etapa 0 – O elemento triangular de deformação constante de 3 nós destina-se a resolver problemas de corpos sólidos bidimensionais com espessura, t , constante. O elemento tem arestas planas e um nó em cada vértice, cujas posições são dadas em coordenadas (x_i, y_i) . Cada nó tem deslocamentos segundo duas direções ortogonais, u_i e v_i [39]. Os nós devem ser numerados em sentido anti-horário no sentido de cálculo correto da área dos elementos [14].



Etapa 1 – O elemento tem forma triangular e apresenta 1 nó em cada vértice. A cada nó estão associadas as variáveis u e v , que representam os deslocamentos sobre direções ortogonais no referencial (x,y) . São apresentados os vetores de variáveis nodais e forças nodais

$$\mathbf{a}_e = \{u_1 \quad v_1 \quad u_2 \quad v_2 \quad u_3 \quad v_3\}^T, \quad \mathbf{f}_e = \{F_1^x \quad F_1^y \quad F_2^x \quad F_2^y \quad F_3^x \quad F_3^y\}^T. \quad (36)$$

Etapa 2 – Obtenção do campo de deslocamentos em função de \mathbf{N} e \mathbf{a}_e

$$\begin{Bmatrix} u \\ v \end{Bmatrix} = \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \end{Bmatrix} \quad \text{ou} \quad \begin{Bmatrix} u \\ v \end{Bmatrix} = \mathbf{N} \mathbf{a}_e. \quad (37)$$

Os polinômios a partir dos quais são obtidas as funções de interpolação para um elemento de duas dimensões, como é o caso dos elementos triangulares, são do tipo

$$P(x) = \alpha_1 + \alpha_2 x + \alpha_3 y + \alpha_4 x^2 + \alpha_5 xy + \alpha_6 y^2 \dots \quad (38)$$

O número de coeficientes do polinômio é igual ao número de variáveis nodais disponíveis para calcular os coeficientes, sendo como tal igual a 3 (três deslocamentos)

$$P(x) = \alpha_1 + \alpha_2 x + \alpha_3 y. \quad (39)$$

A expressão geral para o deslocamento u é igual a (de igual forma para o deslocamento v)

$$u = \alpha_1 + \alpha_2 x + \alpha_3 y. \quad (40)$$

De acordo com a expressão anterior, definem-se os deslocamentos u dos três nós constituintes do elemento

$$\begin{aligned} u_1 &= \alpha_1 + \alpha_2 x_1 + \alpha_3 y_1 \\ u_2 &= \alpha_1 + \alpha_2 x_2 + \alpha_3 y_2 \\ u_3 &= \alpha_1 + \alpha_2 x_3 + \alpha_3 y_3 \end{aligned} \quad (41)$$

e obtém-se assim \mathbf{a}_e em função da matriz \mathbf{C} e do vetor $\boldsymbol{\alpha}$ dos coeficientes de u

$$\begin{Bmatrix} u_1 \\ u_2 \\ u_3 \end{Bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix} \begin{Bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{Bmatrix} \quad \text{ou} \quad \mathbf{a}_e = \mathbf{C}\boldsymbol{\alpha}. \quad (42)$$

As funções de interpolação são obtidas por intermédio da expressão

$$\mathbf{N} = \mathbf{p}^T \mathbf{C}^{-1}, \quad \text{onde} \quad \mathbf{p}^T = \{1 \quad x \quad y\} \quad (43)$$

e \mathbf{C}^{-1} é igual a

$$\mathbf{C}^{-1} = \frac{1}{2\Delta} \begin{bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{bmatrix}, \quad \begin{array}{l} \Delta = \text{área do triângulo} \\ a_1 = x_2 y_3 - x_3 y_2 \\ a_2 = x_3 y_1 - x_1 y_3 \\ a_3 = x_1 y_2 - x_2 y_1 \end{array}, \quad \begin{array}{l} b_1 = y_2 - y_3 \\ b_2 = y_3 - y_1 \\ b_3 = y_1 - y_2 \end{array}, \quad \begin{array}{l} c_1 = x_3 - x_2 \\ c_2 = x_1 - x_3 \\ c_3 = x_2 - x_1 \end{array}. \quad (44)$$

A área do triângulo (Δ) é igual a

$$\Delta = \frac{1}{2} \det \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix}. \quad (45)$$

Como tal

$$\mathbf{N} = \{1 \quad x \quad y\} \frac{1}{2\Delta} \begin{bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{bmatrix} \quad (46)$$

$$\mathbf{N} = [N_1 \quad N_2 \quad N_3] = \frac{1}{2\Delta} [a_1 + b_1 x + c_1 y \quad a_2 + b_2 x + c_2 y \quad a_3 + b_3 x + c_3 y]. \quad (47)$$

Pela expressão (37) tem-se

$$\begin{aligned}
 u &= \frac{1}{2\Delta} \left[[a_1 + b_1x + c_1y]u_1 + [a_2 + b_2x + c_2y]u_2 + [a_3 + b_3x + c_3y]u_3 \right] \\
 v &= \frac{1}{2\Delta} \left[[a_1 + b_1x + c_1y]v_1 + [a_2 + b_2x + c_2y]v_2 + [a_3 + b_3x + c_3y]v_3 \right].
 \end{aligned} \tag{48}$$

As expressões (48) permitem obter os deslocamentos u e v em qualquer ponto do elemento em função de \mathbf{N} e \mathbf{a}_e

Etapa 3 – Definir o vetor de deformações $\boldsymbol{\varepsilon}$ em função da matriz de deformação \mathbf{B} e das variáveis nodais \mathbf{a}_e . Recuperando a expressão (27) e considerando a expressão (37)

$$\boldsymbol{\varepsilon} = \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} = \begin{bmatrix} \partial/\partial x & 0 \\ 0 & \partial/\partial y \\ \partial/\partial y & \partial/\partial x \end{bmatrix} \begin{Bmatrix} u \\ v \end{Bmatrix} = \begin{bmatrix} \partial/\partial x & 0 \\ 0 & \partial/\partial y \\ \partial/\partial y & \partial/\partial x \end{bmatrix} \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \end{Bmatrix}. \tag{49}$$

$$\text{Como } \boldsymbol{\varepsilon} = \mathbf{B}\mathbf{a}_e \Rightarrow \mathbf{B} = \begin{bmatrix} \partial/\partial x & 0 \\ 0 & \partial/\partial y \\ \partial/\partial y & \partial/\partial x \end{bmatrix} \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 \end{bmatrix} \tag{50}$$

$$\mathbf{B} = \begin{bmatrix} \frac{\partial N_1}{\partial x} & 0 & \frac{\partial N_2}{\partial x} & 0 & \frac{\partial N_3}{\partial x} & 0 \\ 0 & \frac{\partial N_1}{\partial y} & 0 & \frac{\partial N_2}{\partial y} & 0 & \frac{\partial N_3}{\partial y} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial y} & \frac{\partial N_2}{\partial x} & \frac{\partial N_3}{\partial y} & \frac{\partial N_3}{\partial x} \end{bmatrix}. \tag{51}$$

Substituindo (47) em (51) e por aplicação dos operadores diferenciais

$$\mathbf{B} = \frac{1}{2\Delta} \begin{bmatrix} b_1 & 0 & b_2 & 0 & b_3 & 0 \\ 0 & c_1 & 0 & c_2 & 0 & c_3 \\ c_1 & b_1 & c_2 & b_2 & c_3 & b_3 \end{bmatrix}. \tag{52}$$

Etapa 4 – Definir o vetor de tensões $\boldsymbol{\sigma}$ em função do vetor de deformações $\boldsymbol{\varepsilon}$ (expressão (14)). A matriz \mathbf{D} é dada por (18), caso se considere EPT ou (20), caso se considere EPD.

Etapa 5 – Definir a matriz de rigidez do elemento \mathbf{k}_e através de

$$\mathbf{k}_e = \int_V \mathbf{B}^T \mathbf{D} \mathbf{B} dv \Rightarrow \mathbf{k}_e = \mathbf{B}^T \mathbf{D} \mathbf{B} \Delta t. \quad (53)$$

O cálculo de \mathbf{k}_e , matriz de ordem 6, pode ser efetuado recorrendo a programas de cálculo simbólico, devido à sua morosidade. Como exemplo, são apresentados os elementos k_{11} e k_{12}

$$k_{11} = \frac{Et}{4(1-\nu^2)\Delta} \left(b_1 + \frac{1-\nu}{2} c_1^2 \right), \quad k_{12} = \frac{Et}{4(1-\nu^2)\Delta} \left(\nu b_1 c_1 + \frac{1-\nu}{2} b_1 c_1 \right). \quad (54)$$

Uma alternativa à utilização das expressões (54) consiste em calcular \mathbf{k}_e individualmente para cada elemento pela expressão (53). Para tal, são utilizados Δ , t , \mathbf{D} e \mathbf{B} para o elemento em causa, em função da geometria do mesmo.

3.2.1.2 Matriz de rigidez do elemento – aproximação isoparamétrica

Na Secção anterior foi formulada \mathbf{k}_e para o elemento triangular de deformação constante de 3 nós pela *aproximação formal*, observando-se desde logo a dificuldade de manipulação das expressões e matrizes. Tal tarefa, para elementos mais complexos, como por exemplo o elemento triangular de deformação linear de 3 nós, ainda é possível de realizar [4], mas torna-se inviável para elementos de grau superior. Para ultrapassar estas dificuldades, foi desenvolvida a *aproximação isoparamétrica* [43] que também permite uma implementação mais expedita. Por outro lado, é também possibilitada a formulação de elementos finitos com arestas curvas. Na presente Secção, é descrita esta formulação para efeitos de comparação com a *aproximação formal*.

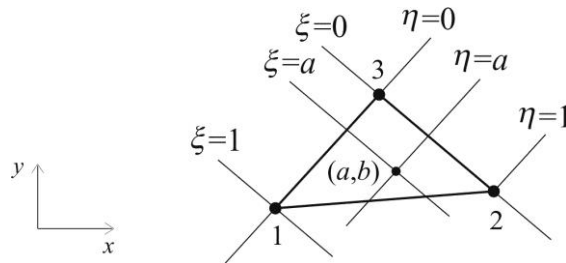
Na *aproximação formal*, a determinação das funções de interpolação consistiu na definição dos deslocamentos no interior do elemento como sendo resultado de funções polinomiais, que por sua vez são função de constantes desconhecidas. Estas constantes são depois determinadas usando os deslocamentos dos nós do elemento respetivo. Este processo, para elementos mais complexos (com maior número de nós e/ou variáveis nodais) torna-se bastante moroso. Na *aproximação isoparamétrica* usam-se coordenadas naturais (ξ , η ; tipicamente com origem no centroide do elemento e comprimento normalizado) em detrimento das globais, e que são definidas pela geometria do elemento e não pela sua

orientação no sistema de eixos global. Consideram-se também funções para relacionar as coordenadas globais genéricas (x,y) com as coordenadas globais dos nós idênticas às funções de interpolação que são usadas para definir os deslocamentos no interior do elemento em função dos deslocamentos nodais. As funções de interpolação podem ser definidas por manipulação matemática entre as coordenadas (x,y) e (ξ,η) (formulação seguida em Logan (2007)), ou alternativamente de forma empírica desde que tendo em consideração as suas propriedades (formulação desenvolvida em [40] ou [9])

$$N_i = \begin{cases} 1 & \text{no nó } i \\ 0 & \text{nos restantes nós} \end{cases} \quad (55)$$

$$\sum_{i=1}^n N_i = 1, \quad \text{com } n - \text{número de nós do elemento.} \quad (56)$$

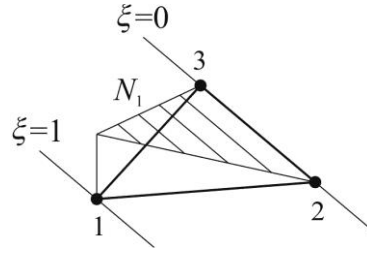
Seguindo esta segunda abordagem para o elemento triangular de deformação constante, efetua-se uma mudança das coordenadas (x,y) para as coordenadas (ξ,η) como se apresenta na figura.



Neste caso, as funções de interpolação em coordenadas locais podem ser definidas da seguinte forma

$$N_1 = \xi \quad , \quad N_2 = \eta \quad , \quad N_3 = 1 - \xi - \eta. \quad (57)$$

Facilmente se observa que as funções de interpolação apresentadas satisfazem as condições definidas. De facto, estas são iguais à unidade no nó respetivo e iguais a 0 nos restantes (55), e a sua soma é igual à unidade (56). É apresentada uma representação de N_1 ao longo do elemento. N_2 e N_3 apresentam características idênticas.



As coordenadas (x,y) são definidas usando as funções de interpolação e as coordenadas nodais (58), e os deslocamentos no interior do elemento são também definidos pelas funções de interpolação e deslocamentos nodais, desconhecidos neste momento (59).

$$\begin{aligned} x &= N_1(\xi, \eta)x_1 + N_2(\xi, \eta)x_2 + N_3(\xi, \eta)x_3 \\ y &= N_1(\xi, \eta)y_1 + N_2(\xi, \eta)y_2 + N_3(\xi, \eta)y_3 \end{aligned} \quad (58)$$

$$\begin{aligned} u &= N_1(\xi, \eta)u_1 + N_2(\xi, \eta)u_2 + N_3(\xi, \eta)u_3 \\ v &= N_1(\xi, \eta)v_1 + N_2(\xi, \eta)v_2 + N_3(\xi, \eta)v_3. \end{aligned} \quad (59)$$

Conforme já foi referido, a formulação é denominada de *isoparamétrica* devido ao facto de serem utilizadas as mesmas funções para interpolar a posição de cada ponto no interior do elemento em coordenadas globais (58) e para interpolar as variáveis em questão no problema (para problemas estruturais e este caso em particular u e v). Substituindo (57) em (58) obtém-se

$$\begin{aligned} x &= x_{13}\xi + x_{23}\eta + x_3 \\ y &= y_{13}\xi + y_{23}\eta + y_3 \end{aligned}, \quad \text{com } x_{ij} = x_i - x_j \text{ e } y_{ij} = y_i - y_j. \quad (60)$$

Por outro lado, substituindo (57) em (59) chega-se a

$$\begin{aligned} u &= u_{13}\xi + u_{23}\eta + u_3 \\ v &= v_{13}\xi + v_{23}\eta + v_3 \end{aligned}, \quad \text{com } u_{ij} = u_i - u_j \text{ e } v_{ij} = v_i - v_j. \quad (61)$$

As expressões(60) relacionam as coordenadas (x,y) com as coordenadas (ξ, η) , enquanto as expressões(61) definem os deslocamentos u e v em coordenadas (ξ, η) . Para a determinação de \mathbf{B} , pela expressão (24), e sabendo que $\boldsymbol{\epsilon}$ é dado por (27) para casos de elasticidade bidimensional, é necessária a derivação parcial de u e v em relação a x e y . No entanto, u e v encontram-se expressos em função das coordenadas naturais ξ e η (61), o que obriga à utilização da regra da corrente para as derivadas parciais de u (posteriormente é feita a generalização para v)

$$\frac{\partial u}{\partial \xi} = \frac{\partial u}{\partial x} \frac{\partial x}{\partial \xi} + \frac{\partial u}{\partial y} \frac{\partial y}{\partial \xi}, \quad \frac{\partial u}{\partial \eta} = \frac{\partial u}{\partial x} \frac{\partial x}{\partial \eta} + \frac{\partial u}{\partial y} \frac{\partial y}{\partial \eta}. \quad (62)$$

Escrito em notação matricial

$$\begin{Bmatrix} \frac{\partial u}{\partial \xi} \\ \frac{\partial u}{\partial \eta} \end{Bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} \begin{Bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial u}{\partial y} \end{Bmatrix}, \quad (63)$$

onde a matriz 2×2 é denominada de *Jacobiano* da transformação, \mathbf{J}

$$\mathbf{J} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix}. \quad (64)$$

Por diferenciação das expressões (60) em ordem a ξ e η , \mathbf{J} toma a forma

$$\mathbf{J} = \begin{bmatrix} x_{13} & y_{13} \\ x_{23} & y_{23} \end{bmatrix}. \quad (65)$$

