

Faculdade de Engenharia da Universidade do
Porto

Modelos de Elementos Finitos para Cálculo Estrutural em Código Aberto

Luís Miguel Cardoso Seixas



FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Mestrado Integrado em Engenharia Mecânica

Orientador: Doutor Sérgio M. O. Tavares

Coorientador: Professor Doutor Paulo M. S. Tavares de Castro

Setembro de 2019

Modelos de Elementos Finitos para Cálculo Estrutural em Código Aberto

Luís Miguel Cardoso Seixas

Mestrado Integrado em Engenharia Mecânica

Setembro de 2019

Resumo

Esta dissertação aborda a implementação de modelos de elementos finitos para cálculo estrutural com *software* em código livre e aberto. O objetivo desta implementação é analisar estruturalmente elementos dos transformadores de potência. O *software* de código aberto utilizado é o Salome-Meca, que permite realizar pré-processamento, cálculo e pós-processamento. Este programa inclui o *software* Code_Aster que é responsável pela componente de cálculo.

O primeiro exemplo estudado é o caso de uma placa com furo central, onde são apresentados os resultados obtidos utilizando o *software* para esta geometria assim como resultados teóricos, de forma a verificar as capacidades do Salome-Meca. Neste capítulo confirma-se que o Salome-Meca apresenta bastantes funcionalidades para o cálculo estrutural, assim como, concluí-se que os resultados obtidos apresentam uma diferença praticamente nula relativamente a resultados teóricos.

Um dos objetivos desta dissertação é realizar rotinas que possibilitem o cálculo automático por elementos finitos com a mínima interação com o utilizador. A partir da introdução dos dados de entrada, este algoritmo realiza a criação de ficheiros necessários ao estudo e, de seguida, é realizado o pré-processamento onde são geradas a geometria e a malha de elementos finitos a partir de código Python no ambiente do Salome-Meca. Após este processo é realizado o cálculo com funções do Code_Aster. Por fim, existe a visualização dos resultados no módulo ParaView ou um tratamento dos dados de saída. A partir da arquitetura apresentada foi possível implementar dois exemplos: análise de painéis e análise de cubas.

Na análise de painéis é estudado o cálculo automático de painéis planos simples e reforçados. As comparações destes resultados com os teóricos ou obtidos por outros *softwares*, permitem concluir que a rotina calcula com bastante rapidez e aproximação o comportamento mecânico destes componentes. Ainda são realizadas análises utilizando a rotina implementada de forma a otimizar estruturalmente os painéis.

A partir da rotina projetada criou-se um algoritmo que permite cálculo estrutural automático de cubas de transformadores. Este exemplo é validado com outro *software* de elementos finitos onde se verifica que os resultados não diferem em praticamente toda a geometria. Também foram criados vários exemplos de cubas sem reforços que são comparados com painéis simples de forma a obter uma relação entre as duas análises.

Palavras-chave: Transformador de Potência; Análise de elementos finitos; Salome-Meca; Code_Aster; Cálculo automático; Otimização estrutural; Cubas; Painéis; Reforços.

Esta página foi intencionalmente deixada em branco.

Abstract

This dissertation addresses the implementation of free and open source software for finite element analysis and structural calculation. The components studied concern power transformers. The open source software used is Salome-Meca, which allows preprocessing, calculation and postprocessing. This program includes the software Code_Aster which is responsible for the solver component.

The first example studied is the plate with a central circular hole, where software and theoretical results are presented and compared, to verify Salome-Meca capabilities. In this chapter, it is confirmed that Salome-Meca has good functionalities to perform structural calculations, and the results exhibit almost zero difference with respect to theoretical results.

One of the main goals of this dissertation is to perform routines that allow automatic calculation of finite elements models with minimal interaction with the user. From the input data, this algorithm creates the necessary files for the study and performs the preprocessing, where the geometry and the mesh are generated from Python code in the environment of Salome-Meca. After this process, the calculation with functions of Code_Aster is performed. Finally, the user is able to visualize the results in ParaView module or output data processing may be executed. This architecture is applied to two examples: panel analysis and tank analysis.

In panel analysis the automatic calculation of simple and stiffened plan panels was studied. The comparisons of these results with theoretical ones or obtained using other software, show that this routine achieves results quickly and with good accuracy. Also the analysis to structurally optimize panels was performed using these algorithms.

From the designed routine, an algorithm is created that allows the analysis of transformer tanks automatically. This example is validated with other finite elements software where it can be verified that the results do not differ across the geometry. Several examples of unreinforced tanks analyses have also been created and compared to simple panels to achieve a relationship between the mechanical behaviour of these two structures.

Keywords: Power transformer; Finite elements analysis; Salome-Meca; Code_Aster; Automatically calculation; Structural optimization; Tanks; Panels; Stiffeners.

Esta página foi intencionalmente deixada em branco.

Agradecimentos

Esta dissertação é resultado de vários contributos para o seu desenvolvimento e é o culminar de vários anos de formação e experiência pessoal. Como tal, gostaria de prestar os meus agradecimentos às pessoas e identidades que influenciaram esta realização.

Em primeiro lugar gostaria de agradecer ao meu orientador, Doutor Sérgio Tavares, pela acompanhamento e conhecimento transmitido, com críticas oportunas para o desenvolvimento sustentável deste projeto e pela proposta deste tema, que me permitiu ganhar o maior interesse pela combinação das áreas científicas de análise estrutural e programação. De igual forma, agradeço ao meu co-orientador, o Professor Doutor Paulo Tavares de Castro, pelas relevantes intervenções e revisões que permitiram que este trabalho beneficiasse da sua experiência reconhecida.

Esta dissertação foi realizada no âmbito de uma parceria com a empresa Efacec, à qual quero prestar os meus sinceros agradecimentos, na pessoa do Engenheiro Carlos Carvalho, pela oportunidade e recursos disponibilizados e pela estratégia de contacto próximo com as instituições de ensino. Em especial a todos os colaboradores da equipa de I&D de Estudos Mecânicos, na pessoa do Engenheiro Hélder Mendes, por todos os conselhos e abertura.

A todos os que me acompanharam nos últimos cinco anos, que tornaram a FEUP não só um lugar que me formou academicamente mas também me deu amigos para a vida. O companheirismo, as gargalhadas e todos os ensinamentos proporcionados, desde tertúlias a clubes, serão recordados e sem eles o meu percurso não teria sido tão enriquecedor.

À Beatriz, que me sempre apoiou incondicionalmente, por todo o carinho e paciência, pelos desabafos num final de dia de trabalho e pelos sorrisos quando mais precisava.

Por fim, mas não com menor importância, gostaria de agradecer à minha família. Ao meu irmão João, pelo seu apoio e momentos divertidos. Ao meu irmão Bruno, pela figura de inspiração e pelos seus ensinamentos. E aos meus Pais, por terem investido na minha educação, com todos os esforços e sacrifícios e a eles devo-lhes o meu futuro.

Esta página foi intencionalmente deixada em branco.

“E pur si muove!”

(“E no entanto, ela move-se!”)

Galileo Galilei

Esta página foi intencionalmente deixada em branco.

Conteúdo

1	Introdução	1
1.1	Contextualização	1
1.1.1	Efacec	2
1.1.2	iCubas5D	3
1.2	Motivações e objetivos	3
1.3	Projeto	4
1.3.1	Metodologia	4
1.3.2	Estrutura	4
2	Estado de Arte	7
2.1	Transformadores	7
2.1.1	Componentes estruturais de um transformador <i>core</i>	10
2.1.2	Análises estruturais de transformadores	12
2.1.3	Método dos Elementos Finitos	14
2.2	Salome-Meca/Code_Aster	15
2.2.1	Módulo geometria	16
2.2.2	Módulo malha	17
2.2.3	Code_Aster/AsterStudy	18
2.2.4	ParaView	19
2.3	Formulações Code_Aster	20
2.3.1	Estado plano tensão	20
2.3.2	Teoria das Placas	21
2.3.3	Teoria das Cascas	21
3	Placa com Furo Central	23
3.1	Fundamentos Teóricos	23
3.1.1	Fator de concentração de tensão	24
3.1.2	Distribuição da tensão	25
3.2	Aplicação	27
3.3	Implementação no Salome-Meca	28
3.3.1	Análise da malha de elementos	31
3.3.2	Refinamento gradual da malha	35
3.3.3	Elementos quadráticos e bi-quadráticos	36
3.3.4	Análise da distribuição da tensão	37
3.3.5	Análise de uma placa a três dimensões	39
3.3.6	Variação do diâmetro do furo	41

3.4	Discussão de resultados	42
4	Arquitetura de uma Rotina de Análise Estrutural Otimizada	45
4.1	Ler dados de entrada e criar ficheiros	46
4.1.1	Ficheiro com formato <i>json</i>	49
4.1.2	Ficheiro com formato <i>py</i> para pré-processamento	50
4.1.3	Ficheiro com formato <i>med</i> para <i>input</i>	50
4.1.4	Ficheiro com formato <i>hdf</i> para visualização do modelo	50
4.1.5	Ficheiro com formato <i>comm</i>	51
4.1.6	Ficheiro com formato <i>export</i>	51
4.1.7	Ficheiro com formato <i>mess</i>	51
4.1.8	Ficheiro com formato <i>med</i> para <i>output</i>	52
4.1.9	Ficheiro com formato <i>py</i> para pós-processamento	52
4.1.10	Pastas para armazenar ficheiros	52
4.1.11	Ficheiro com formato <i>batch</i> para executar o Salome-Meca e Code_Aster	52
4.2	Preparação de ficheiros	53
4.2.1	Arquitetura do ficheiro <i>json</i>	54
4.2.2	Arquitetura do ficheiro <i>py</i> para pré-processamento	54
4.2.3	Arquitetura do ficheiro <i>comm</i>	56
4.2.4	Arquitetura do ficheiro <i>export</i>	57
4.2.5	Arquitetura do ficheiro <i>py</i> para pós-processamento	59
4.3	Processos da análise	59
4.4	Discussão da rotina	62
5	Análise Estrutural de Painéis	63
5.1	Rotina de análise de painéis otimizada	64
5.1.1	Dados de entrada	65
5.1.2	Preparação de ficheiros	68
5.1.2.1	Algoritmo de pré-processamento	69
5.1.2.2	Algoritmo de análise por elementos finitos	71
5.1.2.3	Ficheiro <i>export</i>	73
5.1.2.4	Algoritmo de pós-processamento	73
5.1.3	Processos da análise	77
5.2	Painéis planos simples	77
5.2.1	Fundamentos teóricos	78
5.2.1.1	Equações diferenciais de equilíbrio	81
5.2.1.2	Solução para um painel simplesmente apoiado	83
5.2.1.3	Solução para um painel encastrado	86
5.2.2	Aplicação	86
5.2.3	Análise da sensibilidade da malha de elementos finitos	87
5.2.4	Análise da influência do parâmetro geométrico k	88
5.2.5	Comparação entre caso simplesmente apoiado e caso encastrado	91
5.3	Painéis planos reforçados	93
5.3.1	Aplicação	94
5.3.2	Análise da sensibilidade da malha de elementos finitos	95

5.3.3	Análise da influência dos parâmetros geométricos	96
5.3.3.1	Espessura do painel	98
5.3.3.2	Número de reforços verticais	99
5.3.3.3	Altura dos reforços verticais	100
5.3.3.4	Espessura dos reforços verticais	100
5.3.3.5	Reforço horizontal	101
5.3.3.6	Discussão da análise	101
5.3.4	Otimização de um painel plano reforçado	103
5.3.5	Comparação entre caso simplesmente apoiado e caso encastrado	106
5.3.6	Validação de resultados	107
5.4	Discussão de resultados	111
6	Análise Estrutural de Cubas	113
6.1	Rotina de análise de uma cuba otimizada	113
6.1.1	Dados de entrada	114
6.1.2	Preparação de ficheiros	115
6.1.2.1	Algoritmo de pré-processamento	115
6.1.2.2	Algoritmo de análise por elementos finitos	116
6.1.2.3	Algoritmo de pós-processamento	118
6.2	Aplicação	119
6.3	Análise da sensibilidade da malha de elementos finitos	120
6.4	Resultados e validação	122
6.5	Comparação de resultados com a análise painel plano reforçado . . .	127
6.6	Comparação com painéis simples	128
6.7	Discussão de resultados	131
7	Conclusões e Trabalhos Futuros	133
7.1	Conclusões	133
7.2	Trabalhos futuros	135
	Referências	137

Esta página foi intencionalmente deixada em branco.

Lista de Figuras

1.1	Marca Efacec.	2
2.1	Geração, transmissão e distribuição da energia elétrica.	7
2.2	Representação do funcionamento de um transformador.	9
2.3	Tipos de construção da parte ativa.	10
2.4	Principais componentes de um transformador de potência.	10
2.5	Tipos de reforços utilizados.	11
2.6	Ambiente gráfico do módulo geometria.	17
2.7	Ambiente gráfico do módulo malha.	18
2.8	Ambiente gráfico módulo AsterStudy.	19
2.9	Ambiente gráfico do módulo ParaView.	20
3.1	Dimensões de uma placa semi-infinita.	23
3.2	Comparação entre a função de aproximação apresentado por Pilkey e os valores obtidos por Howland.	26
3.3	Referencial em coordenadas polares.	26
3.4	Geometria implementada no <i>software</i>	29
3.5	Malha de elementos finitos com 5 elementos por fronteira.	29
3.6	Representação gráfica do pós-processamento das tensões segundo x	31
3.7	Fator de concentração de tensão em função do número de elementos.	33
3.8	Tempo de computação em função do número de elementos.	34
3.9	Refinamento gradual da malha de elementos finitos.	35
3.10	Representação gráfica da distribuição da tensão circunferencial no “Teste 10”.	38
3.11	Comparação entre distribuição teórica e distribuição obtida por elementos finitos.	38
3.12	Malha de elementos finitos do modelo a três dimensões.	39
3.13	Distribuição da tensão circunferencial ao longo da placa.	40
3.14	Comparação entres os valores obtidos por elementos finitos e valores de referência.	42
4.1	Fluxograma da rotina principal.	47
4.2	Fluxograma da rotina de pré-processamento.	61
4.3	Fluxograma da rotina de cálculo.	61
4.4	Fluxograma da rotina de pós-processamento para visualização.	62
5.1	Classes da rotina de análise de painéis.	66