Assim, tendo em consideração (63) tem-se

$$\begin{Bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial u}{\partial y} \end{Bmatrix} = \mathbf{J}^{-1} \begin{Bmatrix} \frac{\partial u}{\partial \xi} \\ \frac{\partial u}{\partial \eta} \end{Bmatrix}, \quad (66)$$

onde \mathbf{J}^{-1} representa o inverso do Jacobiano, dado por

$$\mathbf{J}^{-1} = \frac{1}{\det \mathbf{J}} \begin{bmatrix} y_{23} & -y_{13} \\ -x_{23} & x_{13} \end{bmatrix}, \quad \text{com } \det \mathbf{J} = x_{13}y_{23} - x_{23}y_{13}. \quad (67)$$

Desenvolvendo as expressões (45) e (67), observa-se que

$$|\det \mathbf{J}| = 2\Delta. \quad (68)$$

Se os nós do elemento forem numerados no sentido anti-horário, pode-se ainda escrever

$$\det \mathbf{J} = 2\Delta. \quad (69)$$

A partir de (66) e de (67) obtém-se (generalizando também para v)

$$\begin{Bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial u}{\partial y} \end{Bmatrix} = \frac{1}{\det \mathbf{J}} \begin{Bmatrix} y_{23} \frac{\partial u}{\partial \xi} - y_{13} \frac{\partial u}{\partial \eta} \\ -x_{23} \frac{\partial u}{\partial \xi} + x_{13} \frac{\partial u}{\partial \eta} \end{Bmatrix} \quad \text{e} \quad \begin{Bmatrix} \frac{\partial v}{\partial x} \\ \frac{\partial v}{\partial y} \end{Bmatrix} = \frac{1}{\det \mathbf{J}} \begin{Bmatrix} y_{23} \frac{\partial v}{\partial \xi} - y_{13} \frac{\partial v}{\partial \eta} \\ -x_{23} \frac{\partial v}{\partial \xi} + x_{13} \frac{\partial v}{\partial \eta} \end{Bmatrix}. \quad (70)$$

Encontrando-se determinadas as derivadas parciais de u e v relativamente às coordenadas x e y , \mathbf{B} pode ser determinada tendo em consideração (27), (61) e (70), e recordando (69)

$$\boldsymbol{\varepsilon} = \begin{Bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \end{Bmatrix} = \frac{1}{2\Delta} \begin{Bmatrix} y_{23}u_{13} - y_{13}u_{23} \\ -x_{23}v_{13} + x_{13}v_{23} \\ -x_{23}u_{13} + x_{13}u_{23} + y_{23}v_{13} - y_{13}v_{23} \end{Bmatrix}. \quad (71)$$

Decompondo u_{ij} e v_{ij} , e sabendo que [40]

$$y_{13} - y_{23} = y_{12} ; x_{23} - x_{13} = x_{21} ; x_{ij} = -x_{ji}, \quad (72)$$

obté-m-se

$$\boldsymbol{\varepsilon} = \frac{1}{2\Delta} \begin{Bmatrix} y_{23}u_1 + y_{31}u_2 + y_{12}u_3 \\ x_{32}v_1 + x_{13}v_2 + x_{21}v_3 \\ x_{32}u_1 + y_{23}v_1 + x_{13}u_2 + y_{31}v_2 + x_{21}u_3 + y_{12}v_3 \end{Bmatrix} \quad (73)$$

ou ainda, em função de \mathbf{a}_e

$$\boldsymbol{\varepsilon} = \frac{1}{2\Delta} \begin{bmatrix} y_{23} & 0 & y_{31} & 0 & y_{12} & 0 \\ 0 & x_{32} & 0 & x_{13} & 0 & x_{21} \\ -x_{32} & y_{23} & x_{13} & y_{31} & x_{21} & y_{12} \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \end{Bmatrix}. \quad (74)$$

$$\text{Como } \boldsymbol{\varepsilon} = \mathbf{B}\mathbf{a}_e \Rightarrow \mathbf{B} = \frac{1}{2\Delta} \begin{bmatrix} y_{23} & 0 & y_{31} & 0 & y_{12} & 0 \\ 0 & x_{32} & 0 & x_{13} & 0 & x_{21} \\ -x_{32} & y_{23} & x_{13} & y_{31} & x_{21} & y_{12} \end{bmatrix}, \quad (75)$$

pelo que, pelas igualdades expressas em(44), se comprova que \mathbf{B} é idêntica à obtida em (52).

3.2.1.3 Carregamentos equivalentes

Os esforços em elementos planos podem ocorrer sob a forma de esforços concentrados, pressões ou esforços distribuídos, ou ainda forças por unidade de volume, como é o caso das forças gravíticas. No caso dos esforços concentrados, estes podem ser diretamente

aplicados ao modelo de elementos finitos em nós da malha. Em problemas de elasticidade plana, os esforços distribuídos podem atuar em fronteiras internas ou externas da malha e apresentar orientação arbitrária, decomposta em componentes normal e tangencial à fronteira. As forças por unidade de volume atuam sobre a totalidade do modelo. Nestes dois últimos casos, o procedimento consiste na conversão dos esforços em esforços nodais equivalentes, pelo princípio da *Equivalência de Trabalho*. Uma vez que a presente obra não aborda este processo, para informação mais detalhada o leitor interessado nesta matéria poderá consultar os livros de [44] ou [4].

3.2.1.4 Estado de tensão nos elementos

O elemento triangular tem deformações (e conseqüentemente tensões) constantes no seu domínio, devido a **B** ser constante (expressões (52) e (75)). Como tal, σ é caracterizado por σ_x , σ_y e τ_{xy} , cujos valores são calculados por (14), sabendo que ϵ é dado por(24). Como tal tem-se

$$\sigma_e = \mathbf{DBa}_e \quad e = 1, 2, \dots, n \text{ (n - número de elementos)}, \quad (76)$$

onde **D** depende da consideração de EPT (18) ou EPD(20).

Em função do estado de tensão obtido, é possível obter a tensão equivalente de von Mises pela expressão:

$$\sigma_{VM} = \sqrt{\sigma_{xx}^2 + \sigma_{yy}^2 - \sigma_{xx}\sigma_{yy} + 3\tau_{xy}^2} \quad (77)$$

Da propriedade do material σ_{ced} e da tensão máxima pode ser obtida o fator de utilização, F.S.(Fator de Segurança), calculado a partir da equação:

$$FS = \frac{\sigma_{ced}}{\sigma_{max}} \quad (78)$$

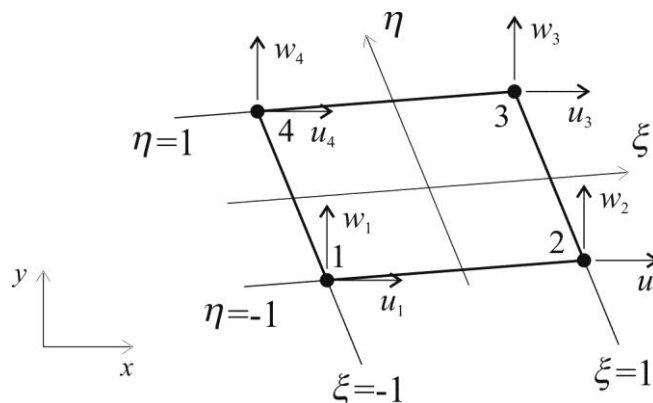
3.2.2 Elemento quadrangular de deformação linear de 4 nós

Este elemento destina-se a resolver problemas em sólidos bidimensionais com espessura constante. Os elementos finitos com forma quadrangular apresentam um maior número de nós e graus de liberdade comparativamente aos elementos finitos triangulares equivalentes (por exemplo, comparando elementos com nós apenas nos vértices, o quadrilátero tem

mais 1 nó). Como tal, as funções de interpolação apresentam um grau superior, o que conduz a uma resolução maior das deformações e tensões no interior do elemento. No entanto, a formulação é mais complexa, já que apenas para o elemento apresentado nesta obra é ainda possível o cálculo explícito dos termos de \mathbf{k}_e . Para os elementos de ordem superior, os termos de \mathbf{k}_e são dados por expressões integrais que devem ser avaliadas por métodos numéricos. O elemento quadrangular de deformação linear de 4 nós, abordado neste livro, é o mais simples mas também o menos preciso na presença de variações bruscas das deformações e tensões, pois apenas admite uma variação linear com x e y das variáveis de campo no seu interior. Cada elemento apresenta arestas planas e deformação linear no seu interior. Tal como o elemento triangular de deformação constante, cada nó deste elemento possui duas coordenadas (x_i, y_i) e deslocamentos segundo duas direções ortogonais (u_i, v_i) . Devido à morosidade associada à *aproximação formal*, é apresentada apenas a *formulação isoparamétrica*. Para mais detalhes sobre a *aproximação formal*, pode ser consultado o livro de Logan [4]. A utilização de elementos de *Serendipity* com 8 nós ou elementos de *Lagrange* com 9 nós já permitiria um maior gradiente de deformações e tensões no interior do elemento, bem como modelação geométrica com arestas curvas de variação quadrática [39].

3.2.2.1 Matriz de rigidez do elemento – aproximação isoparamétrica

A formulação apresentada para o elemento quadrangular de deformação linear de 4 nós poderá ser adaptada para elementos de ordem superior, como sejam o elemento quadrático de 8 nós, que apresenta a possibilidade de modelação de arestas curvas (Gockenbach 2006). São apresentados no esquema seguinte o sistema de coordenadas globais (x,y) e o sistema de coordenadas naturais (ξ,η) para o elemento.



O centro do sistema de coordenadas (ξ, η) está localizado no centroide do elemento. Refere-se que os eixos ξ e η não têm de ser ortogonais nem paralelos aos eixos globais. É no entanto necessário que os nós nos vértices do elemento e as arestas se encontrem limitados pelos valores de +1 ou -1, para facilitar a integração numérica de \mathbf{k}_e . O elemento tem 8 graus de liberdade, $u_1, v_1, \dots, u_4, v_4$, associados com as direções globais x e y , e arestas retilíneas, embora estas não tenham de ser paralelas nem ortogonais. As funções de interpolação podem ser definidas como se segue, tendo em consideração as duas propriedades que estas devem satisfazer, expressões (55) e (56)

$$N_1 = \frac{1}{4}(1-\xi)(1-\eta), \quad N_2 = \frac{1}{4}(1+\xi)(1-\eta), \quad N_3 = \frac{1}{4}(1+\xi)(1+\eta), \quad N_4 = \frac{1}{4}(1-\xi)(1+\eta). \quad (79)$$

As coordenadas (x, y) são definidas usando as coordenadas nodais (80), e os deslocamentos no interior do elemento são também definidos pelos deslocamentos nodais (81)

$$\begin{aligned} x &= N_1(\xi, \eta)x_1 + N_2(\xi, \eta)x_2 + N_3(\xi, \eta)x_3 + N_4(\xi, \eta)x_4 \\ y &= N_1(\xi, \eta)y_1 + N_2(\xi, \eta)y_2 + N_3(\xi, \eta)y_3 + N_4(\xi, \eta)y_4 \end{aligned} \quad (80)$$

$$\begin{aligned} u &= N_1(\xi, \eta)u_1 + N_2(\xi, \eta)u_2 + N_3(\xi, \eta)u_3 + N_4(\xi, \eta)u_4 \\ v &= N_1(\xi, \eta)v_1 + N_2(\xi, \eta)v_2 + N_3(\xi, \eta)v_3 + N_4(\xi, \eta)v_4. \end{aligned} \quad (81)$$

Substituindo (79) em (80) obtém-se

$$\begin{aligned} x &= \frac{1}{4}[(1-\xi)(1-\eta)x_1 + (1+\xi)(1-\eta)x_2 + (1+\xi)(1+\eta)x_3 + (1-\xi)(1+\eta)x_4] \\ y &= \frac{1}{4}[(1-\xi)(1-\eta)y_1 + (1+\xi)(1-\eta)y_2 + (1+\xi)(1+\eta)y_3 + (1-\xi)(1+\eta)y_4]. \end{aligned} \quad (82)$$

Substituindo (79) em (81) chega-se a

$$\begin{aligned} u &= \frac{1}{4}[(1-\xi)(1-\eta)u_1 + (1+\xi)(1-\eta)u_2 + (1+\xi)(1+\eta)u_3 + (1-\xi)(1+\eta)u_4] \\ v &= \frac{1}{4}[(1-\xi)(1-\eta)v_1 + (1+\xi)(1-\eta)v_2 + (1+\xi)(1+\eta)v_3 + (1-\xi)(1+\eta)v_4]. \end{aligned} \quad (83)$$

Torna-se necessária a obtenção das derivadas das expressões (83) em função a x e y (expressão (27)), para obtenção de \mathbf{B} pela expressão (24). Assim, utilizando a regra da corrente para as derivadas parciais de u tem-se (posteriormente é feita a generalização para v)

$$\frac{\partial u}{\partial \xi} = \frac{\partial u}{\partial x} \frac{\partial x}{\partial \xi} + \frac{\partial u}{\partial y} \frac{\partial y}{\partial \xi}, \quad \frac{\partial u}{\partial \eta} = \frac{\partial u}{\partial x} \frac{\partial x}{\partial \eta} + \frac{\partial u}{\partial y} \frac{\partial y}{\partial \eta}. \quad (84)$$

Nas expressões (84), todos os termos são possíveis de determinar por (82) e (83) exceto $\partial u/\partial x$ e $\partial u/\partial y$, precisamente os termos necessários para a definição das deformações pela expressão (27). Escrevendo as expressões (84) em notação matricial

$$\begin{Bmatrix} \frac{\partial u}{\partial \xi} \\ \frac{\partial u}{\partial \eta} \end{Bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} \begin{Bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial u}{\partial y} \end{Bmatrix}, \quad (85)$$

onde o Jacobiano da transformação, \mathbf{J} , toma a forma

$$\mathbf{J} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix}. \quad (86)$$

Por diferenciação das expressões (82) em ordem a ξ e η , \mathbf{J} resulta em

$$\mathbf{J} = \frac{1}{4} \begin{bmatrix} -(1-\eta)x_1 + (1-\eta)x_2 + (1+\eta)x_3 - (1+\eta)x_4 & -(1-\eta)y_1 + (1-\eta)y_2 + (1+\eta)y_3 - (1+\eta)y_4 \\ -(1-\xi)x_1 - (1+\xi)x_2 + (1+\xi)x_3 + (1-\xi)x_4 & -(1-\xi)y_1 - (1+\xi)y_2 + (1+\xi)y_3 + (1-\xi)y_4 \end{bmatrix} \quad (87)$$

$$\mathbf{J} = \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix}. \quad (88)$$

Assim, resolvendo (85) em ordem às incógnitas

$$\begin{Bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial u}{\partial y} \end{Bmatrix} = \mathbf{J}^{-1} \begin{Bmatrix} \frac{\partial u}{\partial \xi} \\ \frac{\partial u}{\partial \eta} \end{Bmatrix}. \quad (89)$$

Mais uma vez calcula-se \mathbf{J}^{-1} como sendo

$$\mathbf{J}^{-1} = \frac{1}{\det \mathbf{J}} \begin{bmatrix} J_{22} & -J_{12} \\ -J_{21} & J_{11} \end{bmatrix}, \quad \text{com } \det \mathbf{J} = J_{11}J_{22} - J_{21}J_{12}. \quad (90)$$

Assim tem-se (generalizando também para v)

$$\begin{Bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial u}{\partial y} \end{Bmatrix} = \frac{1}{\det \mathbf{J}} \begin{bmatrix} J_{22} & -J_{12} \\ -J_{21} & J_{11} \end{bmatrix} \begin{Bmatrix} \frac{\partial u}{\partial \xi} \\ \frac{\partial u}{\partial \eta} \end{Bmatrix} \quad \text{e} \quad \begin{Bmatrix} \frac{\partial v}{\partial x} \\ \frac{\partial v}{\partial y} \end{Bmatrix} = \frac{1}{\det \mathbf{J}} \begin{bmatrix} J_{22} & -J_{12} \\ -J_{21} & J_{11} \end{bmatrix} \begin{Bmatrix} \frac{\partial v}{\partial \xi} \\ \frac{\partial v}{\partial \eta} \end{Bmatrix}. \quad (91)$$