5.2	Tipos de geometria de painéis planos.	69
5.3	Tipos de geometria de painéis curvos.	70
5.4	Tipos de condições de fronteira presentes na rotina.	72
5.5	Relatório final de uma análise de vários modelos de painéis curvos reforçados.	75
5.6	Exemplo da representação gráfica da distribuição da tensão de von Mises [MPa].	76
5.7	Representação geométrica de um painel plano simples.	77
5.8	Representação de uma placa e respetivo referencial (esquerda) e representação de uma secção lateral antes e depois da solitação (direita).	78
5.9	Equilíbrio de esforços na placa sujeita a pressão distribuída uniforme.	81
5.10	Malha de elementos finitos de um painel plano simples com tamanho de elemento igual a 40 mm.	88
5.11	Distribuição dos deslocamentos em função de k	90
5.12	Distribuição da tensão de von Mises em função de k	91
5.13	Representação do deslocamento segundo z de um painel plano simples.	92
5.14	Representação da tensão equivalente de von Mises de um painel plano simples.	93
5.15	Representação geométrica de um painel plano reforçado.	94
5.16	Representação gráfica da geometria de um painel plano reforçado no Salome-Meca.	95
5.17	Malha de elementos finitos de um painel plano reforçado com tamanho de elemento igual a 40 mm.	97
5.18	Representação da tensão equivalente de von Mises em função da massa para cada análise paramétrica de um painel plano reforçado.	102
5.19	Representação da tensão equivalente de von Mises em função da massa para obter um painel plano reforçado otimizado.	103
5.20	Campo de deslocamentos do modelo ótimo caso simplesmente apoiado.	105
5.21	Campo das tensões equivalentes de von Mises do modelo ótimo caso simplesmente apoiado.	105
5.22	Campo de deslocamentos do modelo ótimo caso encastrado.	106
5.23	Campo das tensões equivalentes de von Mises do modelo ótimo caso encastrado.	107
5.24	Representação do deslocamento segundo z de um painel plano reforçado a partir do ANSYS.	108
5.25	Representação da tensão von Mises de um painel plano reforçado a partir do ANSYS.	109
5.26	Distribuição da tensão de von Mises num reforço vertical.	110
6.1	Classes da rotina de análise de cubas.	114
6.2	Exemplo de geometria gerada através do algoritmo de pré-processamento.	117
6.3	Exemplo da visualização de resultados utilizando a rotina implementada para análise estrutural de cubas.	118
6.4	Representação geométrica de uma cuba simplificada.	119
6.5	Malha de elementos finitos de uma cuba com tamanho de elemento igual a 40 mm.	121

6.6	Exemplo de local onde ocorre tipicamente picos de tensão na análise de uma cuba.	122
6.7	Campo de resultados dos deslocamentos para exemplo de aplicação de análise de uma cuba.	124
6.8	Campo de resultados da tensão de von Mises para exemplo de aplicação de análise de uma cuba.	125
6.9	Distribuição da tensão de von Mises obtida por diferentes <i>softwares</i>	126
6.10	Distribuição da tensão de von Mises obtida com diferentes condições de fronteira.	128

Esta página foi intencionalmente deixada em branco.

Lista de Tabelas

2.1	Especificações dos <i>softwares</i> integrados no Salome-Meca 2018 para o Windows	16
3.1	Propriedades de um exemplo da placa semi-infinita com furo central	28
3.2	Valores teóricos da tensão máxima e fator de concentração de tensão	28
3.3	Resultados obtidos após refinamento da malha	32
3.4	Tempo de computação dos teste para análise da malha	33
3.5	Resultados após o refinamento gradual da malha de elementos finitos	36
3.6	Resultados após implementação de elementos QUAD8 e QUAD9	37
3.7	Resultados após implementação do modelo 3D	40
3.8	Resultados obtidos após variar o diâmetro do furo	41
5.1	Fatores de multiplicação para um painel simplesmente apoiado	85
5.2	Fatores de multiplicação para um painel encastrado.	86
5.3	Propriedades de um exemplo de um painel plano simples	87
5.4	Valores teóricos do exemplo de um painel plano simples	87
5.5	Resultados obtidos na análise da sensibilidade da malha	88
5.6	Resultados obtidos para diferentes rácios geométricos de um painel simplesmente apoiado	89
5.7	Resultados obtidos para diferentes rácios geométricos de um painel encastrado	90
5.8	Propriedades de um exemplo de um painel plano reforçado	95
5.9	Resultados obtidos na análise da sensibilidade da malha de um painel plano reforçado	96
5.10	Análise da influência da espessura de um painel plano reforçado	98
5.11	Análise da influência do número de reforços verticais de um painel plano reforçado	99
5.12	Análise da influência da altura dos reforços verticais de um painel plano reforçado.	100
5.13	Análise da influência da espessura dos reforços verticais de um painel plano reforçado	101
5.14	Propriedades do modelo obtido após análises paramétricas	104
5.15	Resultados do modelo ótimo para o caso encastrado	106
5.16	Multiplicadores de carga para primeiro modo de encurvadura para o modelo ótimo	111

6.1	Variáveis geométricas de uma cuba simplificada	119
6.2	Propriedades de um exemplo de uma cuba de um transformador . . .	120
6.3	Resultados obtidos na análise da sensibilidade da malha de uma cuba	121
6.4	Deslocamentos máximos obtidos para cada direção em diferentes <i>softwa-</i> <i>res</i>	123
6.5	Deslocamento máximo obtidos para um painel com diferentes condi- ções de fronteira	127
6.6	Propriedades da cuba para comparação com painel simples	128
6.7	Deslocamentos para diferentes dimensões de painéis simples com di- ferentes condições de fronteira	129
6.8	Tensão von Mises para diferentes dimensões de painéis simples com diferentes condições de fronteira	130

Lista de Códigos

4.1	Classe criada segundo Python para a geração de uma face.	47
4.2	Estrutura de um ficheiro <i>json</i>	54
4.3	Rotina para gerar a malha de elementos finitos de um quadrado	55
4.4	Rotina para cálculo por elementos finitos no Code_Aster	57
4.5	Estrutura de um ficheiro <i>export</i>	58
4.6	Rotina para visualizar resultados no Salome-Meca	59
4.7	Rotina para executar os processos da análise	60

Esta página foi intencionalmente deixada em branco.

Nomenclatura

Abreviaturas

<i>Aster</i>	<i>Analyses des Structures et Thermomécanique pour des Études et des Recherches</i>
<i>CAD</i>	<i>Computer Aided Design</i>
<i>CEA/DEN</i>	<i>Commissariat à l'énergie atomique et aux énergies alternatives/Direction de l'énergie nucléaire</i>
<i>comm</i>	<i>Command</i>
<i>EDF</i>	<i>Électricité de France</i>
<i>FEA</i>	<i>Finite Element Analysis</i>
<i>hdf</i>	<i>Hierarchical Data Format</i>
<i>json</i>	<i>JavaScript Object Notation</i>
<i>med</i>	<i>Modélisation et Échanges de Données</i>
<i>mess</i>	<i>Message</i>
<i>py</i>	<i>Formato Python</i>

Símbolos

<i>a</i>	Comprimento segundo <i>x</i>
<i>a_ci</i>	Comprimento segundo <i>x</i> da cuba inferior
<i>b</i>	Comprimento segundo <i>y</i>
<i>b_ci</i>	Comprimento segundo <i>y</i> da cuba inferior
<i>b_tmp</i>	Comprimento segundo <i>y</i> da tampa
<i>bo_ci</i>	Comprimento do bordo da cuba inferior
<i>bo_tmp</i>	Comprimento do bordo da tampa
<i>c</i>	Comprimento segundo <i>z</i>
<i>c_ci</i>	Comprimento segundo <i>z</i> da cuba inferior
<i>c_tmp</i>	Comprimento segundo <i>z</i> da tampa
<i>d</i>	Diâmetro
<i>D</i>	Módulo de rigidez à flexão
<i>dif</i>	Diferença relativa
<i>E</i>	Módulo de Young

E_1, E_2	Forças eletromotrizes induzidas em cada corrente
F	Força
G	Módulo de cisalhamento
h_{rh}	Altura de reforço horizontal
h_{rv}	Altura dos reforços verticais
I	Intensidade de corrente
k	Rácio entre as dimensões do painel
K_{tg}	Fator de concentração de tensão nominal
K_{tn}	Fator de concentração de tensão total
L	Comprimento total da placa com furo central
m	Massa
M_i	Momento fletor segundo a direção i
M_{ij}	Momento tursor segundo a direção i , normal à direção j
n	Coefficiente de segurança
N	Número de espiras
n_{rv}	Número de reforços verticais
p	Pressão
P	Potência
Q_i	Esforço transversal segundo a direção i
r	Raio
R	Resistência elétrica
t	Espessura
t_{ci}	Espessura da cuba inferior
t_{bo}	Espessura do bordo
t_{rh}	Espessura do reforço horizontal
t_{rv}	Espessura dos reforços verticais
t_{tmp}	Espessura tampa
u	Deslocamento segundo x
v	Deslocamento segundo y
V	Tensão elétrica
w	Deslocamento segundo z
W	Largura total da placa com furo central
x_{enc}	Coefficiente de relação entre resultados da cuba e painel encastrado
x_{sap}	Coefficiente de relação entre resultados da cuba e painel simplesmente apoiado
x_{tot}	Coefficiente de relação entre resultados da cuba, painel encastrado e painel simplesmente apoiado
γ_{ij}	Deformação de corte segundo a direção i , normal à direção j
γ_n	Rácio entre diâmetro do furo e largura da placa
$\delta_1, \delta_2, \delta_3$	Fatores de multiplicação para cálculo do deslocamento w , momento segundo x e momento segundo y , respetivamente
ε_i	Deformação normal segundo a direção i
ν	Coefficiente de Poisson
ρ	Massa volúmica

σ_{adm}	Tensão admissível
σ_i	Tensão normal segundo a direção i
σ_{max}	Tensão máxima
σ_n	Tensão nominal
σ_{vmis}	Tensão equivalente, critério de von Mises
σ_Y	Tensão de cedência
σ_∞	Solicitação remota
τ_{ij}	Tensão de corte segundo a direção i , normal à direção j
Φ_B	Fluxo magnético

Capítulo 1

Introdução

Os modelos de elementos finitos têm nos dias de hoje um papel preponderante na análise estrutural de componentes mecânicos. Estas ferramentas numéricas permitem desenvolver análises que simulam o comportamento de diversas geometrias, materiais e tipos de solicitação e a sua integração tanto a nível académico como a nível industrial permite desenvolver em grande escala produtos que no passado estariam fora do espectro de aplicação. A proliferação das análises por elementos finitos acompanhou desde o seu início o desenvolvimento computacional e de programas capazes de reproduzir estes cálculos utilizando os recursos da algoritmia. Esta solução permitiu a criação de *softwares* que utilizam as simulações por elementos finitos para calcular modelos que seriam praticamente impossíveis sem acesso aos computadores. No âmbito desta dissertação foi proposto a utilização do *software* Code_Aster/AsterStudy que se caracteriza por ser gratuito, de código aberto e utilizado por múltiplos sectores industriais. Esta dissertação terá como foco a apresentação deste *software* de código aberto de elementos finitos assim como a exploração das suas capacidades, implementação de exemplos práticos e rotinas de cálculo automático.

1.1 Contextualização

O tema desta dissertação enquadra-se numa proposta por parte da empresa Efacec Power Solutions, designada posteriormente por Efacec, e mais especificamente pelo departamento de investigação e desenvolvimento da Unidade de Transformadores. Este projeto pretende aplicar simulações de elementos finitos utilizando um

software de código aberto a componentes estruturais de transformadores. A utilização destes programas no contexto industrial desta empresa permite retirar dependência de *softwares* comerciais com custos de licenciamento elevado para além de favorecer a utilização de otimizações estruturais dos componentes que se reflete em menores custos de produção.

1.1.1 Efacec

A Efacec é uma empresa portuguesa que cria soluções para os sectores de Energias, Ambiente e Mobilidade. Com mais de 70 anos de história, teve desde início um papel importante no setor industrial tendo sido uma das primeiras empresas portuguesas cotadas na Bolsa de Valores de Lisboa. Também tem uma presença forte no mercado internacional, com cerca de 75% das receitas totais provenientes de exportações.

A Efacec sempre se distinguiu pelo fabrico de máquinas elétricas, como é o caso dos transformadores de energia elétrica. Os transformadores têm um papel vital em toda a sociedade possibilitando o consumo de energia por parte da população. Apesar do seu funcionamento basear-se em fenómenos eletromagnéticos, durante várias etapas do seu projeto existem componentes de Engenharia Mecânica como o dimensionamento de estruturas metálicas de construção soldada. Este dimensionamento exige várias considerações como análise de condições de funcionamento, criação de modelos CAD ou ainda análise estruturais.



Figura 1.1: Marca Efacec.

1.1.2 iCubas5D

O projeto iCubas5D do departamento de investigação e desenvolvimento da Unidade de Transformadores visa responder a um novo passo industrial com a integração de processos para o desenho e cálculo de cubas de transformadores. O objetivo deste projeto é integrar ferramentas de representação 3D com cálculo e desenho automático através de modelos paramétricos. Assim, é possível reduzir o tempo de fabrico de uma cuba de um transformador e obter modelos mais simples e otimizados.

No contexto deste projeto, esta dissertação integra-se na investigação de ferramentas de análise por elementos finitos e implementação de rotinas de cálculo automático.

1.2 Motivações e objetivos

O tema desta dissertação nasce de uma necessidade de cálculo por elementos finitos de forma automática e eficiente utilizando as características mais interessantes de programas de código aberto como o seu acesso livre e transparência dos algoritmos destes *softwares*. Com a conclusão deste projeto pretende-se validar o *software Code_Aster* para cálculo de modelos de elementos finitos assim como implementar rotinas de cálculo. Ainda se pretende estudar parametricamente vários elementos estruturais de cubas do transformador, com maior incidência nos painéis, para prever otimizações estruturais destes componentes.

Os objetivos desta dissertação são os seguintes:

- Análise e exploração de *softwares* de elementos finitos em código aberto para cálculo estrutural;
- Desenvolvimento de modelos de elementos finitos;
- Verificação de algoritmos de otimização estrutural com base em modelos de elementos finitos;
- Desenvolvimento de rotinas de otimização.

A ferramenta de cálculo automático permite uma análise mais rápida, de fácil implementação e com menores custos quando comparada com a utilização de *softwares* comerciais de elementos finitos e sem a utilização destas rotinas ou até comparando com outros métodos, como os analíticos. Por estas razões prevê-se com o desenvolvimento desta dissertação a possibilidade de integração destas ferramentas em programas de dimensionamento de cubas de transformadores.

1.3 Projeto

Este projeto foi desenvolvido em ambiente empresarial, sendo que foi dividido em quatro fases: pesquisa e aprendizagem de conceitos necessários; implementação de exemplos; desenvolvimento de rotinas de cálculo automático; aplicação das rotinas para análises paramétricas. Na primeira fase inicial procurou-se introduzir conceitos estruturais de cubas de transformadores e obter contacto com o *software*. De seguida, implementou-se exemplos de modelos de elementos finitos e realizou-se algumas análises. Após ter sido obtido algum conhecimento do *software* criaram-se rotinas de cálculo automático. Por fim, estabeleceu-se análises paramétricas utilizando as rotinas implementadas de forma a estudar estruturalmente os componentes assim como validar estes resultados.

1.3.1 Metodologia

A metodologia deste trabalho seguiu os seguintes passos:

- Pesquisa a partir de literatura sobre o projeto mecânico de transformadores;
- Pesquisa a partir de literatura sobre os *softwares* utilizados;
- Pesquisa a partir de literatura sobre conceitos de análises de elementos finitos;
- Desenvolvimento de análises de modelos de elementos finitos;
- Desenvolvimento de rotinas de cálculo automático;
- Implementações e análises paramétricas utilizando as rotinas criadas;
- Comparação e validação dos resultados a partir de outras fontes.