A matriz \mathbf{B} pode ser calculada por (27)e(91)

$$\boldsymbol{\varepsilon} = \begin{Bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \end{Bmatrix} = \frac{1}{\det \mathbf{J}} \begin{Bmatrix} J_{22} \frac{\partial u}{\partial \xi} - J_{12} \frac{\partial u}{\partial \eta} \\ -J_{21} \frac{\partial v}{\partial \xi} + J_{11} \frac{\partial v}{\partial \eta} \\ -J_{21} \frac{\partial u}{\partial \xi} + J_{11} \frac{\partial u}{\partial \eta} + J_{22} \frac{\partial v}{\partial \xi} - J_{12} \frac{\partial v}{\partial \eta} \end{Bmatrix}, \quad (92)$$

ou de forma condensada

$$\boldsymbol{\varepsilon} = \mathbf{A} \begin{Bmatrix} \frac{\partial u}{\partial \xi} \\ \frac{\partial u}{\partial \eta} \\ \frac{\partial v}{\partial \xi} \\ \frac{\partial v}{\partial \eta} \end{Bmatrix}, \quad \text{com} \quad \mathbf{A} = \frac{1}{\det \mathbf{J}} \begin{bmatrix} J_{22} & -J_{12} & 0 & 0 \\ 0 & 0 & -J_{21} & J_{11} \\ -J_{21} & J_{11} & J_{22} & -J_{12} \end{bmatrix}. \quad (93)$$

Por diferenciação das expressões (83) escreve-se

$$\begin{Bmatrix} \frac{\partial u}{\partial \xi} \\ \frac{\partial u}{\partial \eta} \\ \frac{\partial v}{\partial \xi} \\ \frac{\partial v}{\partial \eta} \end{Bmatrix} = \frac{1}{4} \begin{Bmatrix} -(1-\eta)u_1 + (1-\eta)u_2 + (1+\eta)u_3 - (1+\eta)u_4 \\ -(1-\xi)u_1 - (1+\xi)u_2 + (1+\xi)u_3 + (1-\xi)u_4 \\ -(1-\eta)v_1 + (1-\eta)v_2 + (1+\eta)v_3 - (1+\eta)v_4 \\ -(1-\xi)v_1 - (1+\xi)v_2 + (1+\xi)v_3 + (1-\xi)v_4 \end{Bmatrix}. \quad (94)$$

Assim define-se

$$\begin{Bmatrix} \frac{\partial u}{\partial \xi} \\ \frac{\partial u}{\partial \eta} \\ \frac{\partial v}{\partial \xi} \\ \frac{\partial v}{\partial \eta} \end{Bmatrix} = \mathbf{G}\mathbf{a}^e = \frac{1}{4} \begin{bmatrix} -(1-\eta) & 0 & (1-\eta) & 0 & (1+\eta) & 0 & -(1+\eta) & 0 \\ -(1-\xi) & 0 & -(1+\xi) & 0 & (1+\xi) & 0 & (1-\xi) & 0 \\ 0 & -(1-\eta) & 0 & (1-\eta) & 0 & (1+\eta) & 0 & -(1+\eta) \\ 0 & -(1-\xi) & 0 & -(1+\xi) & 0 & (1+\xi) & 0 & (1-\xi) \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \\ u_4 \\ v_4 \end{Bmatrix}, \quad (95)$$

onde

$$\mathbf{G} = \frac{1}{4} \begin{bmatrix} -(1-\eta) & 0 & (1-\eta) & 0 & (1+\eta) & 0 & -(1+\eta) & 0 \\ -(1-\xi) & 0 & -(1+\xi) & 0 & (1+\xi) & 0 & (1-\xi) & 0 \\ 0 & -(1-\eta) & 0 & (1-\eta) & 0 & (1+\eta) & 0 & -(1+\eta) \\ 0 & -(1-\xi) & 0 & -(1+\xi) & 0 & (1+\xi) & 0 & (1-\xi) \end{bmatrix}. \quad (96)$$

Recuperando (93) e relembando a expressão (24) tem-se

$$\boldsymbol{\varepsilon} = \mathbf{A}\mathbf{G}\mathbf{a}_e, \quad \text{logo} \quad \mathbf{B} = \mathbf{A}\mathbf{G}. \quad (97)$$

O método utilizado no elemento triangular de deformação constante de 3 nós para a determinação de \mathbf{k}_e consistiu na utilização da expressão(53). No entanto, neste caso observa-se que \mathbf{B} e $\det\mathbf{J}$ são função das coordenadas naturais (ξ, η) . Como tal, torna-se necessário determinar \mathbf{k}_e em função das coordenadas naturais [45], [46]. Para tal, é utilizada a energia de deformação do corpo, dada por

$$U = \int_V \frac{1}{2} \boldsymbol{\sigma}^T \boldsymbol{\varepsilon} dV. \quad (98)$$

Decompondo para cada elemento

$$U = \sum_e t_e \int_{V_e} \frac{1}{2} \boldsymbol{\sigma}^T \boldsymbol{\varepsilon} dA, \quad (99)$$

onde t_e representa a espessura do elemento e . Sabe-se também que [40]

$$dA = dx dy = \det \mathbf{J} d\xi d\eta. \quad (100)$$

Por outro lado (relembando que \mathbf{D} é uma matriz simétrica)

$$\boldsymbol{\sigma} = \mathbf{D}\mathbf{B}\mathbf{a}_e \quad (101)$$

$$\boldsymbol{\sigma}^T = (\mathbf{D}\mathbf{B}\mathbf{a}_e)^T = \mathbf{a}_e^T \mathbf{B}^T \mathbf{D}^T = \mathbf{a}_e^T \mathbf{B}^T \mathbf{D}. \quad (102)$$

Recuperando (99) e considerando as expressões (100)e(102)

$$U = \sum_e t_e \int_A \frac{1}{2} \mathbf{a}_e^T \mathbf{B}^T \mathbf{D} \mathbf{B} \mathbf{a}_e \det \mathbf{J} d\xi d\eta \quad (103)$$

$$U = \sum_e \frac{1}{2} \mathbf{a}_e^T \left[t_e \int_{-1}^1 \int_{-1}^1 \mathbf{B}^T \mathbf{D} \mathbf{B} \det \mathbf{J} d\xi d\eta \right] \mathbf{a}_e. \quad (104)$$

Como por definição

$$U = \sum_e \frac{1}{2} \mathbf{a}_e^T \mathbf{k}_e \mathbf{a}_e \quad (105)$$

obtem-se

$$\mathbf{k}_e = t_e \int_{-1}^1 \int_{-1}^1 \mathbf{B}^T \mathbf{D} \mathbf{B} \det \mathbf{J} d\xi d\eta. \quad (106)$$

\mathbf{k}_e representa a matriz de rigidez do elemento, de dimensão 8. Como \mathbf{B} e $\det \mathbf{J}$ não são constantes, mas função das coordenadas naturais, a integração deve ser efetuada numericamente [39].

3.2.2.2 Integração numérica

O método de integração de \mathbf{k}_e utilizado é o *Método da Quadratura de Gauss*[44], embora haja muitos outros métodos numéricos para estimativa numérica de integrais definidos. Começando pelo caso unidimensional, suponha-se que é necessário avaliar o integral

$$I = \int_{-1}^1 f(\xi) d\xi. \quad (107)$$

A aproximação de Gauss, utilizando n pontos de integração, é dada por

$$I = \int_{-1}^1 f(\xi) d\xi \approx w_1 f(\xi_1) + w_2 f(\xi_2) + \dots + w_n f(\xi_n) \approx \sum_{i=1}^n w_i f(\xi_i), \quad (108)$$

onde $w_i (i=1 \dots n)$ representam os pesos de cada ponto de integração e $\xi_i (i=1 \dots n)$ os pontos de amostragem ou pontos de Gauss. O Método da Quadratura de Gauss consiste na seleção dos valores corretos de pesos de cada ponto de integração e localizações dos pontos de Gauss de forma a que a expressão (108) apresente uma solução exata para polinómios $f(\xi)$ do grau mais alto possível. Assim, infere-se que, se a expressão de integração de n pontos é exata para polinómios de um dado grau, também apresenta bons resultados para uma função que não seja polinomial. O método para definir os valores dos pesos e dos pontos de Gauss consiste na imposição de erro nulo na integração. Se for analisado o caso mais simples, de integração com um ponto ($n=1$), o Método de Gauss permite escrever

$$I = \int_{-1}^1 f(\xi) d\xi \approx w_1 f(\xi_1). \quad (109)$$

Como há duas constantes a determinar, w_1 e ξ_1 , considera-se que a aproximação apresenta um resultado exato para um polinómio de grau 1. Então, se $f(\xi) = a_0 + a_1 \xi$, o erro, que deve ser nulo, é dado por

$$\text{Erro} = \int_{-1}^1 (a_0 + a_1 \xi) d\xi - w_1 f(\xi_1) = 0. \quad (110)$$

Resolvendo o integral e substituindo $f(\xi_1)$ define-se o erro como

$$E = \left[a_0 \xi + \frac{a_1}{2} \xi^2 \right]_{-1}^1 - w_1 (a_0 + a_1 \xi_1) = 0 \Leftrightarrow E = 2a_0 - w_1 (a_0 + a_1 \xi_1) = 0. \quad (111)$$

Para se ter um erro nulo para quaisquer valores de a_0 e a_1 , deve-se fazer (Logan 2007)

$$\frac{\partial E}{\partial a_i} = 0, \quad (112)$$

para $i=n^\circ$ de variáveis a determinar (neste caso duas variáveis). Resulta assim para o presente caso

$$w_1 = 2 ; \xi_1 = 0, \quad (113)$$

o que significa que o valor de amostragem deve ser $f(0)$, com um peso de 2. Assim, a expressão (109) toma a forma

$$I = \int_{-1}^1 f(\xi) d\xi \approx 2f(0). \quad (114)$$

Generalizando para n pontos de integração, seguindo o mesmo procedimento, obtém-se os pesos e os pontos de integração para qualquer valor de n . A tabela seguinte apresenta estes valores até seis pontos de integração [47].

Número de pontos, n	Localização dos pontos, ξ	Pesos, w_i
1	0,0	2,0
2	$\pm 1/\sqrt{3} = \pm 0,5773502692$	1,0
3	$\pm 0,7745966692$	0,5555555556
	0,0	0,8888888889
4	$\pm 0,8611363116$	0,3478548451
	$\pm 0,3399810436$	0,6521451549
5	$\pm 0,9061798459$	0,2369268851
	$\pm 0,5384693101$	0,4786286705
	0,0	0,5688888889
6	$\pm 0,9324695142$	0,1713244924
	$\pm 0,6612093865$	0,3607615730
	$\pm 0,2386191861$	0,4679139346

Dos resultados apresentados constata-se que os pontos de Gauss se encontram localizados simetricamente relativamente a $\xi=0$, e que os pontos simétricos têm pesos idênticos. De uma maneira geral, o Método da Quadratura de Gauss com n pontos, apresenta solução exata se a função for polinomial de grau $(2n-1)$ ou inferior. Por outro lado, ao usar o Método de Gauss com n pontos, efetivamente substitui-se a função $f(\xi)$ por um polinómio de grau $(2n-1)$ e assim a precisão da integração numérica depende do grau de aproximação deste polinómio à função real a aproximar. Caso a função $f(\xi)$ não seja polinomial, o Método de Gauss é inexato, mas melhora a precisão com n .

Para integrais a duas dimensões do tipo

$$I = \int_{-1}^1 \int_{-1}^1 f(\xi, \eta) d\xi d\eta \quad (115)$$

pode ser aplicada a expressão (108) em sucessão, isto é, integrando primeiro relativamente a uma coordenada e posteriormente relativamente à outra. Assim

$$I \approx \int_{-1}^1 \left[\sum_{i=1}^n w_i f(\xi_i, \eta) \right] d\eta \quad (116)$$

$$I \approx \sum_{j=1}^n w_j \left[\sum_{i=1}^n w_i f(\xi_i, \eta_j) \right] \quad (117)$$

$$I \approx \sum_{i=1}^n \sum_{j=1}^n w_i w_j f(\xi_i, \eta_j). \quad (118)$$

De notar que não é necessário usar o mesmo número de pontos de Gauss em cada coordenada, isto é, pode-se ter $i \neq j$. O caso descrito pode ser aplicado à expressão de \mathbf{k}_e , apresentada em (106), onde $\mathbf{B} \det \mathbf{J}$ são funções das coordenadas naturais (ξ, η) . O método de determinação de \mathbf{k}_e consiste na integração numérica de uma matriz de ordem 8, elemento a elemento. No entanto, considerando a simetria desta matriz, o processo é simplificado, na medida em que não se torna necessário o cálculo dos elementos posicionados abaixo da diagonal principal. Considerando

$$f(\xi, \eta) = t_e (\mathbf{B}^T \mathbf{D} \mathbf{B} \det \mathbf{J})_{ij} \quad (119)$$

como sendo o elemento ij da matriz a integrar numericamente, a expressão (118) pode ser escrita da seguinte forma para o elemento ij

$$I_{ij} = k_{eij} \approx \sum_{i=1}^n \sum_{j=1}^n w_i w_j f(\xi_i, \eta_j). \quad (120)$$

Usando dois pontos de integração em cada dimensão obtém-se, desenvolvendo a expressão anterior

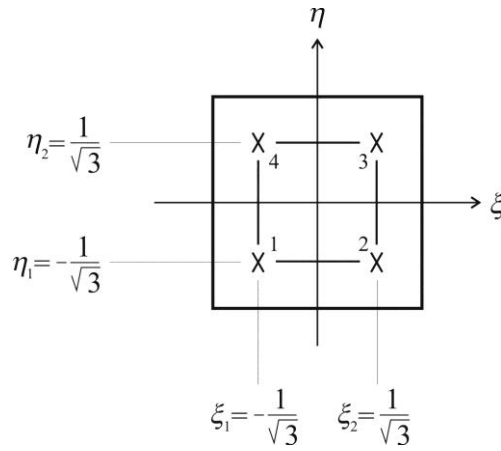
$$k_{eij} \approx \sum_{i=1}^2 \sum_{j=1}^2 w_i w_j f(\xi_i, \eta_j) \quad (121)$$

$$k_{eij} \approx \sum_{i=1}^2 [w_i w_1 f(\xi_i, \eta_1) + w_i w_2 f(\xi_i, \eta_2)] \quad (122)$$

$$k_{eij} \approx w_1^2 f(\xi_1, \eta_1) + w_1 w_2 f(\xi_1, \eta_2) + w_2 w_1 f(\xi_2, \eta_1) + w_2^2 f(\xi_2, \eta_2), \quad (123)$$

onde os pesos e as localizações dos pontos de Gauss são

$$w_1 = 1,0, \quad w_2 = 1,0, \quad \xi_1 = \eta_1 = -\frac{1}{\sqrt{3}}, \quad \xi_2 = \eta_2 = +\frac{1}{\sqrt{3}}. \quad (124)$$



3.2.2.3 Carregamentos equivalentes

Os carregamentos distribuídos e esforços por unidade de volume são tratados de forma idêntica ao elemento triangular de deformação constante de 3 nós, cuja descrição e sugestão de referências para maior detalhe se encontram na Secção 3.2.1.3.

3.2.2.4 Estado de tensão nos elementos

As deformações e tensões são definidas pelas expressões (24) e (14), respetivamente, considerando \mathbf{B} em coordenadas naturais (ξ, η) [38]. Tem-se assim que

$$\boldsymbol{\varepsilon}_e(\xi_i, \eta_j) = \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix}_{e(\xi_i, \eta_j)} = \mathbf{B}_e(\xi_i, \eta_j) \mathbf{a}_e \quad (125)$$

$$\boldsymbol{\sigma}_e(\xi_i, \eta_j) = \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix}_{e(\xi_i, \eta_j)} = \mathbf{D}_e \boldsymbol{\varepsilon}_e(\xi_i, \eta_j), \quad (126)$$

onde, para o cálculo de $\mathbf{B}(\xi_i, \eta_j)$, é necessária a substituição de ξ e η pelos valores relativos ao ponto de Gauss no qual estão a ser calculado os valores, ξ_i e η_j .

A partir da obtenção do estado de tensão, a expressão a utilizar para o cálculo da tensão equivalente de von Mises encontra-se em (77).

Da propriedade do material σ_{ced} e da tensão máxima pode ser obtido o F.S, calculado a partir da equação (78).

4 Desenvolvimento do *software*

Neste capítulo é descrito o programa desenvolvido e quais os recursos utilizados para tal, estabelecendo uma estrutura e descrevendo todas as funcionalidades do programa. Posteriormente apresentam-se os resultados obtidos pelo programa, assim como a validação do *software* desenvolvido para elementos bidimensionais triangulares e quadriláteros.

4.1 Estrutura

A criação do programa informático em ambiente Matlab[®] para estudo de estruturas planas pelo Método de Elementos Finitos seguiu a seguinte estrutura lógica:

- Inicialização do programa a partir do ficheiro StartGUI.fig ou a partir da linha de comandos do Matlab[®] escrevendo StartGUI.fig;
- Escolha da língua em que se quer trabalhar, clicando na respetiva bandeira. Os idiomas disponíveis nesta versão são o Português e o Inglês;
- Escolha do tipo de malha (triangular ou quadrangular) a analisar. Devido à diferença inerente de resolução de cada tipo de malha, cada análise tem um código distinto;
- Desenho da figura a analisar por coordenadas / escolha do quadrilátero a analisar;
- Introdução de carregamentos e restrições;

- Introdução das propriedades do material e da malha;
- Visualização de resultados sob o Método de Elementos Finitos.

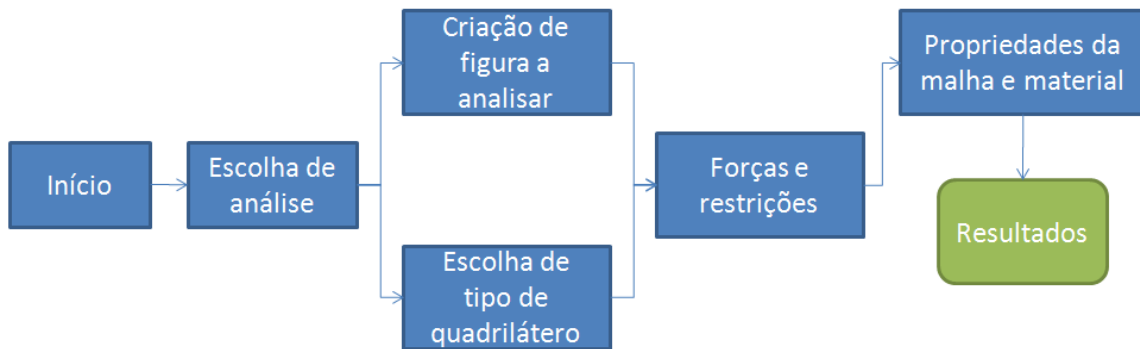


Figura 14: Esquema de funcionamento do *software* desenvolvido.

No esquema da Figura 14 observa-se como o *software* se desenvolve até à obtenção de resultados. De salientar que em qualquer passo deste processo se pode carregar um trabalho previamente guardado para continuar a análise.

4.2 Métodos de programação

O Matlab[®] tem um funcionamento programático similar às linguagens de programação C#, C++ e Java, pois muitas funções são utilizadas da mesma forma.

Utilizou-se o módulo GUIDE para auxílio da criação do ambiente gráfico. Esta ferramenta permite a fácil implementação de objetos de interface como botões, caixas de seleção, tabelas, entre outros a partir caixa de ferramentas deste módulo. Ao longo da construção da interface gráfica utilizando este módulo, o GUIDE gera automaticamente o código base de controlo em MATLAB[®], que pode ser alterado para programar o comportamento pretendido mais facilmente. Guardar um ficheiro de interface gráfica gera um ficheiro de extensão *.fig, que pode ser executado a partir do MATLAB[®]. O ficheiro tipo *.fig necessita um ficheiro *.m associado com o devido código programático.

O código escrito para a interface gráfica é guardado num ficheiro com extensão *.m. Aqui são guardadas todas as instruções e rotinas a serem executadas. Este ficheiro não necessita de um ficheiro *.fig associado pois o MATLAB® tem por si uma interface gráfica embutida. Contudo, a interface gráfica MATLAB® não pode ser livremente editada, enquanto o que se deseja é um ambiente personalizado.

O ambiente personalizado que foi criado necessita ambos ficheiros do tipo *.fig e *.m para ser executado.

Recorreu-se a algumas funções externas para a malhagem e assemblagem de matriz de rigidez local e global. Os restantes cálculos foram efetuados nos ficheiros *.m.

Dada a complexidade de criação da malha para elementos triangulares, recorreu-se ao *distmesh2d*, um código em MATLAB® que cria malha a partir de um conjunto específico de instruções. A explicação do funcionamento deste código distancia-se do âmbito desta dissertação, pelo que pode ser encontrado em [32].

Para malhagem de elementos quadrangulares, recorreu-se também a um código externo [48], enquanto para a resolução do sistema de equações utilizaram-se algumas funções encontradas em [49].

4.3 Bases de dados

O programa desenvolvido não recorre a bases de dados, exceptuando as do próprio MATLAB®. A estrutura pensada para o programa requer sempre introdução de dados pelo utilizador, no qual vão sendo criadas as variáveis necessárias ao longo do processo. Existe sim, a possibilidade de criação da própria base de dados do utilizador, já que o programa incorpora a função de guardar e carregar o trabalho feito.

4.4 Funcionamento do *software*

A partir da interface gráfica do programa, o utilizador terá que introduzir dados ao longo da execução do mesmo. Estes dados serão necessários para a implementação do MEF. Os dados necessários são os seguintes:

- Um domínio fechado (figura criada na janela Make2d_PT ou escolha de quadrilátero em Q4_PT);
- Restrições e ações (opções na janela ForceNreact_PT);
- Propriedades do material e propriedades da malha (introduzidos na janela VALUEGUI_PT).

O núcleo do código deste programa em MATLAB[®] encontra-se no ficheiro VALUEGUI_PT.m. Aqui criam-se as matrizes **p** e **t**, que fazem parte do código pacote *distmesh2d* no caso de malhagem com elementos triangulares. A matriz **p** contém a localização cartesiana dos nós dos elementos. A matriz **t** contém a identificação dos nós pertencentes a um dado elemento. No caso de elementos quadrangulares, os nós e posição dos elementos são criados a partir de uma outra metodologia, que adquire na mesma a forma de matrizes **p** e **t** por conveniência.

Reunindo-se estes valores inicia-se a análise pelo MEF – aproximação formal, no caso de malha de elementos triangulares e análise pelo MEF – aproximação isoparamétrica, no caso de malha de elementos quadrangulares (ver capítulo 3). São geradas matrizes vazias com as dimensões necessárias para albergar as matrizes de rigidez local [**k**] e global [**K**]. É gerada a matriz de elasticidade [**D**], dependendo se a análise é por EPT ou EPD, comum a todos os elementos, pois todos os elementos partilham da mesma propriedade de material. São também inicializados os vetores de deslocamentos e forças, assim como as matrizes de tensões e deformações. Isto é feito para permitir uma melhor fluidez do programa, pois todas as variáveis que permitem criar estas matrizes existem e o MATLAB[®] tem um funcionamento mais eficiente com matrizes pré-criadas do que com matrizes criadas incrementalmente[49].

Segue-se a análise elemento a elemento, onde se compila a posição de cada nó de cada elemento com os graus de liberdade do sistema utilizando a função externa *feeldof*.

Posteriormente calcula-se a área do triângulo e as funções de forma para obedecer à equação (52) para definir a matriz de deformação $[B]$. Através dos dados obtidos é então assemblada a matriz de rigidez do elemento singular a partir da equação (53), que é incrementada na matriz de rigidez global. Estes passos são repetidos até ser analisado o último elemento, onde o ciclo de análise elemento a elemento é concluído.

São depois aplicadas as restrições no sistema global de equações, que são identificadas pelos nós na face que estiver restrita pelo utilizador. Calculam-se os deslocamentos do sistema a partir da equação $u = k^{-1}(a, a)ff(a)$, onde u é o vetor de deslocamentos, k a matriz de rigidez global, ff o vetor de forças e a os nós na fronteira.

É então feita uma pós-computação para retirar as matrizes de tensão e deformação, permitindo assim o utilizador visualizar e conferir estes valores.

Como nota final, o programa não tem um ficheiro executável, pelo que deve ser inicializado a partir do programa Matlab[®].

4.4.1 Arranque do programa

O programa de malhagem e análise pelo MEF é iniciado a partir do ficheiro *StartGUI.fig*, fazendo duplo clique sobre este ficheiro ou digitando “*StartGUP*” na linha de comando do MATLAB[®]. Nesta janela o utilizador inicializará o programa escolhendo o idioma pretendido. A partir deste ponto é apresentado o programa em língua Portuguesa.

4.4.2 Escolha do tipo de malhagem

Após a escolha do idioma pretendido, o utilizador escolhe qual o tipo de malha que quer utilizar, tendo como opções Malhagem Triangular e Malhagem Quadrangular, como mostra a Figura 15. Caso escolha a Malhagem Triangular, é reencaminhado para a secção de construção, janela *Make2dGUI_PT*, onde o utilizador poderá construir a figura que pretende analisar. Caso escolha a Malhagem Quadrangular, o utilizador é reencaminhado para a janela *Q4_PT*, onde poderá escolher analisar entre três quadriláteros arbitrados, cada um com a sua estrutura de malha. Se pretender, pode nesta janela carregar um trabalho

previamente guardado. Poderá ainda ler os créditos com informação do programa e do autor da presente tese.

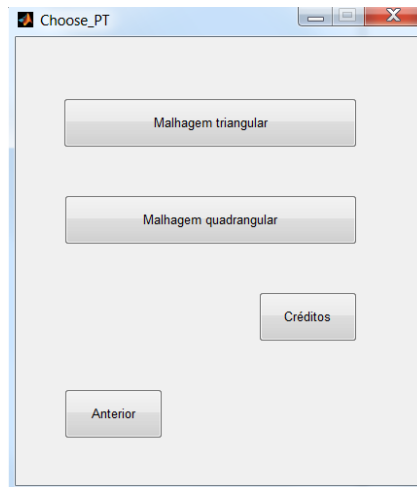


Figura 15: Escolha do tipo de malha.

Na próxima secção é onde é criado o domínio para análise (ver capítulo 2.1.2), um dos pré-requisitos para aplicação do MEF.

4.4.2.1 Criação geometria

Após a selecção do tipo de malhagem, caso tenha sido malhagem por elementos triangulares, é aberta a janela *Make2dGUI_PT* representada na Figura 16. Nesta janela poderá criar a sua figura a partir das caixas de coordenadas.

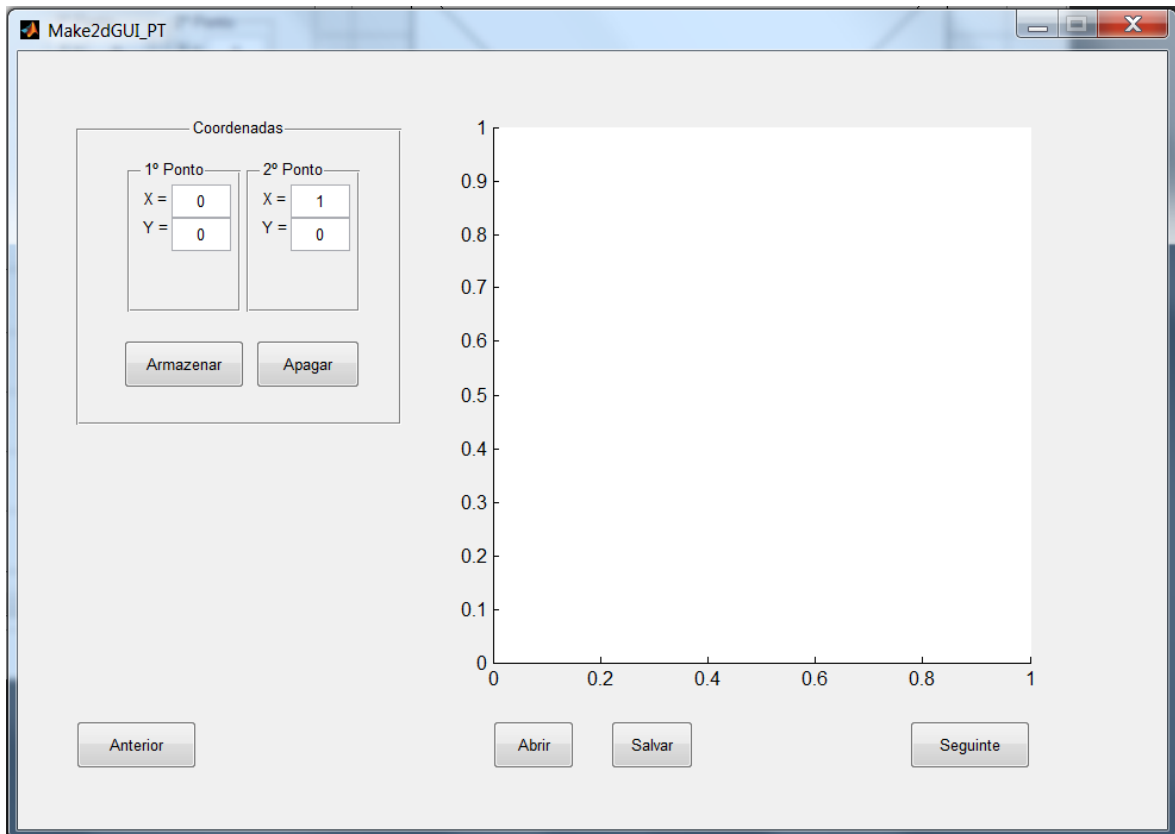


Figura 16: Janela de criação de geometria.

Segundo a abordagem informática utilizada, a maneira correta de utilizar desenhos a sua estrutura é de começar num ponto à sua escolha, desenhar a sua figura a partir desse ponto continuamente recorrendo ao botão “Armazenar”, e acabar no mesmo ponto em que começou, onde o programa pode então assumir que o utilizador terminou o desenho da figura a analisar. Caso necessário pode-se recorrer ao botão “Apagar” para apagar o último par de coordenadas introduzidas. Para melhor clareza, a partir do ponto inicial, a primeira secção de introdução de coordenadas ficará inativa, onde se transformará num lembrete da última coordenada introduzida, conforme se pode verificar na Figura 17.

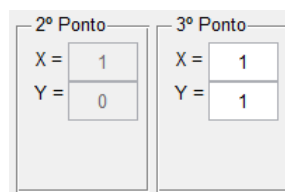


Figura 17: Demonstração da introdução de coordenadas.

A figura criada deverá ser então iniciada no ponto pretendido, no qual se continuará deste ponto em diante, como é demonstrado na Figura 18.

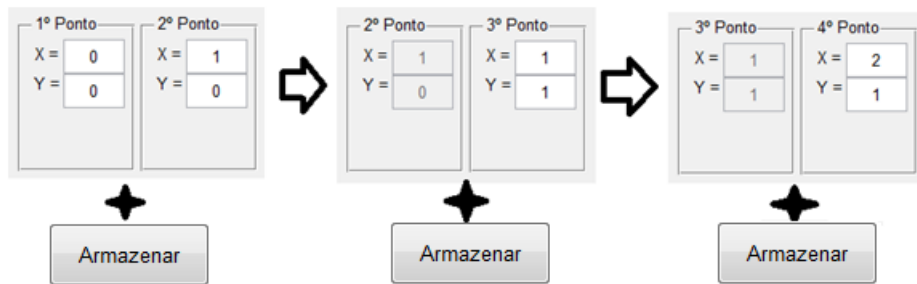


Figura 18: Introdução de coordenadas.

O resultado obtido a partir da introdução de dados segundo a Figura 18 pode ser verificado pela Figura 19.

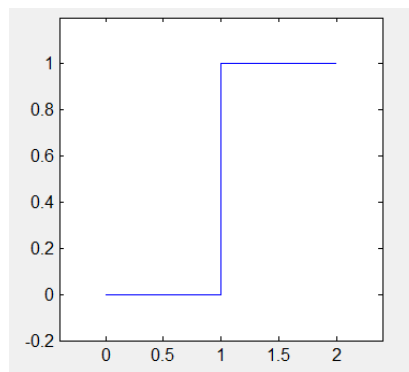


Figura 19: Resultado visual da introdução de coordenadas.