1.3.2 Estrutura

Este documento encontra-se dividido em sete capítulos, para além do resumo, das referências bibliográficas e dos anexos. O objetivo desta leitura é proporcionar ao leitor um sentido crescente de conhecimentos, sendo os primeiros capítulos introdutórios ao tema, os capítulos intermédios de desenvolvimento e os capítulos finais onde se utiliza os conceitos estudados e de conclusão.

O primeiro capítulo, é o presente, e tem como objetivo introduzir a dissertação, contextualizar e explicitar os objetivos.

O segundo capítulo diz respeito ao estado de arte, e apresenta conceitos já estabelecidos e que são necessários aos capítulos posteriores. Neste âmbito foi apresentado os conceitos sobre transformadores, de análise de elementos finitos e ainda foi apresentado o *software* utilizado.

O terceiro capítulo tem como objetivo criar um modelo introdutório ao *software* sendo que foi escolhido uma placa com um furo central. Neste capítulo são apresentadas várias análises, comparações e implementações no *software*

O quarto capítulo diz respeito à apresentação de uma arquitetura de uma rotina que permite cálculo automático utilizando o *software* de elementos finitos. Neste capítulo é apresentado a rotina de forma abrangente a várias aplicações, sendo que é abordado cada passo da sua construção.

O quinto capítulo aborda a implementação da rotina de cálculo automático na análise estrutural de painéis de diversas geometrias, reforçados ou simples e com diferentes condições de fronteira. São realizadas análises paramétricas, exemplos de aplicação, otimizações estruturais e validações de resultados.

O sexto capítulo refere-se à implementação da rotina de cálculo automático na análise estrutural de uma cuba simples. São demonstrados exemplos de aplicação assim como validação de resultados para além de análises da influência de alguns fatores. Também foram comparados os resultados com os resultados de análise dos painéis isolados.

O sétimo capítulo procura chegar a conclusões gerais deste trabalho assim como explicitar alguns trabalhos que poderão ser realizados como continuação deste tema.

No que diz respeito aos capítulos de análise de componentes estruturais, o terceiro, o quinto e o sexto, procura-se sempre demonstrar exemplos, comparar com literatura ou outros *softwares*, validar resultados e realizar análise de influência das variáveis de cada modelo.

Esta página foi intencionalmente deixada em branco.

Capítulo 2

Estado de Arte

2.1 Transformadores

Os transformadores são máquinas elétricas que permitem transformar tensão elétrica entre dois circuitos e por consequência a corrente, mantendo como constante, salvo pequenas perdas, a potência elétrica associada ao circuito. A utilização destas máquinas tem motivos técnicos, económicos e de segurança visto que permite adaptar a tensão nas diferentes etapas desde a produção da energia até ao seu consumo como se pode observar pela figura 2.1, [20].

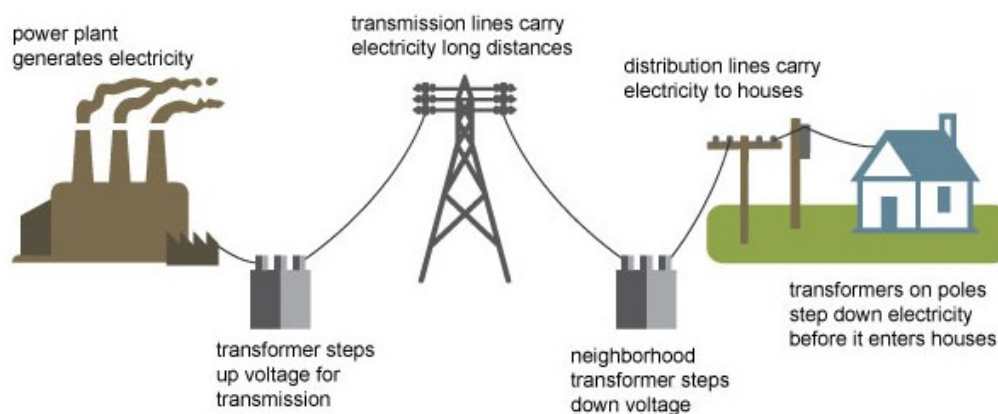


Figura 2.1: Geração, transmissão e distribuição da energia elétrica, [1].

Após a geração de energia elétrica existem várias etapas até que a eletricidade esteja disponível para consumo. Este longo caminho que frequentemente a energia tem de percorrer resulta em perdas energéticas devido à resistência dos cabos, como dita a seguinte relação:

$$P = VI = RI^2 \quad (2.1)$$

Na equação 2.1, P é a potência elétrica, V a tensão elétrica, I a intensidade de corrente e R a resistência elétrica.

Como se verifica, de forma a diminuir as perdas de potência elétrica ou se diminuir a resistência dos cabos ou a intensidade de corrente; como diminuir a resistência terá um custo económico elevado existe a necessidade do transformador que permite elevar a tensão que por sua vez diminui a intensidade de corrente.

Nas etapas em que a eletricidade tem de percorrer existe então um transformador que eleva a tensão para a transmissão de energia numa rede de alta tensão e seguidamente, perto de centros urbanos ou industriais, existe um transformador que transforma a eletricidade de alta tensão em média tensão. Os transformadores responsáveis por esta etapa são denominados de transformadores de potência. Dentro dos centros urbanos, ainda existem transformadores de distribuição que permitem reduzir de média tensão para baixa tensão e assim possibilitam o consumo desta energia, [13].

O funcionamento de um transformador baseia-se no princípio de indução apresentado por Michael Faraday em 1831. O princípio defende que caso um circuito elétrico se encontre na presença de um fluxo magnético variável é possível induzir uma força eletromotriz ou uma tensão nesse circuito. Sendo o circuito constituído por um enrolamento, a tensão será proporcional ao fluxo pelo número de espiras. Por sua vez, se dois circuitos estiverem sujeitos ao mesmo fluxo magnético irão ser induzidas duas correntes em que a tensão irá diferenciar entre os dois circuitos pelo número de espiras que o constituem. Esta formulação pode ser apresentada para um transformador ideal em que a força eletromotriz é igual à tensão induzida pelas seguintes relações, [20]:

$$V_1 = E_1 = N_1 \frac{d\Phi_B}{dt} \quad (2.2)$$

$$V_2 = E_2 = N_2 \frac{d\Phi_B}{dt} \quad (2.3)$$

$$\frac{V_1}{V_2} = \frac{N_1}{N_2} \quad (2.4)$$

Na equação 2.2, V_1 e V_2 são as tensões induzidas em cada corrente, E_1 e E_2 são as forças eletromotrizes induzidas em cada corrente, N_1 e N_2 o número de espiras de cada enrolamento e Φ_B o fluxo magnético.

Na figura 2.2 encontra-se uma representação do princípio de Faraday num transformador. Esta figura também representa a parte ativa de um transformador, que é constituída pelos enrolamentos e pelo núcleo.

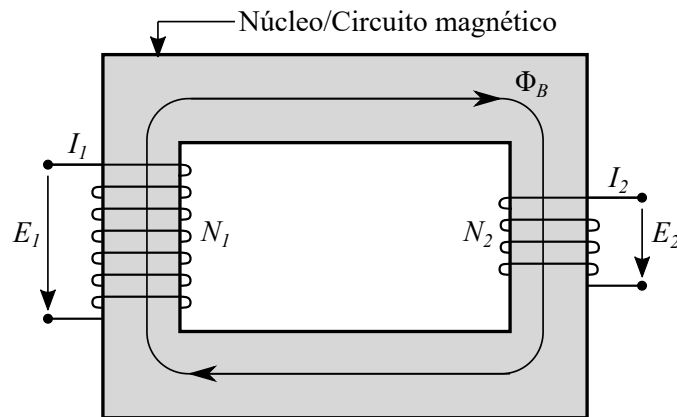


Figura 2.2: Representação do funcionamento de um transformador.

A invenção do transformador utilizando o princípio de Faraday foi uma das maiores razões da possibilidade do uso de corrente alternada e a larga distribuição da eletricidade. Antes desta máquina elétrica os equipamentos teriam de ser todos dimensionados para a mesma tensão de utilização, para além de, a eletricidade ser restringida a centros urbanos pela incapacidade de transmissão a grandes distâncias. O transformador veio então revolucionar toda a rede elétrica e é hoje um componente indispensável para o consumo de eletricidade, [20].

A construção de um transformador varia conforme a sua utilização seja para transformar alta tensão em média tensão, transformadores de potência, ou média em baixa tensão, transformadores de distribuição. Tipicamente os transformadores de potência têm uma parte ativa maior e estão sujeitos a maiores esforços mecânicos portanto é necessário uma maior robustez de toda a estrutura. Os transformadores também dependem do tipo de parte ativa implementado, sendo que existem transformadores do tipo *core* e transformadores do tipo *shell*. Nos transformadores do tipo *core* os enrolamentos são construídos em torno do núcleo com uma forma cilíndrica e tipicamente o enrolamento de menor tensão é colocado no interior do de maior tensão de forma concêntrica. No caso de transformadores do tipo *shell* o núcleo envolve os enrolamentos, o que resulta numa forma oval dos enrolamentos em que os de tensão mais elevada são colocados lado a lado com os de menor tensão. Na figura 2.3 encontram-se uma representação destas duas configurações da parte ativa para um transformador trifásico, [20].

A escolha da utilização entre transformadores *core* e *shell* varia conforme a funcionalidade pretendida e também depende de fatores económicos. No âmbito desta dissertação os transformadores analisados estruturalmente serão essencialmente do tipo *core*.

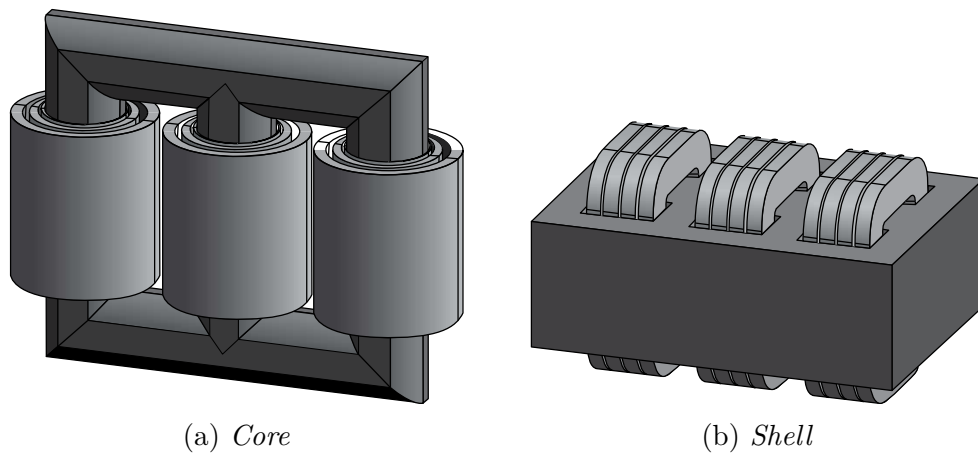


Figura 2.3: Tipos de construção da parte ativa, [20].

2.1.1 Componentes estruturais de um transformador *core*

Na secção anterior foi referida a parte ativa de um transformador que é constituída pelo núcleo e pelos enrolamentos e é responsável pela função principal do transformador. No entanto um transformador ainda é composto por outros componentes como representado na figura 2.4.

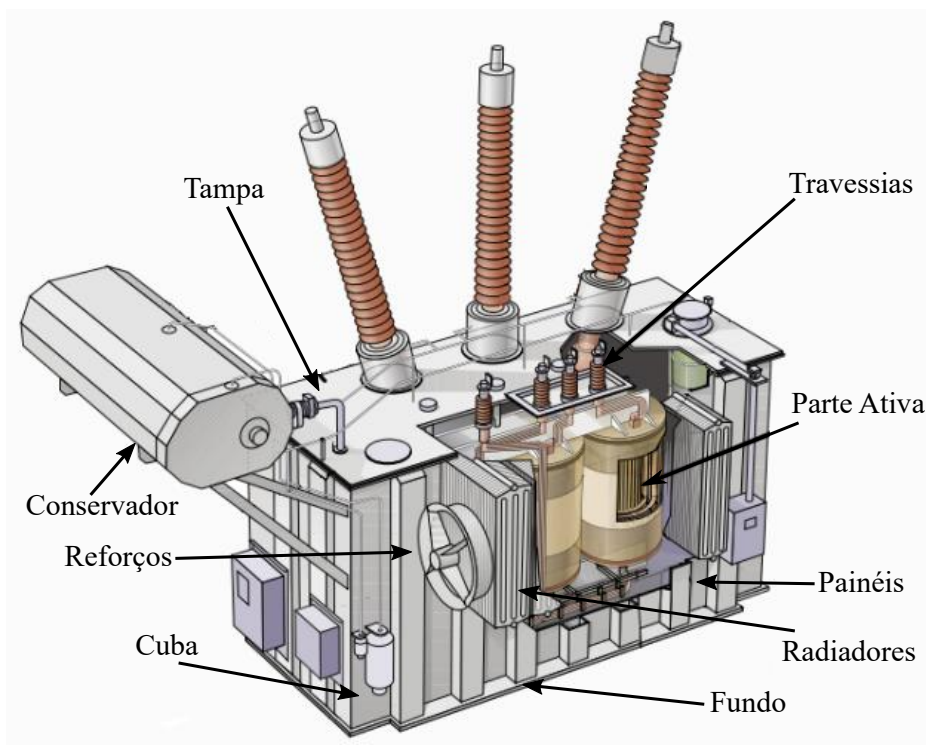


Figura 2.4: Principais componentes de um transformador de potência, (Adaptado de [4]).

Dos componentes indicados na figura 2.4, a cuba será o componente responsável por suportar os diferentes esforços mecânicos que o transformador irá estar sujeito ao longo da sua vida. A principal utilidade da cuba é armazenar o óleo que permite refrigerar a parte ativa assim como isolá-la eletricamente. No entanto, também tem como funcionalidades servir de proteção física contra o meio ambiente, suporte, fixação do transformador, transporte e ainda permite a inserção de acessórios, [13]

A cuba é essencialmente composta por três partes: fundo (cuba inferior), painéis (cuba superior) e ainda a tampa. Esta três partes poderão ser unidas por soldadura ou por parafusos, sendo que se os componentes forem aparafusados facilita o acesso caso ocorra algum problema durante o serviço do transformador, no entanto a soldadura permite uma melhor estanquidade de forma a que não existam fugas de óleo, [13].

Os painéis da cuba terão de suportar diversos tipos de solicitações, portanto de forma a aumentar a rigidez deste componente utilizam-se reforços. Os reforços, se forem colocados em locais estratégicos, permitem diminuir em grande parte a massa da cuba devido à diminuição da espessura necessária do painel, o que se traduz num custo económico deste componente bastante mais reduzido. O perfil de reforços mais utilizados são em barra, em T ou em caixa como representados na figura 2.5, [13].