O utilizador poderá retroceder na criação de figura caso o ponto inserido não seja o desejado clicando no botão “Apagar”. Nesta janela poderá também guardar o seu trabalho clicando no botão “Salvar” ou carregar um trabalho previamente feito a partir do botão “Abrir”. Se o utilizador preferir recomeçar a criação da figura, pode fazê-lo carregando no botão “Apagar” até desaparecerem todos os pontos previamente desenhados, ou carregando no botão “Anterior”. Ao carregar no botão “Anterior” o utilizador será questionado se quer realmente sair, pois qualquer trabalho realizado até este ponto será apagado se não for previamente gravado através do botão “Salvar”. Isto pode ser observado na Figura 20.

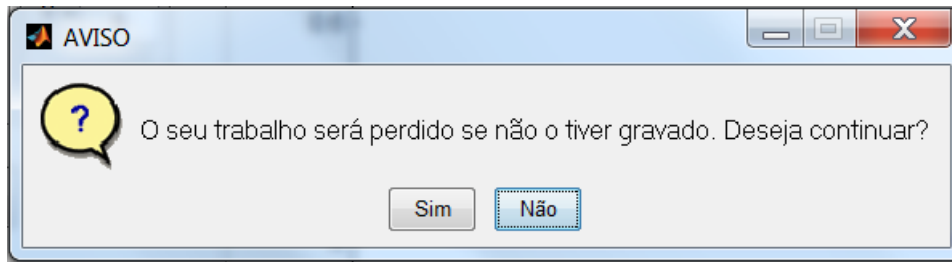


Figura 20: Aviso quando pressionar o botão “Anterior”.

Quando o utilizador finalizar a sua figura e criar um domínio fechado, poderá continuar para a próxima secção carregando no botão “Seguinte”, ao que será aberta a janela *ForceNreact_PT*.

4.4.2.2 Escolha da geometria

Após a selecção do tipo de malhagem, caso tenha sido malhagem por elementos quadrangulares, é aberta a janela *Q4_PT* representada na Figura 21. Nesta janela o utilizador poderá escolher o tipo de geometria que pretende analisar a partir das hipóteses apresentadas.

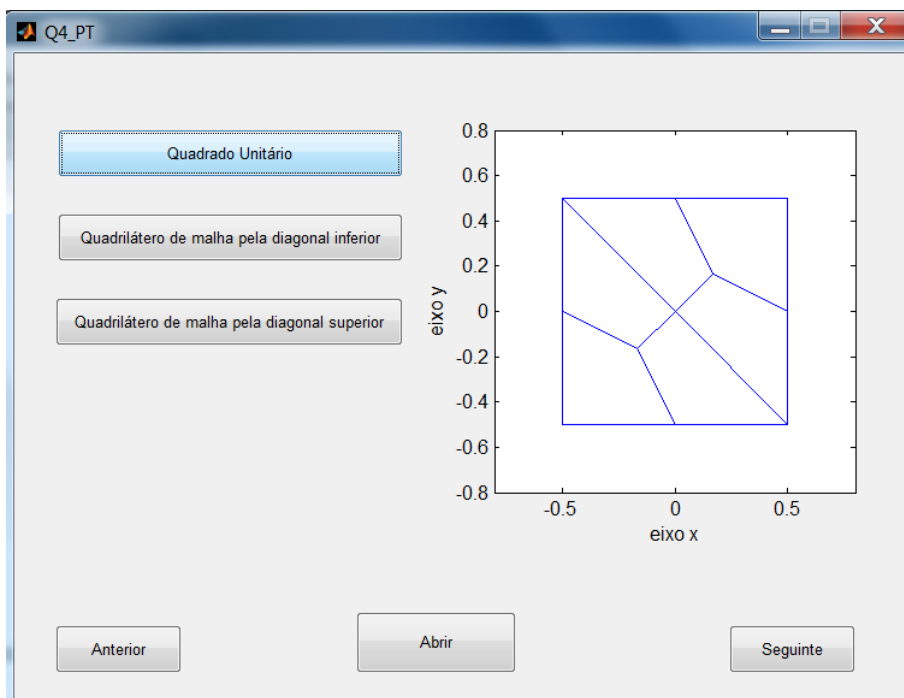
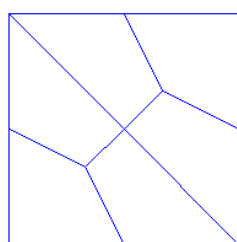


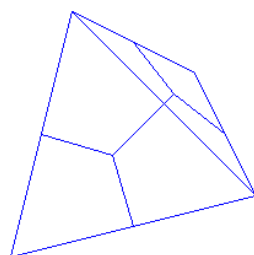
Figura 21: Janela de escolha de geometria.

As escolhas são limitadas a três casos:

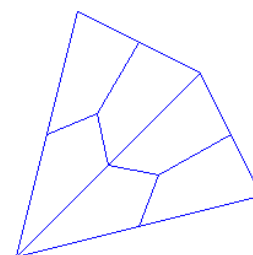
- Quadrilátero unitário, com centróide na origem do eixo (Figura 22-a);
- Quadrilátero arbitrado com malha pela diagonal inferior (Figura 22-b);
- Quadrilátero arbitrado com malha pela diagonal superior (Figura 22-c);



a) Quadrilátero Unitário



b) Quadrilátero arbitrário com malha pela diagonal inferior



c) Quadrilátero arbitrário com malha pela diagonal superior

Figura 22: Quadriláteros disponíveis para análise.

Ao clicar em qualquer um dos três botões à esquerda da área de pré-visualização, o utilizador escolherá o tipo de quadrilátero que quer analisar, tendo também uma pré-visualização disponível do mesmo.

O código de malhagem para elementos quadriláteros utiliza diagonais para subdividir as figuras em regiões triangulares, posteriormente utilizando diagonais intermédias para fazer outra subdivisão. Tendo isto em conta, o código tem a limitação de produzir malhas unicamente para figuras com vértices pares, e neste caso específico, quadriláteros.

Caso o utilizador pretenda carregar um trabalho previamente gravado, carrega no botão “Abrir” e será redireccionado para a próxima janela de trabalho, já com o seu trabalho anterior carregado.

Finalizada a escolha, o utilizador clica no botão “Seguinte”, onde será fechada esta janela e aberta a próxima, *ForceNreact_PT*.

4.4.3 Condições fronteira e carregamentos

É nesta janela que podem ser criadas as condições fronteira e carregamentos pontuais. A figura feita anteriormente é carregada no lado esquerdo da janela. No lado direito existem duas secções: restrições e forças.

Para introduzir uma restrição numa linha, o utilizador terá de seleccionar a mesma a partir duma lista disponível numa caixa *dropdown*, onde, após a selecção, deverá ser escolhido o tipo de restrição pretendido, ou seja, uma restrição de deslocamento no eixo X, no eixo Y ou em ambos eixos, conforme demonstrado na Figura 23.

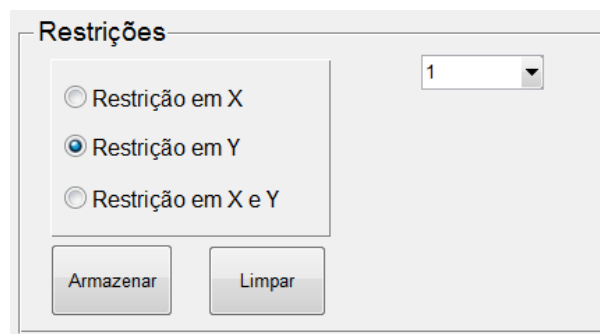


Figura 23: Atribuição de restrições nas linhas de fronteira.

Terminada a selecção, é armazenada a decisão em vetores independentes a partir do botão “Armazenar”. O tipo de restrição é representado na figura sob espessura de linha mais grossa, com o tipo de linha sendo tracejado, pontilhado, ou traço e ponto para restrição em X, em Y e em X e Y respetivamente, como pode ser visto na Figura 24.

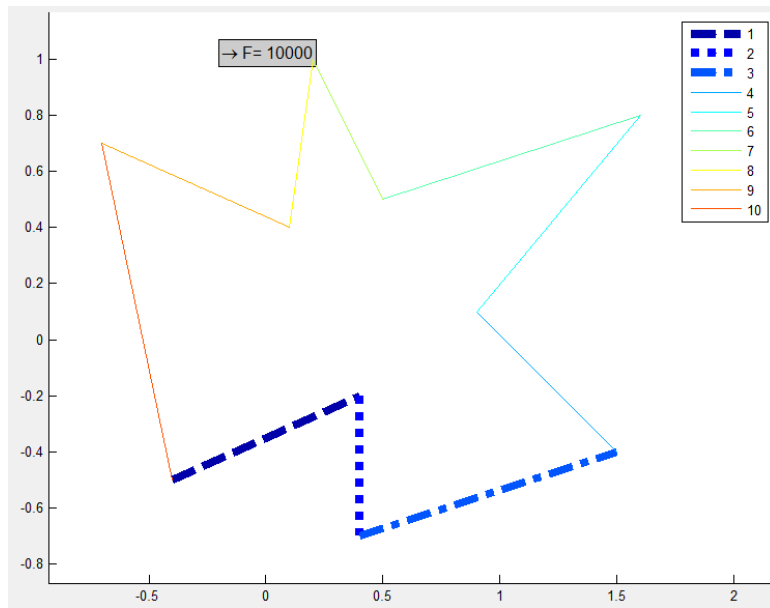


Figura 24: Representação dos diferentes tipos de restrição.

A seleção de uma restrição diferente à escolhida vai prevalecer sobre a anterior aquando o armazenamento da decisão pelo botão “Armazenar”. Ao introduzir o mesmo tipo de restrição numa linha já restrita, surgirá uma caixa de texto a avisar que a linha já tem a restrição que o utilizador pretende. O utilizador pode valer-se do botão “Limpar” para eliminar todas as restrições introduzidas.

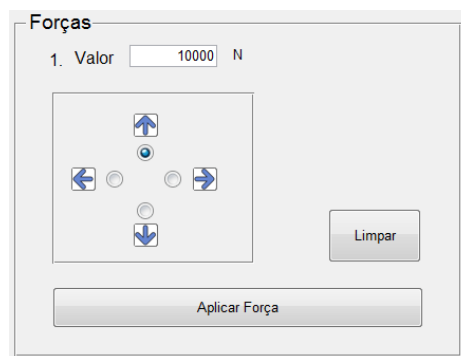


Figura 25: Painel de seleção da direção da força a aplicar.

Para introduzir uma força, escolhe-se a sua direção e intensidade a partir do painel de botões com setas e da caixa de texto com descrição de força, respetivamente(Figura 25). Após a caracterização da força, o utilizador deverá clicar no botão “Aplicar Força” e seleccionar o ponto na figura onde quer aplicar a força. A correta selecção deste ponto é

caracterizada por um marcador e uma caixa de texto ao lado deste que representa as suas coordenadas (Figura 26). Estando este ponto seleccionado, o utilizador deverá clicar no botão “Ponto Escolhido”, que anteriormente era o botão “Aplicar Força”, com o intuito de armazenar a direção, intensidade e localização da força. Este armazenamento é demonstrado visualmente por uma caixa de texto de cor cinzenta que surge no local de aplicação da força com a respetiva direção e intensidade.

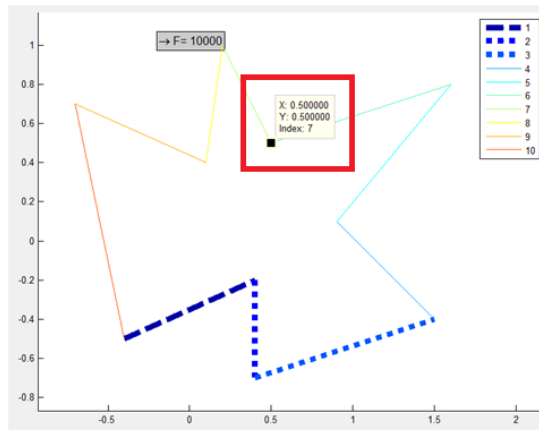


Figura 26: Aplicação de força num ponto da área de visualização.

Quando o utilizador tiver terminado a caracterização de restrições e forças na area de visualização, pode então seguir para a próxima janela, *VALUEGUI_PT*, clicando no botão “Seguinte”. Note-se que o utilizador pode partir para a janela seguinte sem ter definido restrições ou forças, pois a malhagem da figura é independente destes valores. Contudo, não serão representados quaisquer valores aquando da representação de resultados.

4.4.4 Malhagem e propriedades do material

É na janela *VALUEGUI_PT* onde é aplicada a análise numérica e o pós-processamento. Aqui é representada a figura juntamente com as forças e reações aplicadas criadas pelo utilizador. Ao lado esquerdo estão dois painéis: Propriedades da Malha e Propriedades do Material. Sobre a figura existe um conjunto de botões com os símbolos σ e ϵ , representativos de tensão e deformação.

Representado na Figura 27, estão painel Propriedades da Malha.

Para o caso de análise com elementos triangulares, o utilizador introduz qual o tamanho médio do elemento finito de que a malha vai ser constituída.

Para o caso de análise com elementos quadrangulares, é introduzido na caixa de texto o número de subdivisões da malha. Quanto mais elevado este número, maior é o número de elementos gerados, e conseqüentemente, refinamento da malha é mais elevado.

Neste painel está também a opção de escolher o tipo de estado de tensão, EPT ou EPD. Qualquer uma destas escolhas influenciará o resultado final.

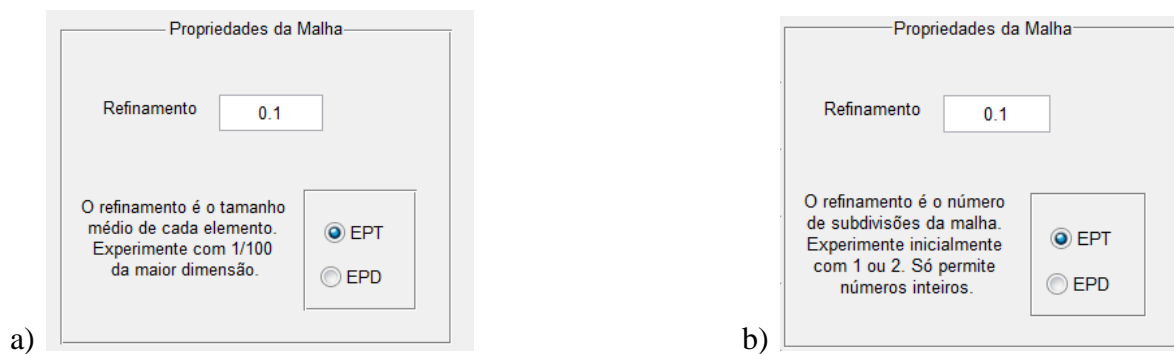


Figura 27 a) Propriedades da malha para análise com elementos triangulares e b) Propriedades da malha para análise com elementos quadrangulares.

No painel Propriedades do Material introduzem-se as propriedades necessárias para caracterizar a figura, como pode ser verificado na Figura 28. Estas propriedades são a espessura da placa fina (ver capítulo 3) em milímetros, módulo de Young (E) em Pascal, coeficiente de Poisson (ν) e tensão de cedência, que serão utilizadas para gerar os resultados.

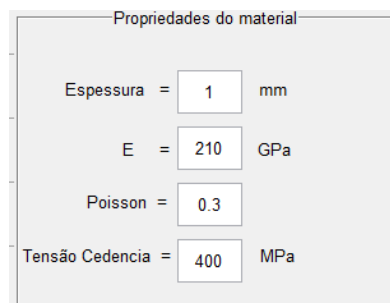


Figura 28: Painel de Propriedades do Material.

Após o utilizador escolher as propriedades da malha e do material pretendidas, clica no botão “Malhar” para executar o código de malhagem e respetiva análise MEF.

4.4.5 Apresentação de resultados

Finalizado o processo de malhagem e cálculo MEF, é apresentada por defeito a tensão de von Mises em cada elemento na área de visualização, como pode ser verificado na Figura 29. É possível visualizar diferentes resultados se o utilizador recorrer aos botões no topo da area de visualização, cada um indicando o tipo de resultados possíveis de apresentar. Estes resultados incluem as tensões nos eixos X e Y, tensão de cisalhamento e tensão equivalente, deformação nos eixos X e Y e distorção no plano XY.

Torna-se visível um terceiro painel, que apresenta a tensão máxima de von Mises, assim como o fator de segurança, ambos obtidos após a computação da malha e resultados.

Na resolução do modelo com elementos quadrangulares, utilizou-se dois pontos na quadratura de Gauss.

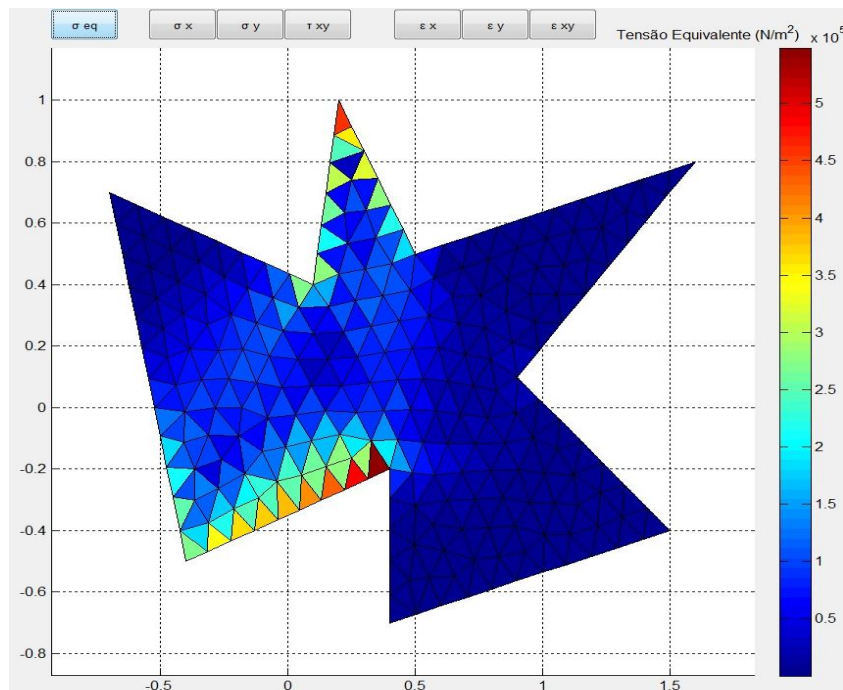
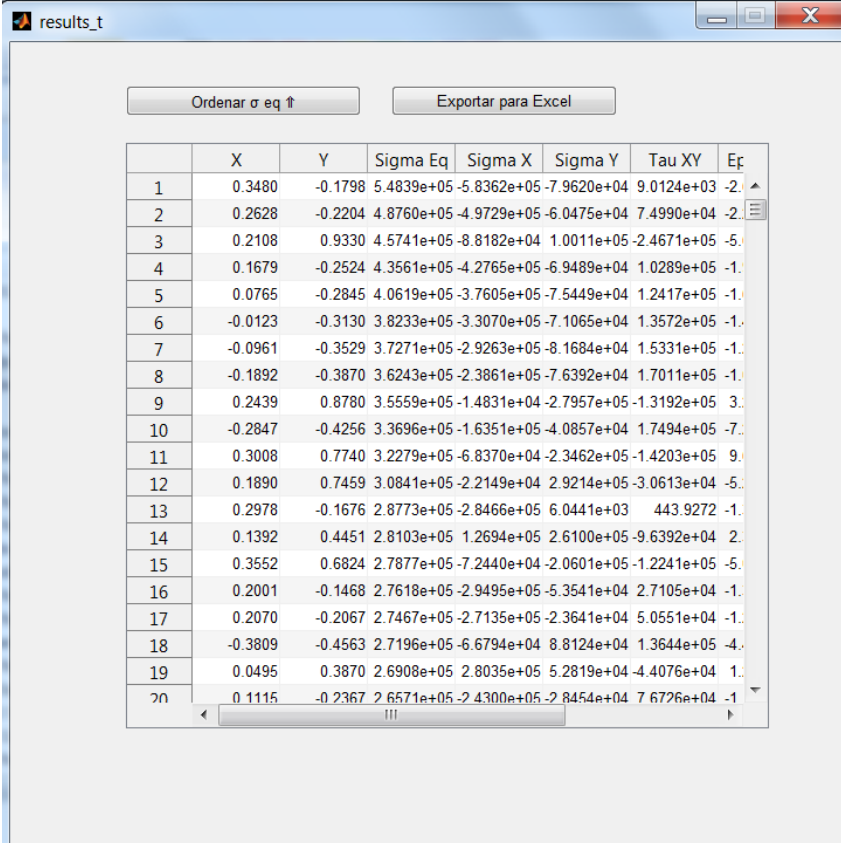


Figura 29: Área de visualização com resultados pós-computação.

É agora possível interagir com o botão “Resultados”. Clicando em “Resultados” abre-se uma janela independente, Figura 30, para poder visualizar os dados em forma numérica, podendo exportá-los para um ficheiro Excel através do botão “Exportar para Excel”. O utilizador pode organizar os dados por ordem ascendente ou descendente a partir do botão “Ordenar σ_{eq} ↑” como se pode observar na figura. A ordenação de dados é efetuada pela coluna Sigma eq, referente à tensão de von Mises.



	X	Y	Sigma Eq	Sigma X	Sigma Y	Tau XY	E _f
1	0.3480	-0.1798	5.4839e+05	-5.8362e+05	-7.9620e+04	9.0124e+03	-2.
2	0.2628	-0.2204	4.8760e+05	-4.9729e+05	-6.0475e+04	7.4990e+04	-2.
3	0.2108	0.9330	4.5741e+05	-8.8182e+04	1.0011e+05	-2.4671e+05	-5.
4	0.1679	-0.2524	4.3561e+05	-4.2765e+05	-6.9489e+04	1.0289e+05	-1.
5	0.0765	-0.2845	4.0619e+05	-3.7605e+05	-7.5449e+04	1.2417e+05	-1.
6	-0.0123	-0.3130	3.8233e+05	-3.3070e+05	-7.1065e+04	1.3572e+05	-1.
7	-0.0961	-0.3529	3.7271e+05	-2.9263e+05	-8.1684e+04	1.5331e+05	-1.
8	-0.1892	-0.3870	3.6243e+05	-2.3861e+05	-7.6392e+04	1.7011e+05	-1.
9	0.2439	0.8780	3.5559e+05	-1.4831e+04	-2.7957e+05	-1.3192e+05	3.
10	-0.2847	-0.4256	3.3696e+05	-1.6351e+05	-4.0857e+04	1.7494e+05	-7.
11	0.3008	0.7740	3.2279e+05	-6.8370e+04	-2.3462e+05	-1.4203e+05	9.
12	0.1890	0.7459	3.0841e+05	-2.2149e+04	2.9214e+05	-3.0613e+04	-5.
13	0.2978	-0.1676	2.8773e+05	-2.8466e+05	6.0441e+03	443.9272	-1.
14	0.1392	0.4451	2.8103e+05	1.2694e+05	2.6100e+05	-9.6392e+04	2.
15	0.3552	0.6824	2.7877e+05	-7.2440e+04	-2.0601e+05	-1.2241e+05	-5.
16	0.2001	-0.1468	2.7618e+05	-2.9495e+05	-5.3541e+04	2.7105e+04	-1.
17	0.2070	-0.2067	2.7467e+05	-2.7135e+05	-2.3641e+04	5.0551e+04	-1.
18	-0.3809	-0.4563	2.7196e+05	-6.6794e+04	8.8124e+04	1.3644e+05	-4.
19	0.0495	0.3870	2.6908e+05	2.8035e+05	5.2819e+04	-4.4076e+04	1.
20	0.1115	-0.2367	2.6571e+05	-2.4300e+05	-2.8454e+04	7.6726e+04	-1.

Figura 30: Janela de resultados da análise.

4.5 Validação do software desenvolvido

Um dos meios possíveis para se validar os resultados é utilizar um modelo simples para resolução algébrica e comparação. Outro meio é a utilização de um software comercial com resolução MEF. Nesta tese foi utilizado o programa ABAQUS® para a validação de resultados.

4.5.1 Elementos bidimensionais triangulares

A apresentação visual dos resultados de ambos programas para resolução do MEF para elementos bidimensionais triangulares pode ser encontrada na Figura 32. Por conveniência, as figuras de todos os resultados obtidos encontram-se no anexo B. Na Tabela 1 encontra-se uma comparação da tensão equivalente de von Mises obtida por número de elementos entre o *software* desenvolvido e o ABAQUS®. A tensão equivalente de von Mises para o ensaio no ABAQUS® foi obtida a partir da média de valores encontrados nos três nós do elemento com maior tensão, já que não foi possível requerir ao ABAQUS® que representasse um valor único por elemento.

Verifica-se uma boa correlação entre os resultados, com uma diferença média de 9%, o que valida o *software* para resolução de MEF para elementos bidimensionais triangulares. Esta correlação pode ser visualizada na Figura 31. Possíveis causas para esta diferença poderão ser a diferença entre a organização dos nós nas malhas, resultando em valores ligeiramente distintos, e uma diferente formulação do elemento finito em questão.

Tabela 1: Comparação de resultados entre o *software* desenvolvido e o ABAQUS®.

MATLAB		ABAQUS		
Nº Elementos	σ von Mises	Nº Elementos	σ von Mises	Diferença
384	4,57E+05	367	3,86E+05	-16%
595	5,45E+05	602	4,89E+05	-10%
1089	6,12E+05	1042	6,29E+05	3%
2551	1,14E+06	2582	1,05E+06	-8%
10554	2,30E+06	10008	2,04E+06	-12%
			Média	-9%

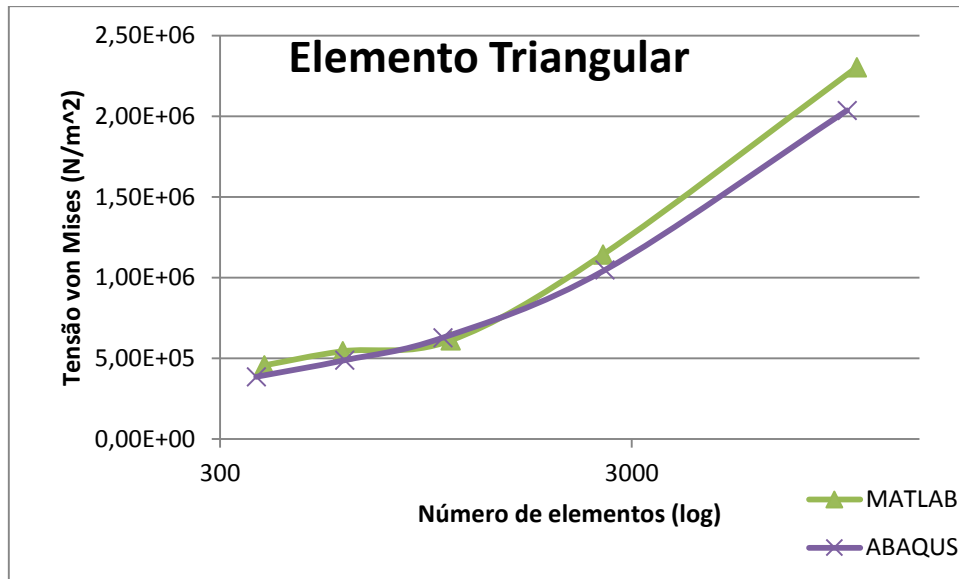


Figura 31: Correlação entre resultados obtidos em MATLAB® e ABAQUS®.

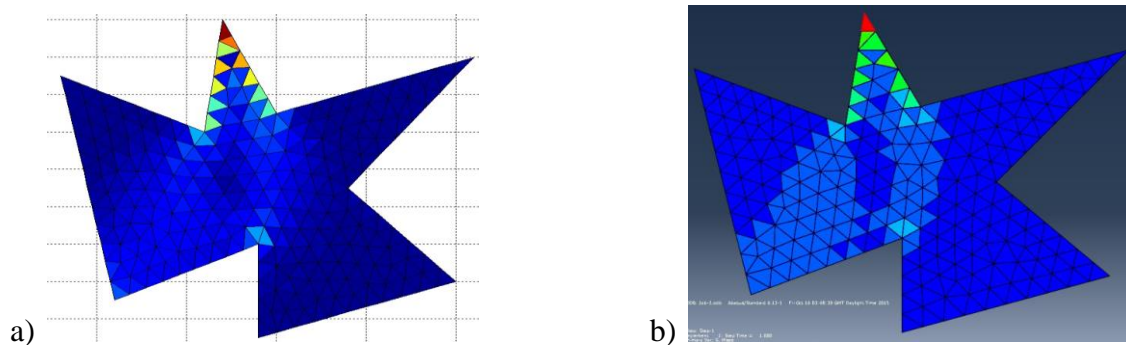


Figura 32 a) Figura com 384 elementos em MATLAB® e b) Figura com 367 elementos em ABAQUS®.

4.5.2 Elementos bidimensionais quadriláteros

A apresentação visual dos resultados de ambos programas para resolução do MEF para elementos bidimensionais quadriláteros pode ser encontrada na Figura 34. Por conveniência, as figuras de todos os resultados obtidos encontram-se no anexo B. Na Tabela 2 encontra-se uma comparação da tensão equivalente de von Mises obtida por número de elementos entre o *software* desenvolvido e o ABAQUS®. A tensão equivalente de von Mises para o ensaio no ABAQUS® foi obtida a partir da média de valores

encontrados nos quatro nós do elemento com maior tensão, já que não foi possível requerir ao ABAQUS® que representasse um valor único por elemento.

Tabela 2: Comparação de resultados entre o software desenvolvido e o ABAQUS®.

MATLAB		ABAQUS		Diferença
Nº Elementos	σ von Mises	Nº Elementos	σ von Mises	
54	3,82E+04	48	4,17E+04	9%
96	5,09E+04	93	5,24E+04	3%
216	7,63E+04	225	8,87E+04	16%
384	1,02E+05	341	9,66E+04	-5%
486	1,15E+05	484	1,42E+05	24%
600	1,27E+05	644	1,39E+05	-8%
			Média	9%

Verifica-se uma boa correlação entre os resultados, com uma diferença média de 9%, o que valida o software para resolução de MEF para elementos bidimensionais quadrangulares. Esta correlação pode ser visualizada na Figura 31. Possíveis causas para esta diferença poderão ser a diferença entre a organização dos nós nas malhas, resultando em valores ligeiramente distintos, e uma possível diferente formulação do elemento finito em questão.

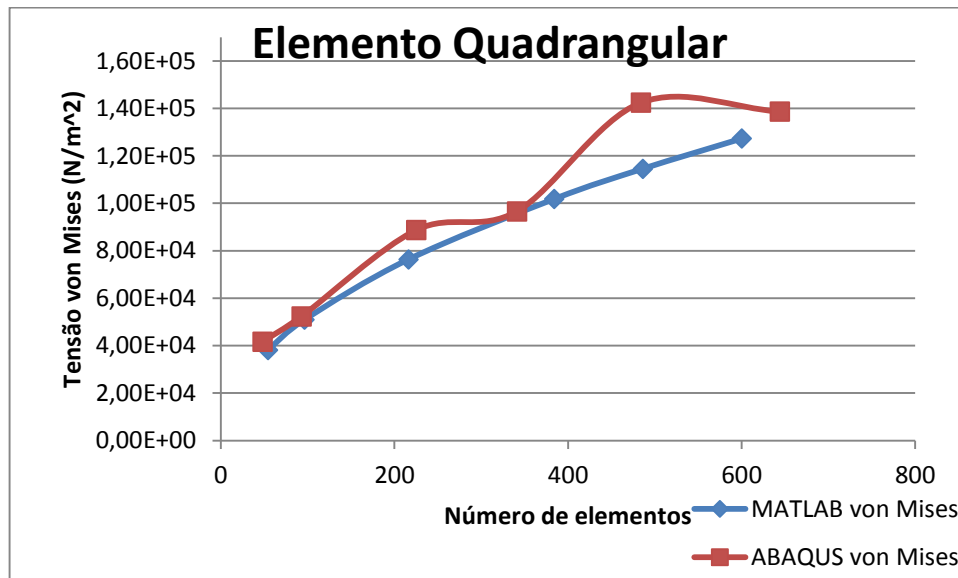


Figura 33: Correlação entre resultados obtidos em MATLAB® e ABAQUS®.