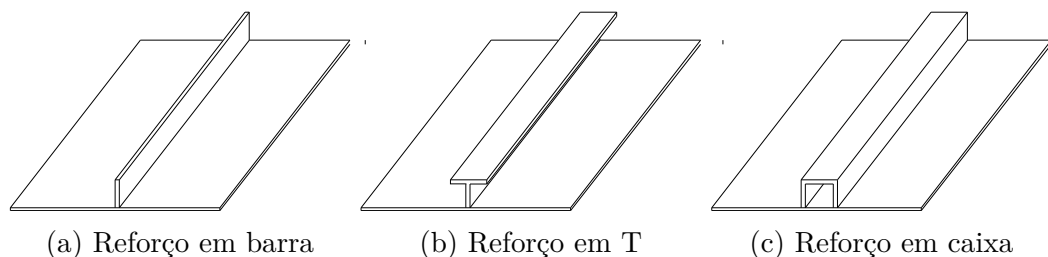


Figura 2.5: Tipos de reforços utilizados, (Adaptado de [13]).

Os reforços poderão ser verticais ou horizontais, no entanto, a direção que permite suportar melhor as solicitações no painel será a paralela ao menor lado, tipicamente o lado vertical. No âmbito desta dissertação somente se irá analisar reforços em barra.

Existem outros componentes pertencentes à cuba que são de interesse estrutural como os olhais de suspensão que permitem a elevação do transformador, os apoios para macaco, o suporte para o conservador ou os rodados que permitem fixação ao exterior. Estes acessórios têm frequentemente cálculos mecânicos associados a cada um devido à sua função num transformador e solicitações específicas, [13].

2.1.2 Análises estruturais de transformadores

Devido às diversas solicitações a que o transformador irá estar sujeito durante o seu ciclo de vida é necessário projetar os vários componentes de forma a suportar estes esforços e prever o comportamento mecânico da estrutura. A partir destas análises é possível dimensionar cada um dos constituintes com o objetivo de obter uma estrutura otimizada dependendo sempre do custo associado, especificações de clientes, especificações de fabrico, entre outras condicionantes. As solicitações presentes num transformador são as seguintes:

- **Elevação** - O transformador ao longo do seu fabrico, transporte e serviço terá de ser elevado por diversas razões. Esta elevação poderá ser realizada através dos olhais de suspensão, no topo da cuba, ou a partir dos apoios para macaco, no fundo da cuba. Do ponto de vista estrutural será importante considerar os esforços presentes neste processo nestes componentes e dimensioná-los de acordo com a massa que terão de suportar, [13].
- **Transporte** - O transporte do transformador poderá consistir em movimentos locais onde se contabiliza a massa do transformador ou grandes distâncias em que se considera que o transformador irá sofrer acelerações inerentes ao meio de transporte, [13].
- **Esforços sísmicos e do vento** - O meio ambiente onde o transformador irá ser instalado também poderá colocar a estrutura dos componentes em risco com cargas sísmicas fortes associadas a uma aceleração ou vento com velocidade alta que incida no transformador. Por estas razões é necessário dimensionar os componentes mecânicos de forma a suportar estes esforços, como por exemplo o suporte do conservador e radiadores, [13].
- **Sobrepresão** - Num transformador cheio de óleo por vezes ocorre falhas durante o seu serviço que resultam numa subida repentina da pressão do óleo. A cuba terá de ser projetada para este aumento de pressão mantendo-se sempre no regime elástico, visto que, uma rutura do tanque poderá implicar fugas de óleo com perigos para o ambiente ou até risco de incêndio, [13].
- **Curto-circuito** - O curto-circuito ocorre quando a corrente percorre um caminho não pretendido e quando isso acontece poderá descarregar grandes picos de força na estrutura de aperto no caso de transformadores *core*. Estes esforço é um dos mais destrutivos e poderá limitar o tempo de vida de um transformador, [13].

- **Coluna de óleo** - O fundo da cuba de um transformador terá de ser dimensionado para suportar a massa do óleo presente dentro da cuba, [13].
- **Vácuo** - Durante o processo de enchimento da cuba com óleo terá de ser retirado todo o ar presente dentro do transformador. Como consequência cria-se uma pressão negativa dentro da cuba de cerca de 0,1 MPa. Toda a cuba, e com especial atenção os painéis, terão de ser dimensionados para suportar esta pressão de forma a que o seu comportamento se mantenha dentro do regime elástico, [13].

No âmbito desta dissertação as análises estruturais realizadas serão para a situação em que existe um esforço devido ao vácuo visto que é do interesse estudar o comportamento dos painéis, dos reforços, assim como a cuba completa.

Devido à necessidade de dimensionamento e previsão do comportamento mecânico é necessário realizar análises estruturais que contabilizam as solicitações referidas. As análises estruturais realizadas poderão ser das seguintes formas:

- Métodos analíticos - como por exemplo recorrendo à Teoria da Elasticidade;
- Métodos numéricos - como por exemplo, o Método dos Elementos Finitos;
- Métodos empíricos - como por exemplo, o teste ao vácuo;
- Métodos híbridos - como por exemplo, recorrer a dados antigos para prever o comportamento, atribuindo hipóteses aproximadas.

Atualmente os métodos mais utilizados serão os analíticos e as análises por elementos finitos.

Os métodos analíticos permitem uma facilidade de implementação, no entanto o seu desenvolvimento para diferentes geometrias ou condições que não estejam presentes na literatura poderá ser muitas vezes complexo. Também se indica como desvantagem o facto destes métodos frequentemente sobredimensionarem os componentes de forma a garantir um bom funcionamento, como é o caso do dimensionamento de painéis reforçados.

Em contrapartida, têm cada vez mais sido implementadas as análises por elementos finitos que permitem verificar o comportamento mecânico de geometrias mais complexas com uma boa aproximação à realidade. Estas análises recorrem à discretização da geometria em elementos menores que após estabelecida a relação entre cada elemento e definido o tipo de comportamento permite resolver equações diferenciais parciais e assim obter os resultados para a estrutura completa. O desenvolvimento

e grande implementação destas análises devem-se em grande parte à crescente capacidade computacional dos dias de hoje que permite desenvolver programas que eficientemente realizam cálculos estruturais. Os *softwares* de análises de elementos finitos têm portanto um papel fundamental tanto a nível académico como industrial. No entanto, o cálculo por elementos finitos utilizando estes *softwares* também terá as suas desvantagens visto que poderá resultar numa exigência computacional mais elevada, que logicamente terá como consequência uma análise demorada. A sua implementação industrial também enfrenta alguns obstáculos visto que as vantagens de um dimensionamento mais otimizado utilizando os métodos de elementos finitos são muitas vezes ultrapassadas pelo conhecimento empírico e analítico que permite um dimensionamento de forma autónoma e com maior rapidez. Ainda se indica o facto de o uso destes *softwares* obrigar frequentemente a um licenciamento com custos bastante elevados o que desencoraja a sua utilização. Exemplos de *softwares* de elementos finitos utilizados para cálculo estrutural são o ANSYS, ABAQUS ou Nastran.

Como resposta aos *softwares* de elementos finitos comerciais existem bastantes alternativas sobre licenças gratuitas, também conhecidos como *softwares* de código aberto ou *open source*. Um exemplo deste *software* é o Salome-Meca que permite pré-processamento, cálculo e pós-processamento dos modelos. No âmbito desta dissertação irá-se utilizar este *software* para realizar as análises estruturais e estudar a sua implementação.

2.1.3 Método dos Elementos Finitos

O Método dos Elementos Finitos é um procedimento numérico para obter soluções aproximadas através de equações diferenciais. É bastante utilizado em toda a engenharia por resolver problemas físicos de alguma complexidade evitando assim análise analíticas que só é possível em alguns casos particulares.[15]

Este método consiste em dividir o domínio do problema em subdomínios, denominados de elementos finitos, que por sua vez são constituídos por nós que permitem a continuidade dos elementos. O conjunto dos elementos finitos de um domínio tem o nome de malha. Após esta discretização, cada elemento é caracterizado pelas equações diferenciais do problema, que são resultado de aproximações das equações de equilíbrio. Por fim, procede-se à montagem de todas as equações obtidas que podem ser resolvidas para calcular as variáveis em estudo.[15]

2.2 Salome-Meca/Code_Aster

O Salome-Meca e o Code_Aster são *softwares* que permitem análises por elementos finitos e são distribuídos sobre a *GNU General Public License*, que significa que são livres de utilização, partilha e modificação, assim como terá de estar disponível para visualização e cópia todo o código fonte do *software*. Estes tipos de programa também são designados de código aberto ou *open source* visto que todo o conteúdo do desenvolvimento do programa encontra-se disponível para o utilizador comum, [2].

O *software* Code_Aster, em que “Aster” significa *Analyses des Structures et Thermomécanique pour des Études et des Recherches* (Análise Estrutural e Termomecânica para Estudos e Pesquisas) é um programa de cálculo mecânico. Este programa começou a ser desenvolvido pela EDF (*Électricité de France*) em 1989 devido à necessidade desta companhia realizar cálculo estrutural de reatores nucleares. Em 2001 o Code_Aster começou a ser distribuído como um *software* livre e inicia a sua partilha nível global. Com este pensamento, o *software* foi capaz de evoluir ao longo dos anos devido a várias contribuições da comunidade de utilizadores, seja com código, com casos de estudo ou ainda com a vasta documentação acessível de todas as componentes deste programa, [8].

O Code_Aster só desempenha a função de cálculo (*solver*), ou seja, este programa lê um modelo que se encontre já discretizado e realiza o cálculo estrutural que permite obter os resultados. No entanto, numa típica análise por elementos finitos são necessárias ferramentas de pré-processamento e pós-processamento. O pré-processamento permite editar ou criar a geometria para além de ser responsável pela discretização do modelo geométrico numa malha de elementos finitos. Já a ferramenta de pós-processamento tem como objetivo mostrar os resultados obtidos num ambiente gráfico para além de ser possível tratar os dados após o seu cálculo. De forma a implementar estes processos a EDF, a CEA/DEN e a OPEN CASCADE desenvolveram o *software* SALOME com um objetivo de criar um ambiente gráfico para simulações numéricas, [7].

No âmbito destes dois *softwares*, foi desenvolvido o Salome-Meca, que integra as funcionalidades de pré e pós-processamento do SALOME, com o cálculo por elementos finitos do Code_Aster. Este programa é distribuído por várias plataformas incluindo pela SimulEase que desenvolveu uma versão livre que permite ser utilizada no sistema operativo Windows 10. Na tabela 2.1 encontram-se as versões integradas dos dois *softwares* referidos no Salome-Meca 2018 para o Windows:

Tabela 2.1: Especificações dos *softwares* integrados no Salome-Meca 2018 para o Windows, [3]

Code_Aster	<i>stable</i> 13.6
SALOME	8.3.0
ParaView	5.1.2

O Salome-Meca será o programa utilizado no âmbito desta dissertação para realizar as análises estruturais. Este programa oferece vários tipos de soluções, em que é permitido criar e editar geometria no módulo específico e ainda a malha de elementos finitos. O cálculo é realizado no módulo AsterStudy que funciona como um ambiente gráfico para o *software* Code_Aster e permite realizar vários tipos de análises. Como ferramenta de pós-processamento o Salome-Meca oferece uma integração do programa ParaView (designada como ParaViS no *software*), desenvolvido originalmente pela Kitware Inc.. Este programa é desenvolvido em várias linguagens de programação tendo contribuições em Python, Fortran e C++ e é compatível com a utilização de rotinas Python dentro de cada módulo que permite realizar cada processo de forma automática, [28].

2.2.1 Módulo geometria

O módulo de geometria é o módulo responsável pela criação do modelo geométrico e edição. Esta componente permite criar simples objetos geométricos como vértices, linhas, faces, assim como integra várias funções de edição de geometria típicas de uma ferramenta de desenho. Também é possível a utilização da consola Python integrada neste módulo para criar estes modelos a partir de funções próprias que só podem ser acedidas neste ambiente e assim realizar desenho automático, [28].

Este módulo é indicado como uma das principais desvantagens do uso deste *software* devido à sua pouca versatilidade quando comparados com outras ferramentas de desenho que limita bastante a edição de geometria. Em versões recentes do *software* SALOME, este módulo já se encontra substituído por outro com o mesmo objetivo mas com uma estrutura bastante diferente, no entanto, a versão utilizada no decorrer desta dissertação não inclui estas novas funcionalidades. Na figura 2.6 é possível visualizar o ambiente gráfico deste módulo.

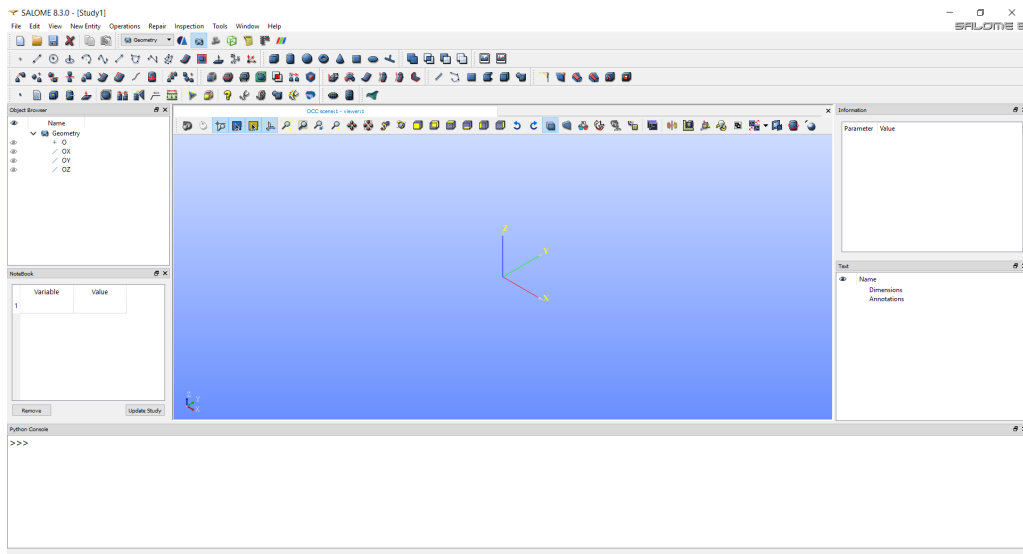


Figura 2.6: Ambiente gráfico do módulo geometria.

2.2.2 Módulo malha

O módulo relativo à malha tem como principal objetivo criar a discretização do modelo geométrico em elementos finitos sendo que existem várias funções que permitem realizar este processo de acordo com as intenções do utilizador. A discretização poderá ser segundo elementos de uma dimensão (segmentos), elementos de duas dimensões (triangulares ou quadrangulares) e ainda elementos de três dimensões (tetraedros, hexaedros, entre outros). O número de nós por elemento também poderá ser alterado sendo que o *software* possibilita a criação de elementos lineares, quadráticos ou bi-quadráticos, segundo as suas funções de forma, [28].

A criação da malha de elementos finitos é realizada através de hipóteses e algoritmos, sendo que as algoritmos definem o tipo de elemento que se pretende utilizar enquanto as hipóteses definem os parâmetros do elemento que como consequência influenciam a discretização. Exemplos de hipóteses para os elementos de uma dimensão são o tamanho de elemento, número de segmentos ou ainda implementar uma progressão geométrica que permite refinar a malha gradualmente segundo uma aresta. No caso de duas dimensões e três dimensões existem hipóteses que permitem limitar a área e o volume de cada elemento respetivamente, [28].

Este módulo à semelhança do de geometria possibilita a implementação de rotinas em Python para realizar a criação da malha e executar as funções também disponíveis no ambiente gráfico. O ambiente gráfico encontra-se representado na figura 2.7.

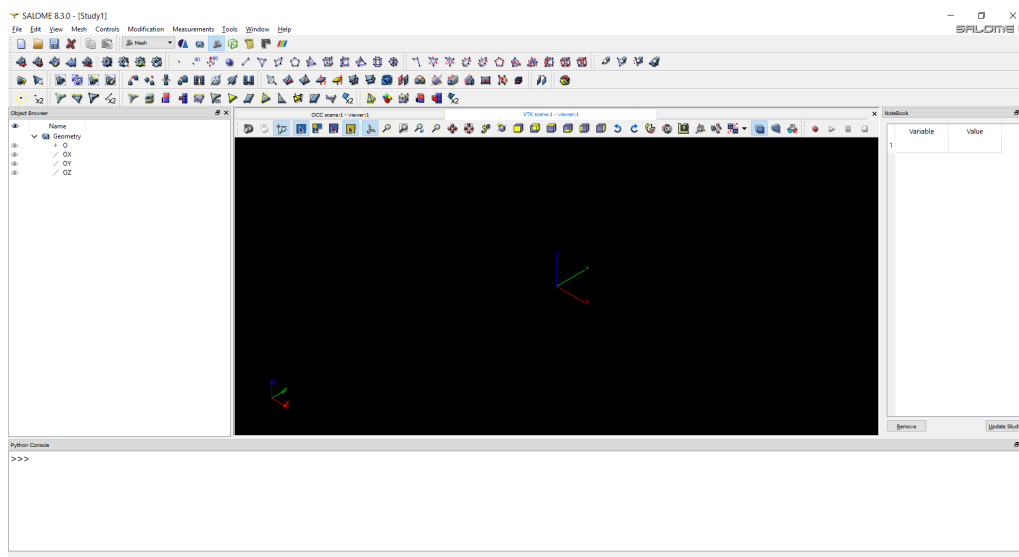


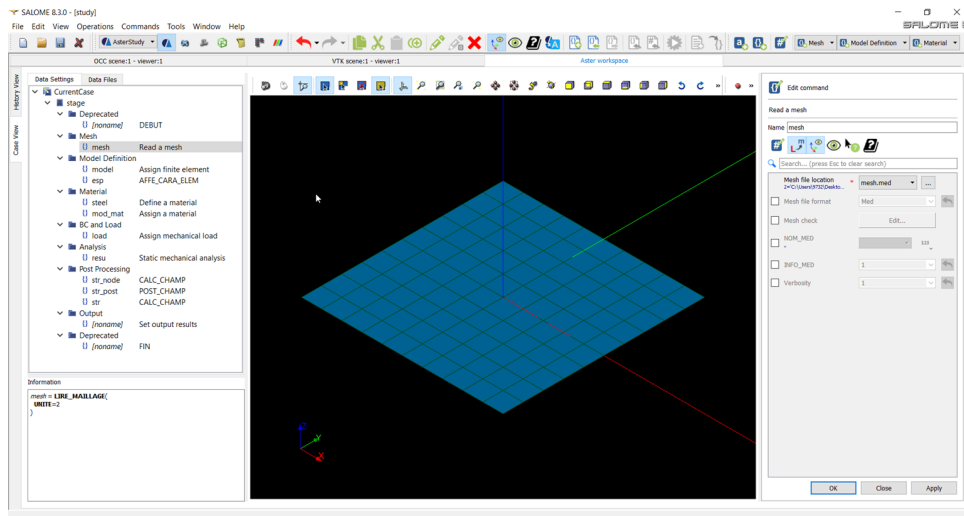
Figura 2.7: Ambiente gráfico do módulo malha.

2.2.3 Code_Aster/AsterStudy

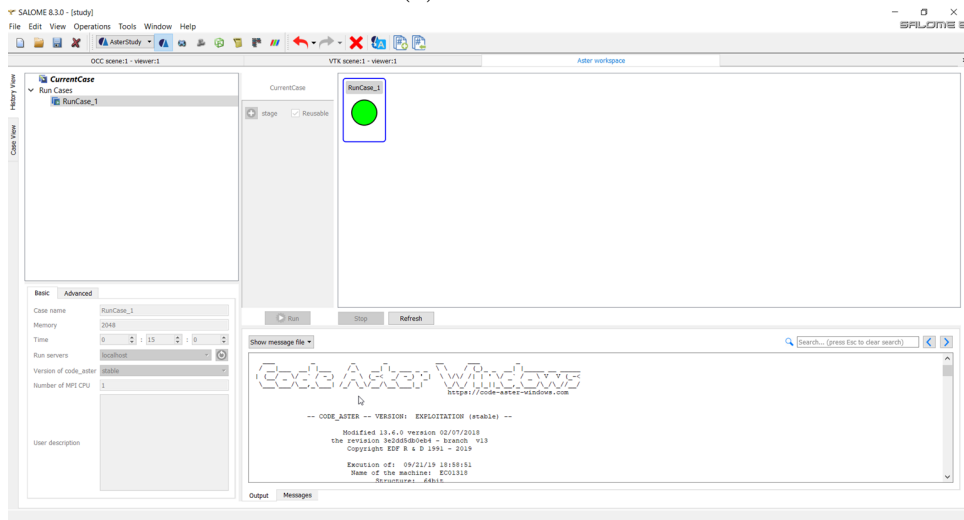
O AsterStudy é o módulo do *software* Salome-Meca responsável pela integração do Code_Aster no ambiente gráfico do programa. Este módulo permite a criação de funções da linguagem utilizada pelo Code_Aster para realizar as análises de forma interativa. Estas funções em alternativa também poderão ser escritas usando o modo texto, [28].

O AsterStudy permite assim estudar fenómenos mecânicos, térmicos e acústicos utilizando diversas modelações de elementos finitos. Assim que estiver estabelecida toda a rotina de análise, é possível correr o o cálculo que no final indica se foi bem-sucedido ou pelo contrário, sendo que os erros estarão presentes num relatório produzido por este módulo. Se assim for indicado na própria rotina, também é possível exportar os resultados ou para ficheiros de texto ou para visualização, [28].

O ambiente gráfico deste módulo encontra-se dividido em duas janelas. Na *Case View* é possível editar e criar o algoritmo de cálculo e todos os conceitos associados. A segunda janela, *History View* é relativa à execução do estudo, definição de alguns parâmetros da execução e visualização das mensagens de erro. Na figura 2.8 é possível visualizar o ambiente gráfico deste módulo.



(a) Case View



(b) History View

Figura 2.8: Ambiente gráfico módulo AsterStudy.

2.2.4 ParaView

O módulo ParaView (ou ParaViS) permite ao utilizador visualizar os campos de resultados obtidos após o cálculo no AsterStudy. Nesta ferramenta é possível editar a visualização, como alterar o mapa de cores utilizado para representação dos resultados, visualizar os efeitos calculados ao longo do tempo e ainda mostrar um modelo deformado. Também poderá ser do interesse do utilizador obter resultados pelas coordenadas geométricas ou ao longo de uma aresta e de seguida exportar os resultados para uma tabela. Assim, este módulo de pós-processamento facilita a compreensão dos resultados calculados, [28]. O ambiente gráfico do ParaView encontra-se na figura 2.9

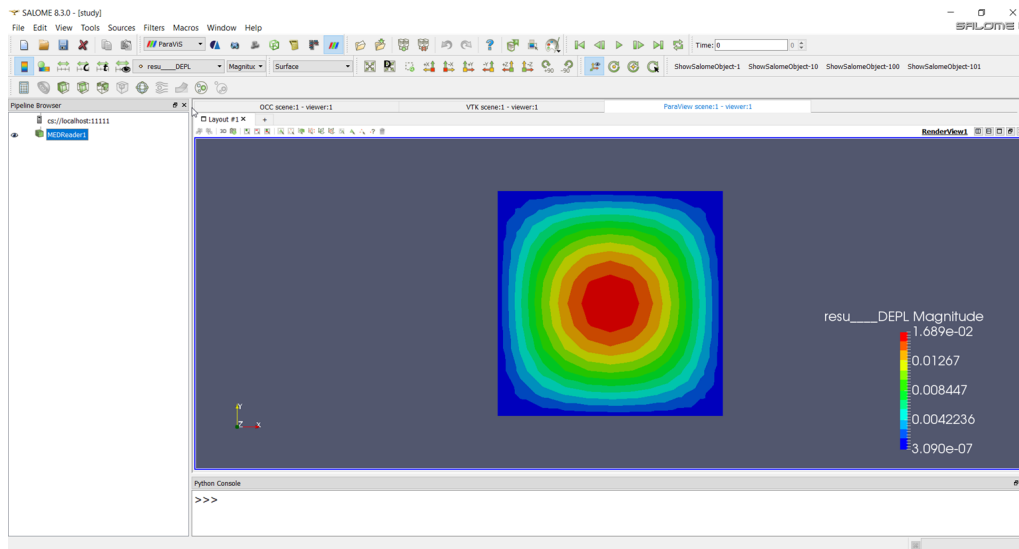


Figura 2.9: Ambiente gráfico do módulo ParaView.

2.3 Formulações Code_Aster

O Code_Aster oferece bastantes soluções para simular modelos numéricos utilizando o método do elementos finitos. Neste *software* é possível realizar análises térmicas, estáticas ou acústicas. Os elementos explorados nesta dissertação utilizando o Code_Aster foram elementos em estado plano tensão, elementos de casca e elementos de placa. O algoritmo implementado no Code_Aster utiliza a definição de conceitos. Exemplos de conceitos são o material do modelo, a espessura, a modelação, as solicitações ou ainda as condições de fronteira. De seguida, encontra-se uma breve descrição das modelações utilizadas no âmbito desta dissertação.

2.3.1 Estado plano tensão

A modelação estado plano tensão é referida no *software* Code_Aster como `C_PLAN`. Esta modelação possibilita a utilização de elementos finitos triangulares com três nós cada elemento (`TRIA3`) e com seis nós cada elemento (`TRIA6`) e ainda quadrangulares com quatro nós cada elemento (`QUAD4`), com oito nós cada elemento (`QUAD8`) e ainda com nove nós cada elemento (`QUAD9`). Em relação aos graus de liberdade de cada nó, a modelação `C_PLAN` define que cada nó só se poderá movimentar segundo a direção horizontal ou vertical no plano de cada elemento, portanto, cada nó terá dois graus de liberdade (`DX` e `DY`), [21].

Esta modelação permite definir zonas de contacto assim como solicitações com deformação inicial, força segundo uma aresta, forças volúmicas, gravidade, pressão e rotação, [21].

2.3.2 Teoria das Placas

A modelação que usa a Teoria das Placas é referida no Code_Aster como DKT. Estes elementos são usados quando a espessura da estrutura é bastante menor que as suas dimensões e não contabilizam a curvatura da geometria. A geometria é criada segundo superfícies sendo a espessura atribuída pelo próprio algoritmo,[29]

Este algoritmo só possibilita a utilização de elementos lineares, isto é, elementos triangulares de três nós e elementos quadrangulares de quatro nós. Em relação aos graus de liberdade, em cada nó existe os três graus de liberdade de translação e os três graus de liberdade de rotação, [29].

A integração utilizada na modelação DKT é a integração de Hammer com três pontos no caso dos elementos triangulares e integração de Gauss com quatro pontos no caso de elementos quadrangulares, [29].

2.3.3 Teoria das Cascas

Os elementos de casca utilizados no Code_Aster têm uma formulação semelhante aos elementos de placa no entanto contabilizam a curvatura da geometria e esforços de membrana. Neste caso o Code_Aster limita a sua utilização somente a elementos triangulares com sete nós (TRIA7) ou elementos quadrangulares com nove nós (QUAD9). Esta discretização também é realizada utilizando elementos Heterosis que evita o fenómeno de *shear locking* que pode acontecer quando a espessura da casca é bastante menor que o raio de da curvatura. Estes elementos são caracterizados pelos nós da fronteira possuírem os três graus de liberdade de translação e rotação, enquanto o nó do meio do elemento só possui graus de liberdade de rotação, [24]

A integração utilizada nesta modelação é a integração de Hammer com sete pontos no caso dos elementos triangulares e integração de Gauss com nove pontos no caso de elementos quadrangulares. Ainda se refere que este algoritmo utiliza integração reduzida seletiva para também prevenir os efeitos de *locking*. Esta integração, ao contrário da integração reduzida uniforme, utiliza a integração de Gauss adequada para o cálculo da matriz de rigidez, no entanto, reduz o grau utilizado para o cálculo das deformações e tensões, [24].

Esta página foi intencionalmente deixada em branco.

Capítulo 3

Placa com Furo Central

Neste capítulo é apresentada a formulação do caso de estudo da placa com um furo central. É do interesse desta secção apresentar a abordagem teórica da análise de tensões, apresentar a implementação do modelo no *software* Salome-Meca e por fim comparar e analisar os resultados obtidos.

3.1 Fundamentos Teóricos

A placa de furo central tem sido uma problemática bastante estudada, por isso naturalmente encontra-se várias soluções em bibliografia especializada. A geometria considerada no âmbito deste capítulo encontra-se representada na figura 3.1.

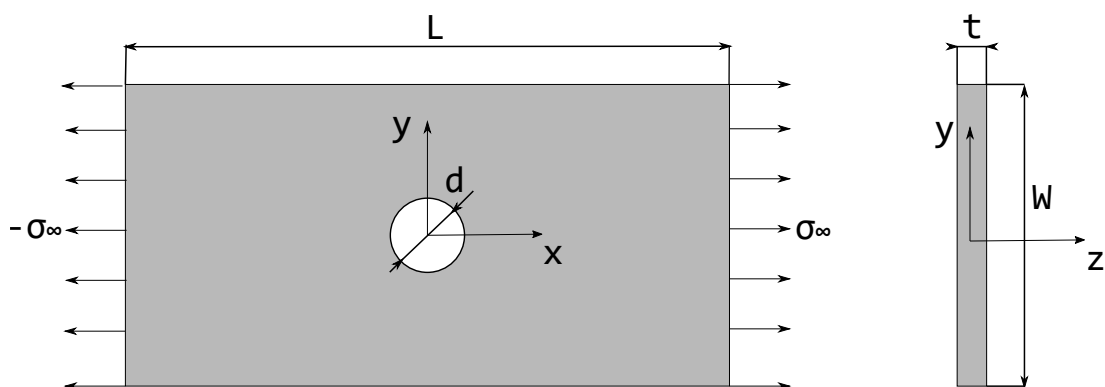


Figura 3.1: Dimensões de uma placa semi-infinita.