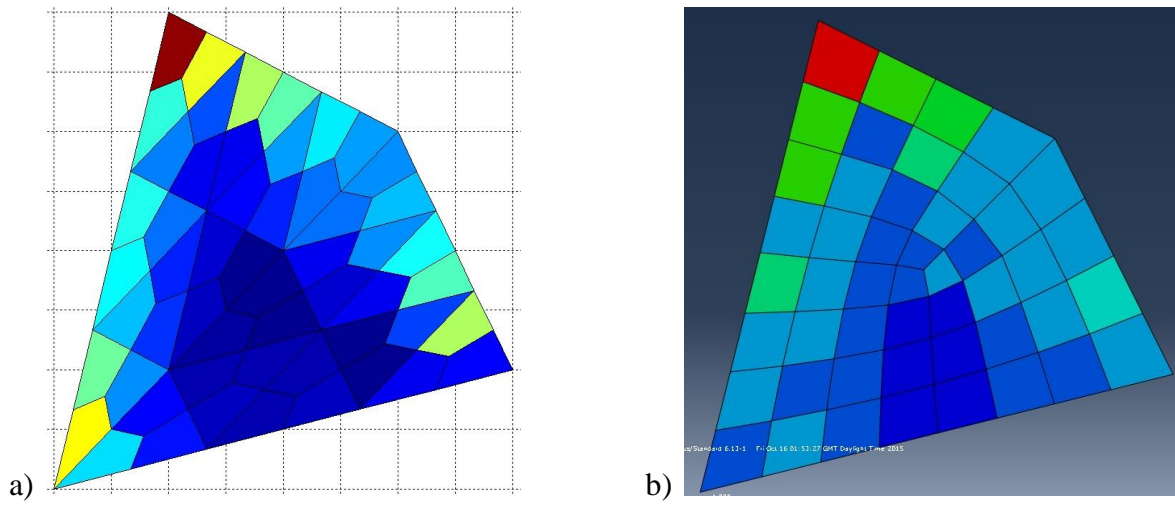


Figura 34 a) Figura com 54 elementos em MATLAB® e b) Figura com 48 elementos em ABAQUS®.

5 Conclusões

Neste trabalho pretendeu-se o desenvolvimento de um *software* na plataforma de programação MATLAB[®] que permita a modelação de um modelo arbitrado, podendo introduzir os carregamentos e restrições, e posteriormente visualizar os resultados em ambiente gráfico ou tabular. Implementou-se um código de geração de malha para ambos casos de malhagem com elementos bidimensionais triangulares e elementos bidimensionais quadrangulares. Para a resolução dos sistemas de equações do MEF, utilizou-se a aproximação formal para elementos bidimensionais triangulares, e para elementos bidimensionais quadrangulares utilizou-se a aproximação isoparamétrica, permitindo ambas aproximações, EPT ou EPD. Verificaram-se os resultados obtidos pelo *software* desenvolvido com o *software* ABAQUS[®], comprovando-se uma boa correlação entre eles. O *software* permite a gravação e carregamento de sessão, como também permite a exportação dos resultados em formato *.xls. Este demonstra o fator de segurança obtido a partir da tensão máxima equivalente. É possível também a escolha linguística do programa.

O *software* desenvolvido cumpre assim todos os objetivos a que se propôs, incluindo os seus sub-objetivos.

Sugestões para trabalhos futuros:

- Análise do código para reduzir tempo de processamento;
- Refinamento do código anti-colisão na plataforma de criação Make2dGUI;
- Refinamento da malha em pontos específicos. Sugere-se o uso da função *fh* do *distmesh* para elementos triangulares;
- Capacidade de importação de ficheiros do tipo CAD;
- Criação de bibliotecas com geometrias pré-definidas diversas;
- Melhoramento do código para análise de outro tipo de geometria.

6 Bibliografia

- [1] A. J. M. Ferreira, *Método dos Elementos Finitos*, Porto: Faculdade de Engenharia da Universidade do Porto, 2003.
- [2] J. M. R. S. Tavares, *Introdução ao Método dos Elementos Finitos*, Porto: Faculdade de Engenharia da Universidade do Porto, 1998.
- [3] E. B. d. L. Cascas, *O Método dos Elementos Finitos*, 2000.
- [4] D. L. Logan, *A first Course in the Finite Element Method*, Toronto: Thomson, 2007.
- [5] H. L. S. e. S. S. Lima, *Método dos Elementos Finitos em Análise de Estruturas*, São Paulo: Editora da Universidade de São Paulo, 2003.
- [6] A. F. M. Azevedo, *Método dos Elementos Finitos*, Faculdade de Engenharia da Universidade do Porto, 2003.
- [7] J. A. T. d. F. e. C. Tiago, “Análise Elástica de Estruturas Reticuladas,” Instituto Superior Técnico, Lisboa, 2015.
- [8] N. M. V. d. Carvalho, “Cálculo automático de Estruturas. Análise estrutural de pórticos planos utilizando o método dos elementos finitos.,” Instituto Superior de Engenharia de Lisboa, 2010.
- [9] Y. Liu, *Lecture notes: Introduction to the Finite Element Method*, Cincinnati: University of Cincinnati, 1997-2003.

- [10] V. Franco, *Fundamentos do Método dos Elementos Finitos*, Oeiras, 2011.
- [11] O. J. B. A. Pereira, *Introdução ao Método dos Elementos Finitos na análise de problemas planos de elasticidade.*, Lisboa: Instituto Superior Técnico, 2005.
- [12] R. D. S. G. Campilho, *Método de Elementos Finitos*, Portugal:: Publindústria, 2012.
- [13] C. M. A. Calviti, “Estudo do processo de desenvolvimento de motores de combustão interna,” 2008.
- [14] S. S. Rao, *The Finite Element Method in Engineering*, Elsevier, 2011.
- [15] R. J. C. M. Figueira, “CAD/CAE/CAM/CIM,” 2003.
- [16] M. Mahendran, “Applications of Finite Elements Analysis in Structural Engineering,” *Proceedings International Conference on Computer Aided Engineering*, 2007.
- [17] Setembro 2015. [Online]. Available: http://ec.europa.eu/research/transport/projects/items/hercules_en.htm. [Acedido em September 2015].
- [18] Setembro 2015. [Online]. Available: <http://www.pv-calc.com/>. [Acedido em September 2015].
- [19] Setembro 2015. [Online]. Available: http://processmodeling.org/model_em/elmag1.html. [Acedido em September 2015].
- [20] Setembro 2015. [Online]. Available: <http://sites.poli.usp.br/d/pmr5222/MEFtutorial.pdf>. [Acedido em September 2015].
- [21] Setembro 2015. [Online]. Available: <https://en.wikipedia.org/wiki/Nastran>. [Acedido em 2015 September].
- [22] Setembro 2015. [Online]. Available: <http://www.mscsoftware.com/product/msc-nastran>. [Acedido em September 2015].

- [23] Setembro 2015. [Online]. Available: <https://en.wikipedia.org/wiki/Ansys>. [Acedido em September 2015].
- [24] Setembro 2015. [Online]. Available: <http://www.ansys.com/>. [Acedido em September 2015].
- [25] Setembro 2015. [Online]. Available: <http://imechanica.org/files/16-adaptive-mesh.pdf>. [Acedido em October 2015].
- [26] October 2015. [Online]. Available: http://tesis.com.ru/infocenter/downloads/abaqus/abaqus_newsv613.pdf. [Acedido em October 2015].
- [27] Setembro 2015. [Online]. Available: <http://www.3ds.com/products-services/simulia/products/abaqus/latest-release/>. [Acedido em October 2015].
- [28] Setembro 2015. [Online]. Available: <https://en.wikipedia.org/wiki/ADINA>. [Acedido em October 2015].
- [29] Setembro 2015. [Online]. Available: <http://www.adina.com/adinaaui.shtml>. [Acedido em October 2015].
- [30] Outubro 2015. [Online]. Available: <http://www.3ds.com/products-services/catia/products/v5/portfolio/domain/Analysis/product/FMS/>. [Acedido em October 2015].
- [31] Julho 2015. [Online]. Available: <http://www.mathworks.com/matlabcentral/fileexchange/25555-mesh2d-automatic-mesh-generation>. [Acedido em July 2015].
- [32] G. S. P.-O. Persson, “A simple Mesh Generator in MATLAB,” n° 46, 2004.
- [33] T. e. al., “PolyMesher: a general-purpose mesh generator for polygonal elements written in Matlab,” Springer-Verlag, 2012.
- [34] S. N. Campos, “Desenvolvimento de programa em linguagem Matlab para cálculo


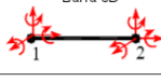
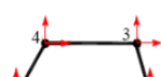


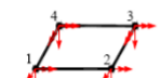

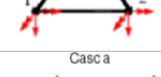


- pelo Método de Elementos Finitos,” Instituto Superior de Engenharia do Porto, Porto, 2014.
- [35] Outubro 2015. [Online]. Available: <http://www.mathworks.com/matlabcentral/>.
- [36] Outubro 2015. [Online].
- [37] Outubro 2015. [Online]. Available: <http://www.sagemath.org/tour.html>.
- [38] F. J. e. B. T., A first Course in Finite Elements, Wiley, New York, 2007.
- [39] C. WF., A primer for finite elements in elastic structures., New-York, Estados Unidos da América: Wiley, 1999.
- [40] B. A. Chandrupatla TR, Introduction to finite elements in Engineering, Upper Saddle River: Prentice-Hall, 1997.
- [41] B. SS., Finite Element Analysis., New Dehli: New Age International Publishers, 2005.
- [42] J. Reddy, Introduction to the Finite Element Method. Second Ed, New-York: McGraw-Hill, 1993.
- [43] B. Irons, “Engineering applications of numerical integration on stiffness methods.,” 1966.
- [44] R. Cook, Finite Element Modeling For Stress Analysis., New-York, 1995.
- [45] O. Zienkiewicz, The finite element method. 3rd ed., New-York: McGraw-Hill, 1977.
- [46] R. F. BG. Thomas, Calculus and analytic geometry., Reading: Addison-Wesley, 1984.
- [47] D. S. AH Stroud, Gaussian quadrature formulas, Englewood Cliffs: Prentice-Hall, 1966.
- [48] H. T. R. e. al., “A new Automated Scheme of Quadrilateral Mesh Generation for Finite Element Analysis,” vol. 3, n° 5, 2014.

[49] Y. W. K. e. H. Bang, The Finite Element Method Using MATLAB, 2nd Ed., CRC Press LLC, 2000.

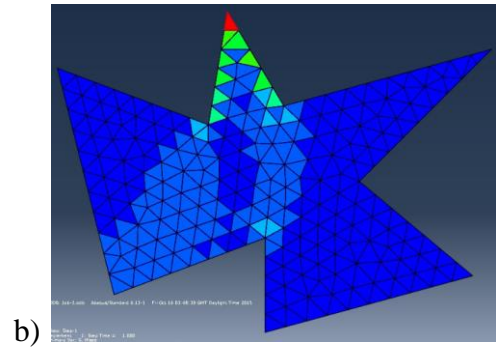
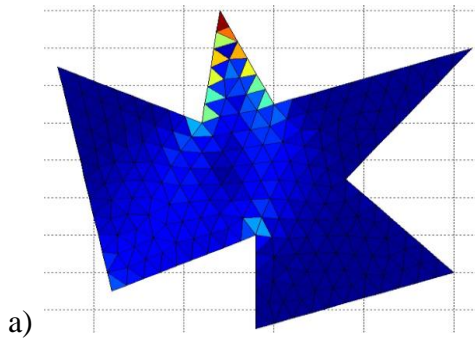
[50] Outubro 2015. [Online]. Available:
[http://www.mathworks.com/matlabcentral/fileexchange/43073-improved-digital-image-correlation--dic-](http://www.mathworks.com/matlabcentral/fileexchange/43073-improved-digital-image-correlation--dic-.). [Acedido em October 2015].

ANEXOS

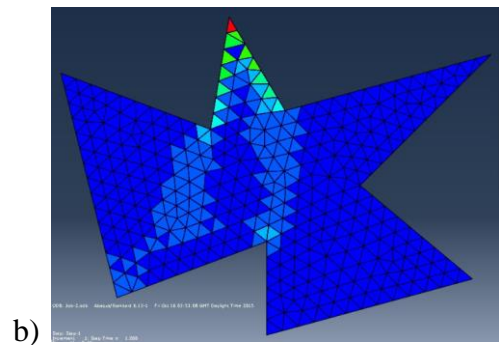
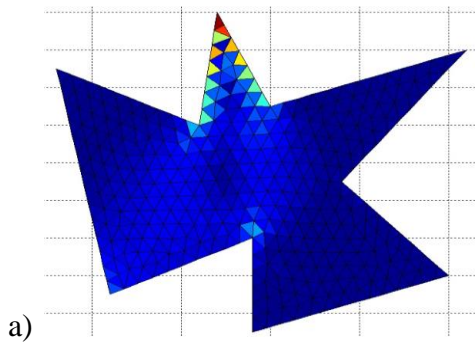
Anexo A - Tipos de elementos finitos e suas aplicações. [8]

Elementos finitos mais utilizados					
	Tipo Elemento Finito	Graus de Liberdade por nó	Numero de nós por elemento	Tipo de Aplicação	
Elementos Finitos	Lineares	<p>Barra 2D</p>  <p>Barra 3D</p> 	<p>3 GL/Nó (2 Translações + 1 Rotação)</p> <p>6 GL/Nó (3 Translações + 3 Rotações)</p>	<p>Barra com 2 nós - 6 GL por elemento Barra com 3 nós - 9 GL por elemento</p> <p>Barra com 2 nós - 12 GL por elemento Barra com 3 nós - 18 GL por elemento</p>	<p>Estruturas retículas planas, treliças, grelhas, muros de suporte</p> <p>Estruturas retículas tridimensionais, treliças, grelhas e muros de suporte</p>
	Laminares Planos	<p>Placa</p>   <p>Laje</p>  	<p>2 GL/Nó (2 Translações)</p> <p>3 GL/Nó (1 Translação + 2 Rotações)</p>	<p>Quadrangulares</p> <p>Placa com 4 nós - 8 GL por elemento Placa com 8 nós - 16 GL por elemento</p> <p>Triangulares</p> <p>Placa com 3 nós - 6 GL por elemento Placa com 6 nós - 12 GL por elemento</p> <p>Quadrangulares</p> <p>Laje com 4 nós - 12 GL por elemento Laje com 8 nós - 24 GL por elemento</p> <p>Triangulares</p> <p>Laje com 3 nós - 9 GL por elemento Laje com 6 nós - 18 GL por elemento</p>	<p>Estruturas planas sujeitas a acções segundo o seu plano médio</p> <p>Estruturas planas sujeitas a acções perpendiculares ao seu plano médio</p>
	Laminares Tridimensionais	<p>Casca</p>  	<p>6 GL/Nó (3 Translações + 3 Rotações)</p>	<p>Quadrangulares</p> <p>Casca com 4 nós - 24 GL por elemento Casca com 8 nós - 48 GL por elemento</p> <p>Triangulares</p> <p>Casca com 3 nós - 18 GL por elemento Casca com 6 nós - 36 GL por elemento</p>	<p>Casca</p>
	Tridimensionais	<p>Cubo</p>  <p>Tetraedro</p> 	<p>3 GL/Nó (3 Translações)</p>	<p>Cubo com 8 nós - 24 GL por elemento Cubo com 20 nós - 60 GL por elemento</p> <p>Tetraedro com 4 nós - 12 GL por elemento Tetraedro com 10 nós - 30 GL por elemento</p>	<p>Estruturas maciças Tridimensionais, como barragens e túneis</p>

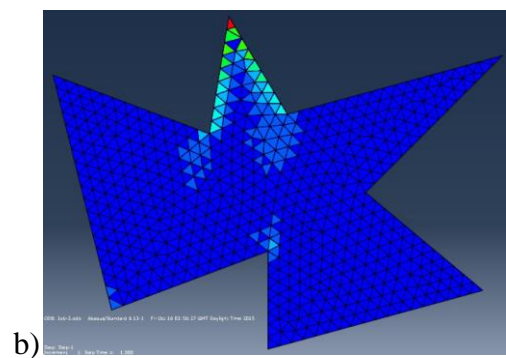
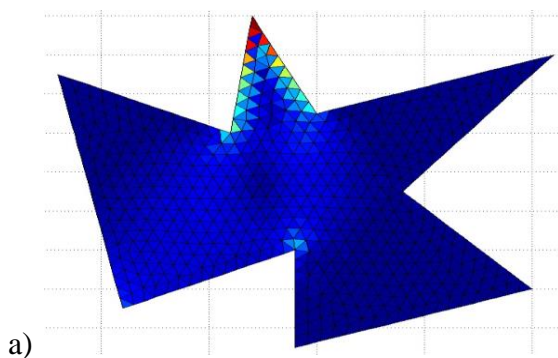
Anexo B - Resultados dos testes efetuados



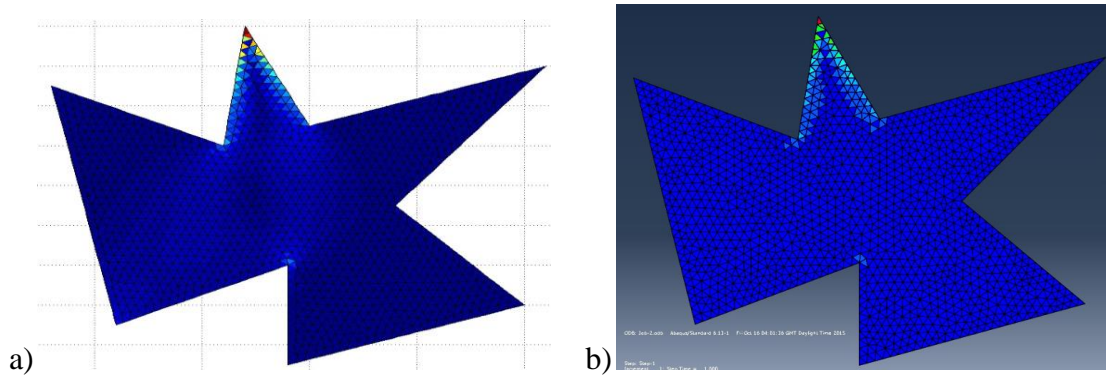
a) Figura com 384 elementos em MATLAB® e b) Figura com 367 elementos em ABAQUS®.



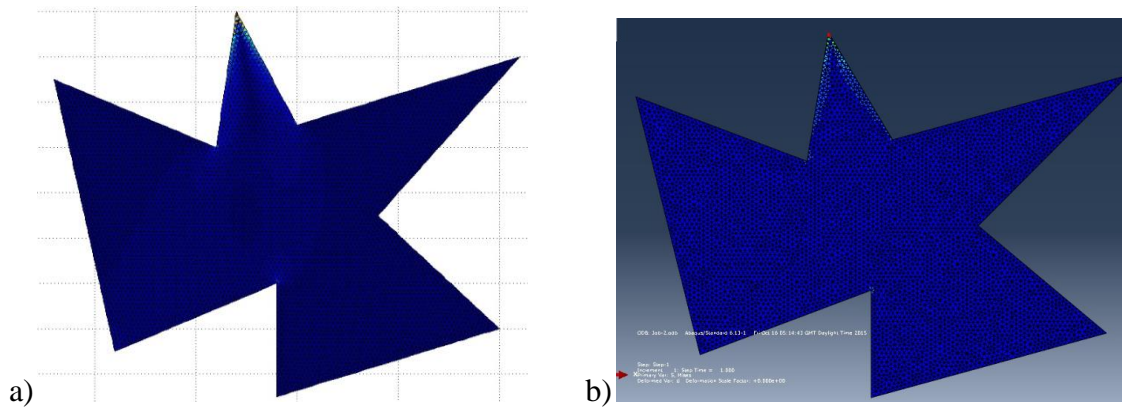
a) Figura com 595 elementos em MATLAB® e b) Figura com 602 elementos em ABAQUS®.



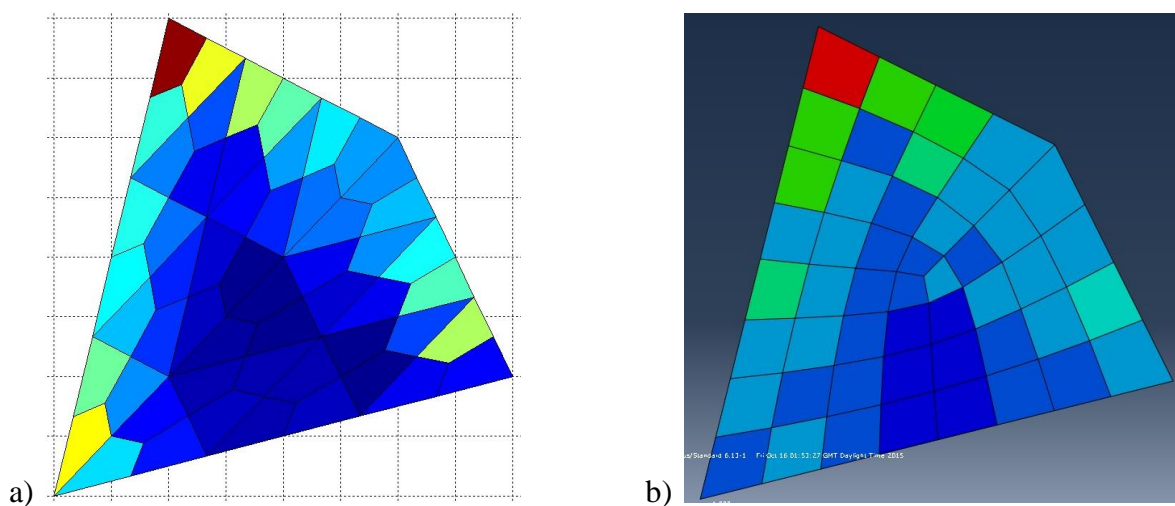
a) Figura com 1089 elementos em MATLAB® e b) Figura com 1042 elementos em ABAQUS®.



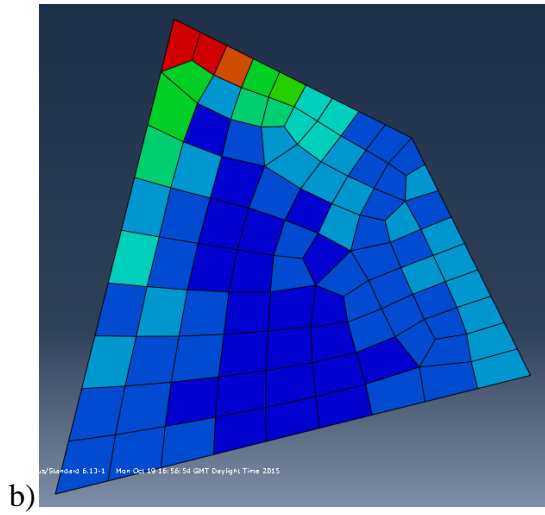
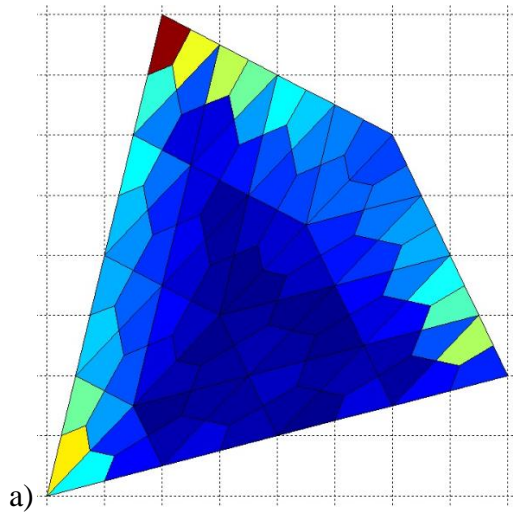
a) Figura com 2551 elementos em MATLAB® e b) Figura com 2582 elementos em ABAQUS.



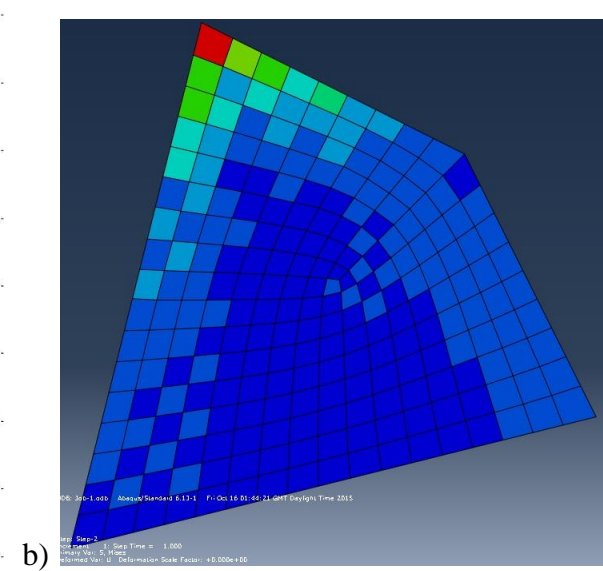
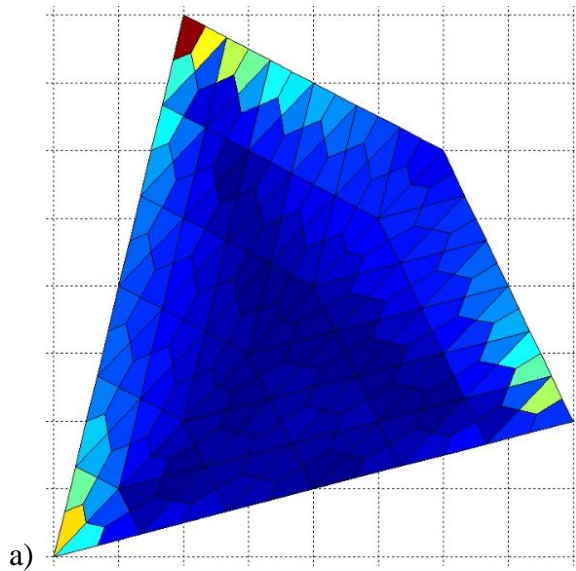
a) Figura com 10554 elementos em MATLAB® e b) Figura com 10008 elementos em ABAQUS.



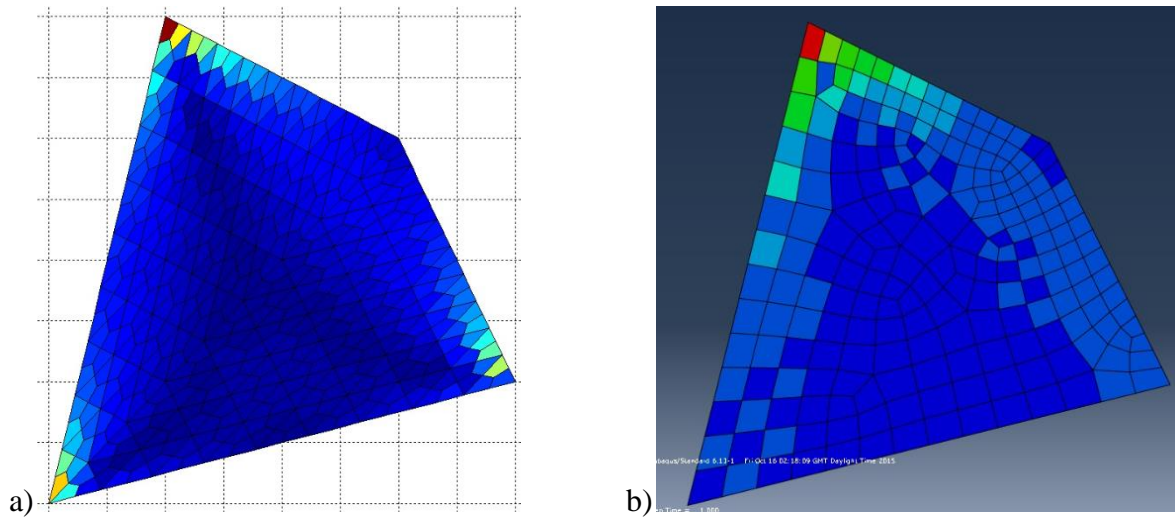
a) Figura com 54 elementos em MATLAB® e b) Figura com 48 elementos em ABAQUS®.



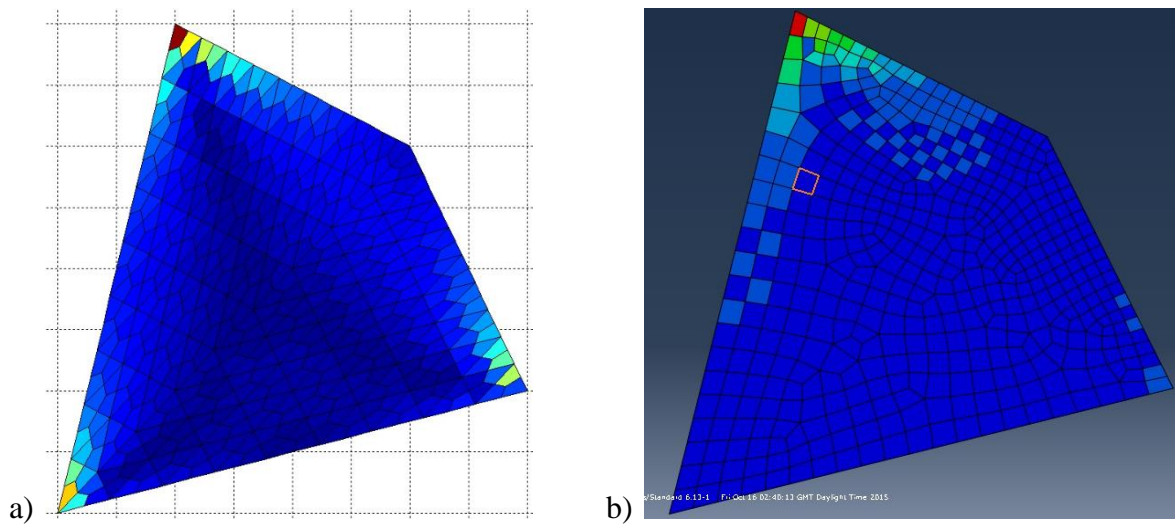
a) Figura com 96 elementos em MATLAB® e b) Figura com 93 elementos em ABAQUS®.



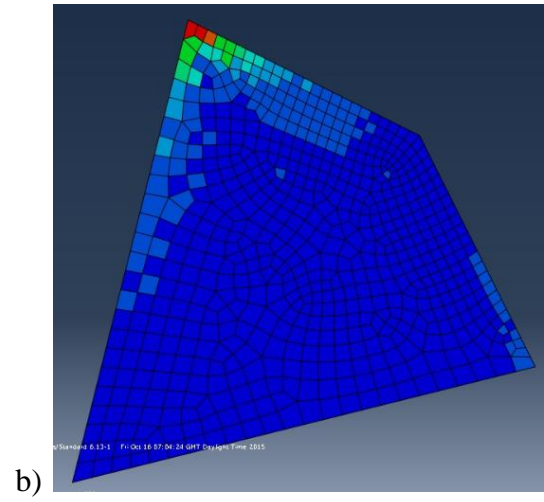
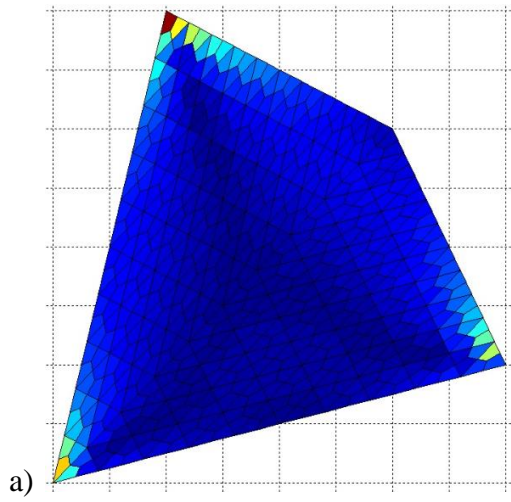
a) Figura com 216 elementos em MATLAB® e b) Figura com 225 elementos em ABAQUS®.



a) Figura com 384 elementos em MATLAB® e b) Figura com 341 elementos em ABAQUS®.



a) Figura com 486 elementos em MATLAB® e b) Figura com 472 elementos em ABAQUS®.



a) Figura com 600 elementos em MATLAB® e b) Figura com 644 elementos em ABAQUS®.