No modelo abordado irão ser admitidas sollicitações uni-axiais nas extremidades da placa, que por sua vez refletem-se numa concentração de tensões na vizinhança do furo. Esta sollicitação total é representada como σ_{∞} , contudo, como o objetivo

é estudar a secção da placa que contém o furo, será de maior interesse considerar uma sollicitação que contabiliza a redução da área nesta zona. Assim sendo, define-se sollicitação nominal (σ_n) como a relação entre a carga aplicada na secção que contém o furo, e relaciona-se com a sollicitação total como demonstrado na equação 3.1, [14].

$$F = \sigma_\infty W t = \sigma_n (W - d) t \rightarrow \sigma_\infty = \left(1 - \frac{d}{W}\right) \sigma_n \quad (3.1)$$

Devido à ocorrência frequente e importância da razão entre o diâmetro do furo e o comprimento da placa na direção normal à sollicitação, no caso em estudo corresponde à largura, este coeficiente futuramente irá ser referido como γ_n ($\gamma_n = d/W$). Logo a equação 3.1 pode ser reescrita, como se verifica em 3.2.

$$\sigma_\infty = (1 - \gamma_n) \sigma_n \quad (3.2)$$

A nível de geometria também existem algumas simplificações que podem ser aplicadas. O exemplo mais comum é a análise somente de um quarto da placa visto existir simetria em relação aos dois eixos, nesse sentido reescreve-se as dimensões apresentadas como:

$$w = \frac{W}{2} \quad ; \quad l = \frac{L}{2} \quad ; \quad r = \frac{d}{2} \quad (3.3)$$

Por fim, é importante referir que as soluções apresentadas serão para o modelo de uma placa semi-infinita que é caracterizado pelas seguintes considerações:

- $L \gg W$;
- $W \gg t$ (placa fina).

3.1.1 Fator de concentração de tensão

Como já foi referido, quando aplicada uma sollicitação na extremidade da placa, terá como consequência o aparecimento de uma tensão mais elevada na zona da vizinhança do furo. A proporção entre a tensão máxima observada (σ_{max}) e a sollicitação aplicada é definida como o fator de concentração de tensão. No entanto, podemos considerar as duas sollicitações, a nominal ou a total, que resultam respetivamente, em um fator de concentração de tensão nominal (K_{tn}) e em um fator de concentração de tensão total (K_{tg}). Estes fatores são apresentados nas equações 3.4 e 3.5, [14].

$$K_{tg} = \frac{\sigma_{max}}{\sigma_{\infty}} \quad (3.4)$$

$$K_{tn} = \frac{\sigma_{max}}{\sigma_n} \quad (3.5)$$

Os fatores podem ser transformados um no outro a partir da relação entre as áreas de aplicação, à semelhança da relação entre as solicitações, como demonstra a seguinte equação:

$$K_{tn} = (1 - \gamma_n) K_{tg} \quad (3.6)$$

No desenvolvimento das soluções apresentadas irá ser utilizado o conceito de fator de concentração de tensão total, K_{tg} , visto contabilizar efetivamente a carga que se aplica. Uma solução analítica foi desenvolvida por Howland em 1929, [11], mas devido ao grau de complexidade inerente, vários autores apresentaram fórmulas de aproximação com desvios relativos bastante baixos. Uma aproximação apresentada por Pilkey, com diferenças menores que 1%, para γ_n menor que 0,5, é a equação 3.7, [14, 11].

$$K_{tg} = \frac{2,000 + 0,284(1 - \gamma_n) - 0,600(1 - \gamma_n)^2 + 1,320(1 - \gamma_n)^3}{1 - \gamma_n} \quad (3.7)$$

Na figura 3.2 é possível verificar a variação do fator K_{tg} com γ_n e ainda as soluções analíticas para cinco pontos que Howland apresentou.

Quando as dimensões da placa são ambas bastante maiores que o diâmetro do furo, portanto, quando γ_n tende para 0, verifica-se que K_{tg} equivale a 3. Dessa forma, conclui-se facilmente que no caso de uma placa fina infinita em ambas as direções a tensão máxima irá ser o triplo da tensão aplicada. Por outro lado, quando γ_n tende para 1, K_{tg} aumenta exponencialmente, já que a área nominal será infinitesimal, logo a tensão máxima será uma ordem de grandeza bastante maior do que a aplicada. No entanto, do ponto de vista construtivo esta solução não é considerada já que σ_{max} atinge valores irrealistas do ponto de vista prático, [14].

3.1.2 Distribuição da tensão

Será do interesse do presente trabalho analisar a distribuição da tensão ao longo da placa. Para este efeito considerou-se as soluções encontradas para o modelo de uma placa infinita com furo central, ou seja, quando γ_n tende para zero. Acontece

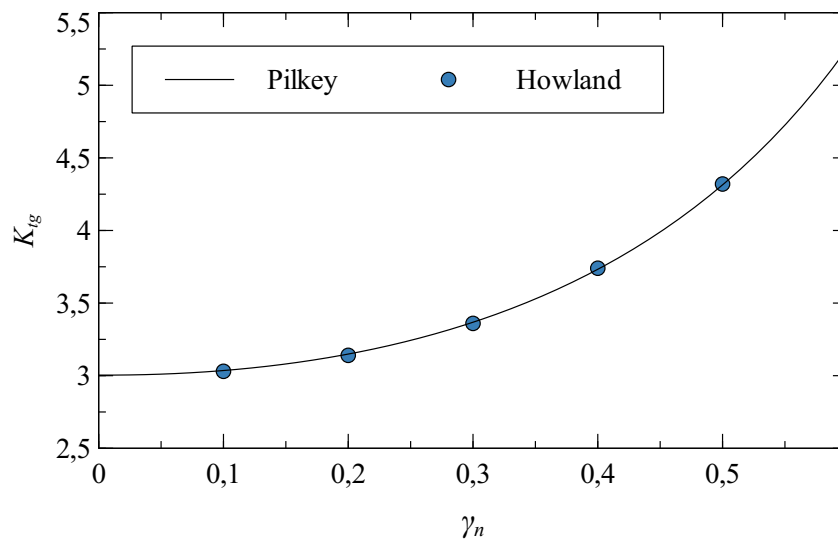


Figura 3.2: Comparação entre a função de aproximação apresentado por Pilkey e os valores obtidos por Howland, [14, 11].

que a tensão máxima será diferente da situação da placa semi-infinita, como já foi observado pelas diferenças do fator de concentração de tensão. Contudo importa observar se a distribuição segue o mesmo padrão ao longo da secção a analisar. Neste estudo considerou-se a distribuição das tensões radiais, circunferenciais e de corte, em função das coordenadas polares. Este referencial pode ser observado na figura 3.3, [16].

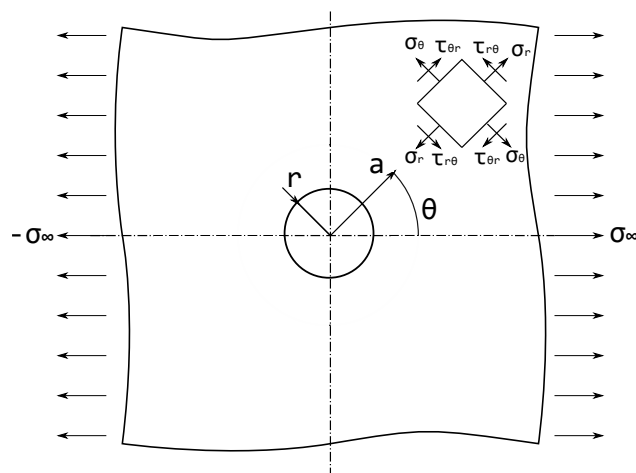


Figura 3.3: Referencial em coordenadas polares, (Adaptado de [14]).

As equações que traduzem o estado de tensão da placa podem ser observadas a partir do sistema de equações 3.8, [16].

$$\begin{aligned}\sigma_r &= \frac{1}{2} \sigma_\infty \left(1 - \frac{r^2}{a^2}\right) + \frac{1}{2} \sigma_\infty \left(1 - \frac{4r^2}{a^2} + \frac{3r^4}{a^4}\right) \cos 2\theta \\ \sigma_\theta &= \frac{1}{2} \sigma_\infty \left(1 + \frac{r^2}{a^2}\right) - \frac{1}{2} \sigma_\infty \left(1 + \frac{3r^4}{a^4}\right) \cos 2\theta \\ \tau_{r\theta} &= -\frac{1}{2} \sigma_\infty \left(1 + \frac{2r^2}{a^2} - \frac{3r^4}{a^4}\right) \sin 2\theta\end{aligned}\quad (3.8)$$

Contudo, como facilmente se observa a zona de maior interesse será a correspondente a $\theta = \pi/2$ (ou $\theta = 3\pi/2$). Portanto, é possível reescrever o sistema 3.8 como:

$$\begin{aligned}\sigma_r &= \frac{3}{2} \sigma_\infty \left(\frac{r^2}{a^2} - \frac{r^4}{a^4}\right) \\ \sigma_\theta &= \frac{1}{2} \sigma_\infty \left(2 + \frac{r^2}{a^2} + \frac{3r^4}{a^4}\right) \\ \tau_{r\theta} &= 0\end{aligned}\quad (3.9)$$

Por sua vez, observa-se que a componente da tensão circunferencial será relativamente mais elevada em módulo do que a tensão radial. Sendo esta tensão máxima quando $a = r$, ou seja, será três vezes a carga aplicada (tensão remota), como já se referiu. No entanto, à medida que a coordenada a aumenta a tensão começa a transformar-se numa função uniforme e a convergir para o valor de σ_∞ , sendo que, no caso $a = 3r$ a tensão circunferencial será igual a $1,07 \sigma_\infty$. Em secções posteriores, será este o limite para o estudo da distribuição, [16].

3.2 Aplicação

De modo a poder implementar o modelo no *software* de análise de elementos finitos foi necessário estabelecer um exemplo e as suas propriedades. Na tabela 3.1 é possível verificar as propriedades geométricas, propriedades do material assim como a carga aplicada. Visto que somente será analisada um quarto da placa, as dimensões são referentes a este caso, segundo as variáveis definidas na equação 3.3.

A partir dos conceitos abordados na secção 3.1 estabelece-se a tensão máxima e o fator de concentração de tensões, cujos os valores se encontram na tabela 3.2.

Tabela 3.1: Propriedades de um exemplo da placa semi-infinita com furo central

Geometria		Material	
l [mm]	40	E [GPa]	210
w [mm]	20	ν	0,3
r [mm]	5	Solicitação	
t [mm]	4	σ_∞ [MPa]	50

Com estes valores também é possível reescrever a equação para a distribuição da tensão circunferencial explícita em 3.9 para o caso de uma placa infinita com as propriedades que se definiriam. Esta substituição é demonstrada em 3.10.

$$\begin{aligned} \sigma_\theta &= \frac{1}{2}\sigma_\infty \left(2 + \left(\frac{\gamma_n}{a/w} \right)^2 + 3 \left(\frac{\gamma_n}{a/w} \right)^4 \right) \\ &= 50 + 25 \left(\frac{0,250}{a/w} \right)^2 + 75 \left(\frac{0,250}{a/w} \right)^4 \end{aligned} \quad (3.10)$$

3.3 Implementação no Salome-Meca

Geometria

Em primeiro lugar foi necessário criar uma geometria que representasse o caso em estudo com as dimensões definidas anteriormente na tabela 3.1. As simplificações já abordadas e agora implementadas no *software* consistem na consideração de estado plano de tensão devido a uma espessura desprezável em relação às outras dimensões e ainda a utilização de um quarto da placa devido às simetrias axiais. Na figura 3.4 é possível observar a geometria criada no *software*.

De salientar que foram estabelecidas algumas divisões bem definidas que correspondem à vizinhança do furo, mais precisamente uma área geométrica quadrangular com uma distância mínima entre cada lado e o centro do furo de $3r$. Esta vizinhança foi estabelecida porque será a zona de maior interesse nesta análise e permitirá assim realizar um refinamento da malha de elementos finitos só nesta área. Por outro lado, este tipo de construção permite usar elementos quadriláteros em vez de triangulares,

Tabela 3.2: Valores teóricos da tensão máxima e fator de concentração de tensão

γ_n	0,250
K_{tg}	3,243
σ_{max} [MPa]	162,158

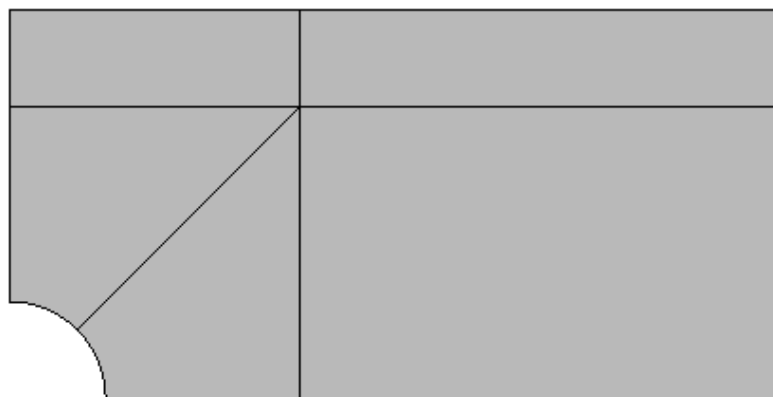


Figura 3.4: Geometria implementada no *software*.

que será do interesse da análise visto representar resultados mais precisos e estáveis, [5].

Malha

Para uma primeira abordagem para a discretização deste geometria definiu-se o número de divisões de cada fronteira e assim estabeleceu-se um número de elementos, cujo conjunto deles representa uma malha de elementos finitos. Uma representação desta malha para uma das primeiras abordagens com um menor número de divisões, pode ser observada na figura 3.5.

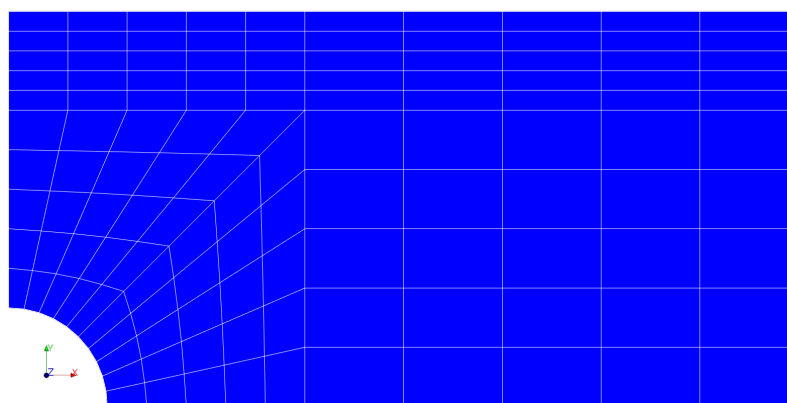


Figura 3.5: Malha de elementos finitos com 5 elementos por fronteira.

A malha de elementos tem uma importância grande na resolução de problemas com elementos finitos já que o número de elementos e até o tipo de elementos

escolhido irão ter influência nos resultados obtidos. Logo, será do interesse deste trabalho analisar qual a tipologia da malha que oferece maior garantia de resultados sem resultar numa exigência de processamento incomportável.

Processamento

Como já abordado, o *software* Salome-Meca utiliza a linguagem Code_Aster para processar os resultados. Cada modelo é composto por um conjunto de funções e cada uma exige certas variáveis de *input*, que permite ao utilizador obter os resultados que pretende.

Para o exemplo estudado utilizou-se uma modelização para Estado Plano Tensão devido à dimensão maior da placa ser um décimo da espessura. No âmbito da linguagem do Code_Aster esta modelação corresponde à função C_PLAN. Este algoritmo de cálculo não considera a espessura da placa, portanto, só apresenta resultados para as duas dimensões, [21].

Ainda no âmbito das funções de Code_Aster utilizadas, foram estabelecidos os seguintes passos:

- Ler a malha de elementos;
- Orientar a normal de cada elemento na mesma direção;
- Atribuir uma formulação/lei constitutiva à malha de elementos;
- Definir o material utilizado;
- Atribuir o material ao modelo;
- Definir solicitações e condições de fronteira;
- Realizar análise estática.

Para esta análise foram utilizadas as condições de fronteira resultantes da simetria do problema, logo os deslocamentos em x (dx) são nulos ao longo da aresta correspondente à reta $x = 0$ e de igual forma, os deslocamentos em y são nulos (dy) ao longo da aresta correspondente à reta $y = 0$, [31].

A partir deste ponto, o algoritmo já é capaz de calcular os deslocamentos em cada nó, que como cada um terá dois graus de liberdade, as variáveis que irão dar saída serão os deslocamentos segundo a direção x e os deslocamentos segundo a direção y .

Pós-processamento

Em qualquer análise de elementos finitos é importante realizar um tratamento dos dados após o cálculo das variáveis. Este processo pode ter vários objetivos, sendo os mais comuns o cálculo de outros resultados, como o campo de tensões, o armazenamento dos resultados num formato facilmente acessível e por fim, a capacidade de visualização dos resultados em ambiente gráfico. Para o cálculo das tensões e por sua vez, o armazenamento destes resultados, o Code_Aster fornece funções para este efeito, sendo que as tensões em cada ponto de Gauss são obtidas através dos deslocamentos e de seguida extrapoladas para cada nó. Por fim, é possível visualizar em ambiente gráfico tanto os deslocamentos como as tensões através da ferramenta ParaView. O campo de tensões correspondente a σ_x para o exemplo de aplicação em estudo pode ser observado na figura 3.6, [30, 32].



Figura 3.6: Representação gráfica do pós-processamento das tensões segundo x .

Como é possível observar, a consideração imposta ao início prevendo que somente existirá uma diferença de resultados significativa na zona da vizinhança do furo, encontra-se correta. Sendo que a tensão máxima será no nó correspondente a $\theta = \pi$ e $a = r$ como referido na secção 3.1.2.

3.3.1 Análise da malha de elementos

Após a realização do modelo para a placa de furo central no Code_Aster, foi do interesse deste trabalho realizar uma análise dos resultados e por sua vez da influência do número de elementos. Para este efeito foram realizados oito testes com diferentes números de divisões da geometria. No entanto, define-se cada teste pelo tamanho de cada elemento presente na fronteira segundo $\theta = \pi$ e $a \leq 3r$, visto que será a zona crítica logo onde se pretende obter os resultados mais precisos. Por

simplicidade esta fronteira no futuro irá ser referida como A_1 e equivale a todos os nós entre $a = r$ e $a = 3r$ segundo a direção $\theta = \pi$.

No âmbito destes testes foram utilizados os elementos quadriláteros com quatro nós cada, sendo que na linguagem do *software* são referidos como QUAD4.

Todos os testes encontram-se submetidos às mesmas condições só alterando o tamanho dos elementos, por consequência este fator leva a uma alteração do campo de tensões. A partir do pós-processamento inerente ao Salome-Meca foi possível obter o σ_{max} de cada teste, sendo que a partir da equação 3.4 calculou-se $\overline{K_{tg}}$. Para cada teste utilizou-se como referência o K_{tg} presente na tabela 3.2 e para esse efeito foi calculado uma diferença analítica relativa. Esta relação é dada por:

$$dif(\%) = \frac{|K_{tg} - \overline{K_{tg}}|}{K_{tg}} \cdot 100 \quad (3.11)$$

Na equação 3.11, $dif(\%)$ a diferença analítica em percentagem, $\overline{K_{tg}}$ o valor obtido de cada teste, e K_{tg} o valor de referência presente na tabela 3.2. Os resultados obtidos para cada teste e as suas características são demonstrados na tabela 3.3.

Tabela 3.3: Resultados obtidos após refinamento da malha

Teste	Nº elementos (A1)	Tamanho (A1) [mm]	σ_{max} [MPa]	$\overline{K_{tg}}$	$dif(\%)$
1	10	1,000	164,422	3,288	1,388
2	15	0,667	165,285	3,306	1,943
3	30	0,333	164,780	3,296	1,634
4	50	0,200	164,148	3,283	1,233
5	100	0,100	163,491	3,270	0,833
6	150	0,067	163,235	3,265	0,678
7	200	0,050	163,099	3,262	0,586
8	400	0,025	162,886	3,258	0,463

Como se pode observar houve uma diminuição relativa da diferença comparando com o valor de referência à medida que se aumenta o número de elementos segundo A_1 . No entanto, esta relação não é proporcional, e é perceptível a partir da figura 3.7.

Ao analisar o gráfico 3.7 destaca-se o primeiro teste, que corresponde à utilização de 10 elementos, por ser um valor mais próximo do teórico do que alguns testes com maior número de divisões e este efeito pode resultar numa falsa interpretação. No entanto o que acontece é que devido a um menor número de elementos a distribuição da tensão não se encontra completamente definida, portanto este valor é influenciado por nós na vizinhança com tensões mais baixas. Por outro lado, é de salientar que

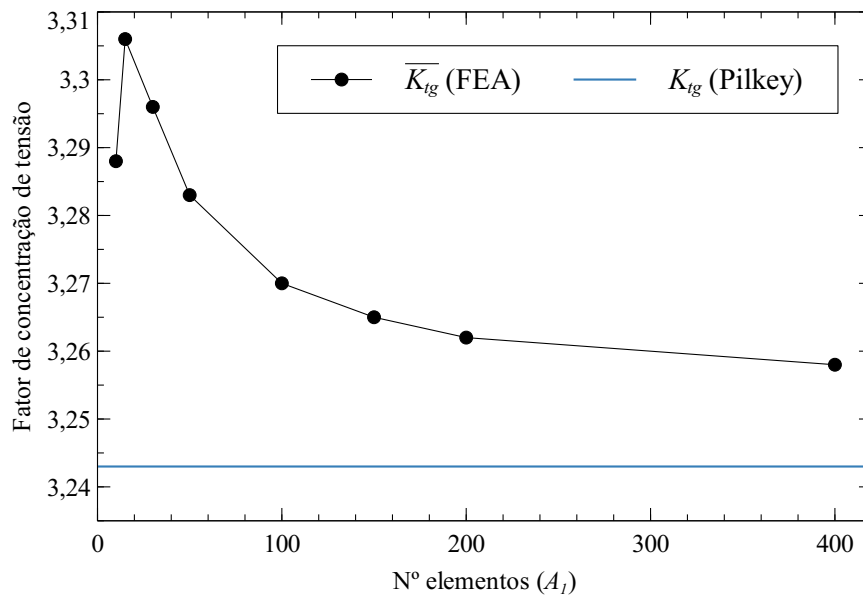


Figura 3.7: Fator de concentração de tensão em função do número de elementos.

uma densidade da malha mais elevada resulta num valor cada vez mais próximo do teórico. No entanto, a partir de certo número de elementos a diferença deixa de ser significativa como se observa a partir do gráfico 3.7.

Um fator de alguma importância na análise de malha de elementos é o tempo de computação de resultados que pode ser observado a partir da tabela 3.4.

Tabela 3.4: Tempo de computação dos teste para análise da malha

Teste	Nº elementos total	Nº elementos (A_1)	Tempo de computação [s]
1	500	10	0,75
2	1125	15	0,80
3	4500	30	1,44
4	12500	50	1,62
5	50000	100	4,47
6	112500	150	9,36
7	200000	200	17,98
8	800000	400	95,48

De acordo com a tabela, o tempo de computação é relativamente proporcional ao número de elementos que compõem a malha inteira. No entanto, como já foi referido, é somente do interesse desta análise observar a vizinhança do furo, mais precisamente no caso em estudo, a fronteira A_1 . Sendo assim, o tempo de análise,

relacionado com o número de elementos respetivo segundo A_1 pode ser visualizado graficamente através da figura 3.8.

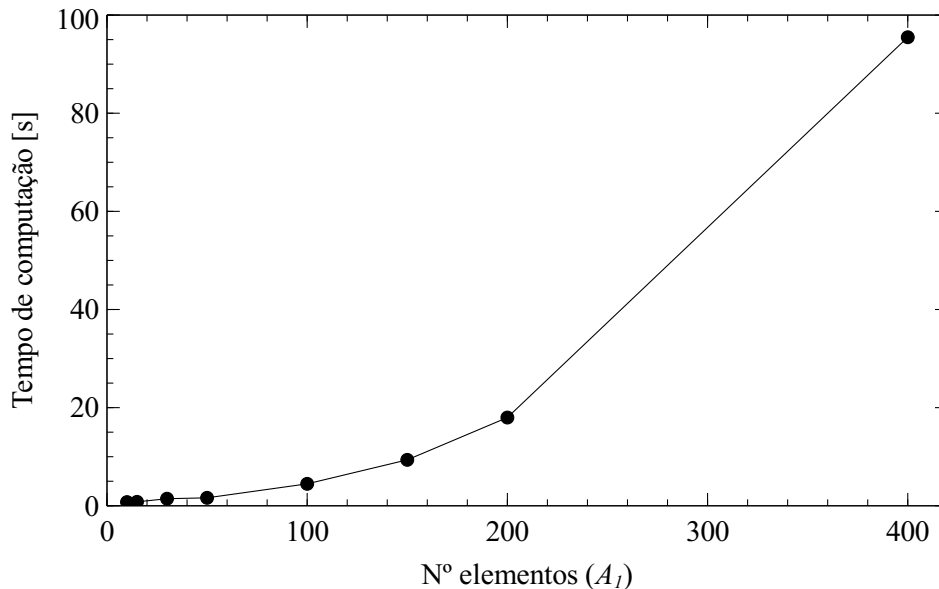


Figura 3.8: Tempo de computação em função do número de elementos.

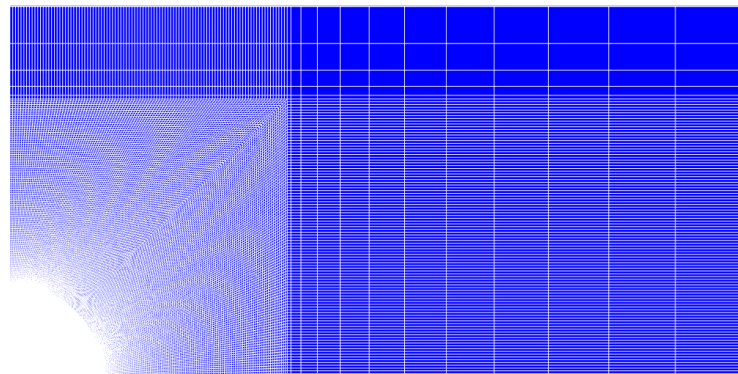
No caso da figura 3.8 já se observa um crescimento acentuado do tempo de computação dos resultados quando se aumenta o número de elementos. Este tempo pode ser um fator decisivo quando se pretende analisar geometrias mais complexas portanto é sempre do interesse do utilizador diminuir a necessidade de custos computacionais mais elevados. Importante realçar que o processador utilizado para este estudo é identificado pelas seguintes características:

- Intel Core i7-8550U @ 1.80GHz.

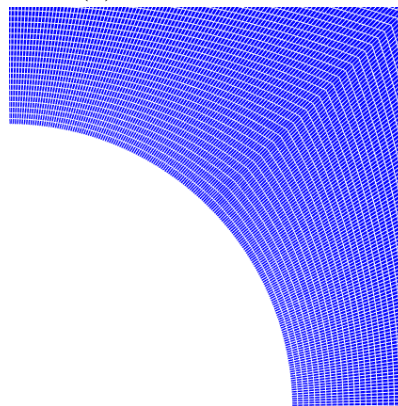
No âmbito desta secção observa-se então, a partir da análise da figura 3.7, que o valor de $\overline{K_{tg}}$ não se altera significativamente com um número superior a 100 elementos na zona de interesse A_1 , logo não há a necessidade da utilização de uma malha tão refinada. Esta malha mais densa tem igualmente como consequência um tempo de análise mais elevado, sendo que este cresce exponencialmente quando se pretende aumentar o número de elementos segundo A_1 . Conclui-se então que terá de existir um compromisso entre um custo computacional não comprometedor e resultados pouco influenciados pelo tamanho do elemento. Sendo que após este estudo, indica-se o “Teste 5” como o teste que já permite obter uma diferença relativa ao valor de referência bastante baixa, ou seja menor que 1%.

3.3.2 Refinamento gradual da malha

Como já foi indicado, um dos problemas de uma malha mais refinada é que tem como consequência um tempo de processamento dos resultados relativamente maior. Um dos objetivos do utilizador será a obtenção de resultados aceitáveis com o menor custo computacional possível. Logo, de modo a respeitar este critério, desenvolveu-se uma malha de elementos finitos para a aplicação em estudo que sofre um refinamento gradual, sendo mais densa na zona de interesse, ou seja, na vizinhança do furo. Numa primeira observação salienta-se que esta organização diminui o número de elementos que constituem a malha da geometria completa portanto, como se verificou pela tabela 3.4, o tempo de computação também diminuirá proporcionalmente. Apesar desta diminuição de elementos, esta malha de elementos finitos permite obter um número de divisões necessário somente na zona crítica de análise. Na figura 3.9 é possível verificar este refinamento gradual assim como a zona de interesse.



(a) Malha completa.



(b) Malha na zona de interesse.

Figura 3.9: Refinamento gradual da malha de elementos finitos.

De modo a poder comparar-se os efeitos deste tipo de discretização, conduziu-se um novo teste cujos resultados se encontram expressos na tabela 3.5 juntamente com os resultados do “Teste 5”, visto ter-se concluído na secção 3.3.1 que era o mais indicado. Neste caso ambos os testes têm o mesmo número de elementos na zona A_1 , ou seja 100 elementos, e cada elemento com 0,1 mm de lado. Lembra-se que devido ao refinamento gradual o “Teste 9” apresenta um número de elementos total menor.

Tabela 3.5: Resultados após o refinamento gradual da malha de elementos finitos

Teste	Nº elementos total	σ_{max} [MPa]	\overline{K}_{tg}	$diff$ (%)	Tempo de computação [s]
5	50000	163,491	3,270	0,833	4,47
9	21760	163,439	3,269	0,802	2,64

Como se pode observar, o “Teste 9” realizado após o refinamento gradual, resultou numa utilização de menos de metade do número de elementos total do “Teste 5” sendo que a diferença analítica entre os valores obtidos e o de referência mantém-se praticamente igual. Portanto, com a utilização deste método foi possível obter um tempo de computação menor com resultados tão ou mais precisos.

3.3.3 Elementos quadráticos e bi-quadráticos

Em todos os testes analisados do exemplo da placa com furo central foram utilizados elementos quadriláteros com 4 nós cada, referidos como QUAD4 na linguagem do *software*. Nesta secção será então abordado a implementação de elementos quadriláteros com 8 nós cada, QUAD8, e 9 nós cada, QUAD9.

Em relação ao *software*, a implementação destes tipos de elementos somente exige uma mudança no módulo Malha e ainda confirmar se o modelo implementado permite a utilização destes elementos, sendo que neste exemplo é usado o modelo C_PLAN que segundo a bibliografia disponível de Code-Aster é admitido, [21].

Os elementos quadriláteros de 8 nós, também nomeados de elementos *Serendipity* diferem dos elementos de 4 nós a partir da adição de um nó a meio de cada lado. Estes quatro nós a mais, têm como consequência a utilização de algumas funções de forma que serão quadráticas o que resulta numa melhor aproximação. Para além disso é aconselhado o uso de oito nós quando a geometria exige quadriláteros em vez de quadrados perfeitos como é o caso do exemplo em estudo, [5].

Em relação aos elementos quadriláteros de 9 nós, também nomeados de elementos *Lagrangianos* é acrescentado um nó interno no centro do elemento, sendo que

contém de igual forma aos elementos *Serendipity* 8 nós na fronteira que são responsáveis pelo contacto com o exterior. Este nó uniformiza todas as funções de forma como quadráticas, enquanto os *Serendipity* só conseguem obter todas as funções lineares possíveis e algumas quadráticas. Apesar desta melhoria de aproximação, o nó interior eleva o custo computacional sem muitas vezes representar resultados mais precisos, por isso mesmo os elementos quadriláteros de 8 nós são mais utilizados, [5].

De modo a verificar se existe alguma melhoria de resultados foram realizados dois testes, cada um com um tipo de elemento. Estes testes de seguida foram comparados com o teste já definido na secção 3.3.2 como o mais indicado, o “Teste 9”, sendo que para isso foi usado o mesmo número de elementos, portanto somente alterou-se o número de nós de cada elemento. Estes resultados encontram-se na tabela 3.6.

Tabela 3.6: Resultados após implementação de elementos QUAD8 e QUAD9

Teste	Tipo de elemento	Nº nós total	σ_{max} [MPa]	\overline{K}_{tg}	$diff$ (%)	Tempo de computação [s]
9	QUAD4	22078	163,439	3,269	0,802	2,64
10	QUAD8	65915	162,614	3,252	0,278	6,09
11	QUAD9	87675	162,614	3,252	0,278	8,47

Como se pode concluir, a utilização de elementos quadráticos gerou uma melhoria significativa de resultados já que a diferença relativa ao valor de referência diminuiu para menos de metade. No entanto, entre o “Teste 10” e o “Teste 11” os resultados são iguais já que o σ_{max} ocorre numa fronteira de um elemento, logo o nó interno não altera este valor. De qualquer forma, é aconselhada a implementação de elementos quadráticos, mais precisamente elementos QUAD8.

3.3.4 Análise da distribuição da tensão

Nas secções anteriores concluiu-se que o “Teste 10” foi o modelo que mais se aproximou ao valor teórico apresentado na secção 3.2, no entanto este valor de comparação é em relação a K_{tg} , ou seja, relativo à tensão máxima. Será então necessário estudar a distribuição da tensão ao longo de A_1 , de modo a verificar se o modelo é válido para avaliar este comportamento.

A distribuição de σ_x ao longo da placa para o “Teste 10” pode ser visualizada na figura 3.10. De salientar que σ_x corresponde a σ_θ , quando $\theta = \pi$, que é a zona em estudo.



Figura 3.10: Representação gráfica da distribuição da tensão circunferencial no “Teste 10”.

A fronteira A_1 corresponde, em coordenadas polares adimensionais em relação à largura da placa, a uma fronteira compreendida entre $a/w = 0,25$ e $a/w = 0,75$, sendo que a direção será $\theta = \pi$. Para estas coordenadas já foi apresentada uma equação que relaciona a tensão circunferencial com a/w na equação 3.10 para o caso em estudo, ou seja, $\gamma_n = 0,25$ e $\sigma_\infty = 50$ MPa. A partir do *software* utilizado é possível determinar σ_θ para todos os nós existentes na zona A_1 do “Teste 10”. A comparação entre estas duas distribuições pode ser observada na figura 3.11.

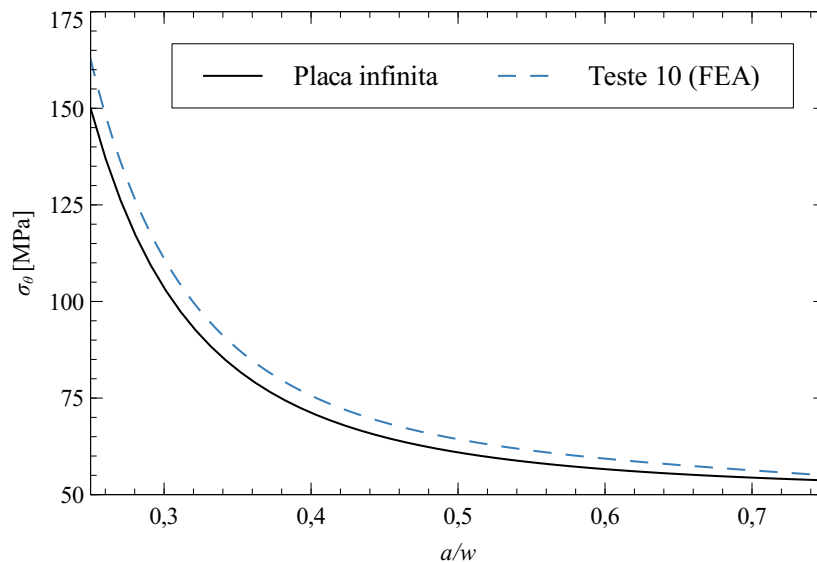


Figura 3.11: Comparação entre distribuição teórica e distribuição obtida por elementos finitos.

Apesar de a equação 3.10 ser formulada para o caso de uma placa infinita em ambas as direções com um furo central, e a geometria do “Teste 10” ser uma placa semi-infinita com furo central, a distribuição de σ_θ em ambos os casos segue o mesmo padrão ao longo de A_1 . Existe um erro inerente ao facto da diferença de geometrias mas é bastante reduzido, sendo que a maior diferença relativa entre as duas distribuições é de 8,093% e localiza-se onde ocorre σ_{max} , ou seja, $a/w = 0,25$ ou $a/w = \gamma_n$. De salientar que à medida que se considera nó mais distantes do furo, as duas distribuições se aproximam, sendo que no caso da distribuição da placa infinita o valor de σ_θ irá convergir para um igual a σ_∞ .

3.3.5 Análise de uma placa a três dimensões

Em todos os estudos realizados até agora foi considerado que a espessura não teria uma influência predominante no estado da tensão, e portanto formulou-se somente um plano a duas dimensões. No entanto, será interessante comparar os dois modelos, um a três dimensões e o do “Teste 10”, cujos os resultados já foram validados.

Os elementos que foram utilizados são do tipo hexaédrico com oito nós, identificados no *software* como HEXA8, sendo que à semelhança dos QUAD4, as funções de forma constituem funções lineares. A discretização nas bases é definida pelo mesmo número de divisões que no “Teste 10”. No entanto, por ser um modelo a três dimensões exige uma discretização ao longo da espessura que definiu-se como quatro elementos, portanto, 1 mm de lado segundo z . Na figura 3.12 é possível observar a malha de elementos finitos utilizada.

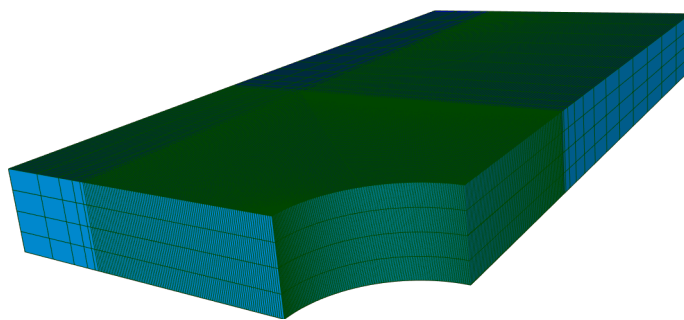


Figura 3.12: Malha de elementos finitos do modelo a três dimensões.

No desenvolvimento do algoritmo em linguagem Code_Aster foi necessário atribuir o modelo 3D e definir as condições de fronteira, que à semelhança do modelo a duas dimensões, utilizou-se as condições resultantes da simetria do problema. Portanto os deslocamentos na direção x (dx) do plano correspondente a $x = 0$ são

nulos, e de igual forma, são nulos os deslocamentos na direção y no plano $y = 0$. De seguida, foi imposta a solitação na face coincidente com o plano $x = w$ como uma força de pressão igual ao valor σ_∞ , [25].

Na análise realizada procedeu-se à comparação entre os resultados obtidos no meio da placa com os obtidos na base sendo que na tabela 3.7 é possível verificar algumas características deste modelo assim como os resultados obtidos. Por motivos de comparação incluiu-se os parâmetros já apresentados do “Teste 10”.

Tabela 3.7: Resultados após implementação do modelo 3D

Teste	Nº nós total	σ_{max} [MPa]	K_{tg}	$diff(\%)$	Tempo de computação [s]
10	65915	162,614	3,252	0,278	6,09
12 meio	110390	157,490	3,150	2,879	46,22
12 base	-	192,028	3,841	18,420	-

Os resultados no meio da placa encontram-se relativamente próximos dos de referência, no entanto com uma diferença maior que no modelo a duas dimensões. A justificação para esta ocorrência encontra-se no facto do número de elementos ao longo da espessura ainda ser muito reduzido o que resulta num resultado pouco preciso. Contudo, um elevado número de elementos ao longo da espessura incrementa exponencialmente o custo computacional portanto o modelo a três dimensões não será o mais indicado para modelar o caso da placa fina.

Por fim, apresenta-se a partir do módulo ParaView a representação gráfica de σ_x na figura 3.13.

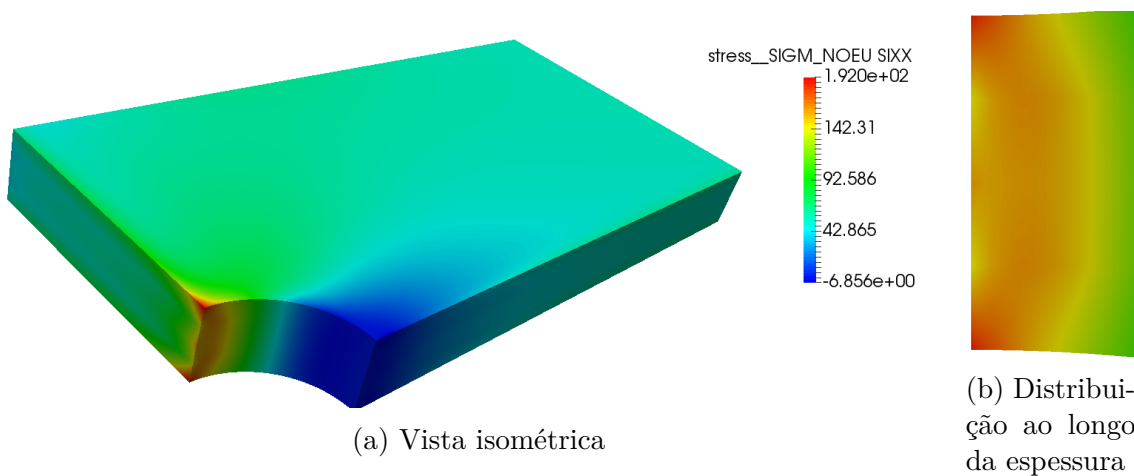


Figura 3.13: Distribuição da tensão circunferencial ao longo da placa.

3.3.6 Variação do diâmetro do furo

Como foi referido na secção 3.1, o parâmetro γ_n representa a razão entre o diâmetro do furo (d) e comprimento normal à solicitação uni-axial, no caso em estudo a largura (W). Do ponto de vista construtivo, este parâmetro pode variar bastante sendo que quanto maior o diâmetro do furo menor a capacidade de prever o comportamento da tensão, no entanto, existe resultados precisos para diferentes γ_n como foi apresentado no gráfico 3.2, [11, 14].

Com o interesse de estudar a capacidade do modelo criado analisar diferentes tipos de construção foram realizados vários testes com diferentes γ_n . A mudança deste parâmetro exige uma mudança de geometria, no entanto, o *software* é capaz de criar esta geometria a partir de variáveis que o utilizador pode alterar, dando *input* ao valor desejado. De seguida, foi necessário discretizar numa malha de elementos finitos a geometria criada. Neste âmbito, decidiu-se para efeitos de referência entre os vários testes manter o tamanho de elemento em 0,1 mm segundo a direção da zona de interesse A_1 . De salientar que a zona A_1 irá variar entre testes já que foi definido que seria entre $a = r$ e $a = 3r$, com $\theta = \pi$. No entanto, sempre que $3r > w$, a zona A_1 será limitada pela própria largura da placa. Ainda em relação à malha utilizada será importante referir que como se observou em secções anteriores o elemento QUAD8 será o mais adequado para obter melhores resultados, logo foi o utilizado. Na tabela 3.8 encontram-se os resultados obtidos para diferentes testes com diferentes γ_n .

Tabela 3.8: Resultados obtidos após variar o diâmetro do furo

Teste	γ_n	σ_{max} [MPa]	$\overline{K_{tg}}$	K_{tg}	$diff(\%)$
13	0,1	151,602	3,032	3,035	0,098
14	0,2	157,725	3,155	3,149	0,191
15	0,3	169,064	3,381	3,368	0,386
16	0,4	188,050	3,761	3,733	0,750
17	0,5	218,625	4,373	4,314	1,368

Após análise da tabela 3.8 confirma-se que à medida que o diâmetro do furo aumenta em relação ao mesmo valor da largura, o fator de concentração de tensão irá aumentar. E ainda que, a diferença relativa ao valor de referência K_{tg} varia com γ_n , sendo que para valores maiores esta diferença será igualmente maior, logo pode-se concluir que os resultados serão menos precisos em soluções construtivas com o diâmetros maiores em relação à largura. No entanto, no caso de $\gamma_n = 0,5$ a diferença é de 1,368% que se considera relativamente próximo ao valor de referência,

portanto considera-se que este modelo consegue calcular o comportamento neste tipo de exemplos.

A partir do gráfico 3.14 é possível visualizar a comparação entre os valores obtidos por Howland, a aproximação apresentada por Pilkey e os resultados obtidos por elementos finitos.

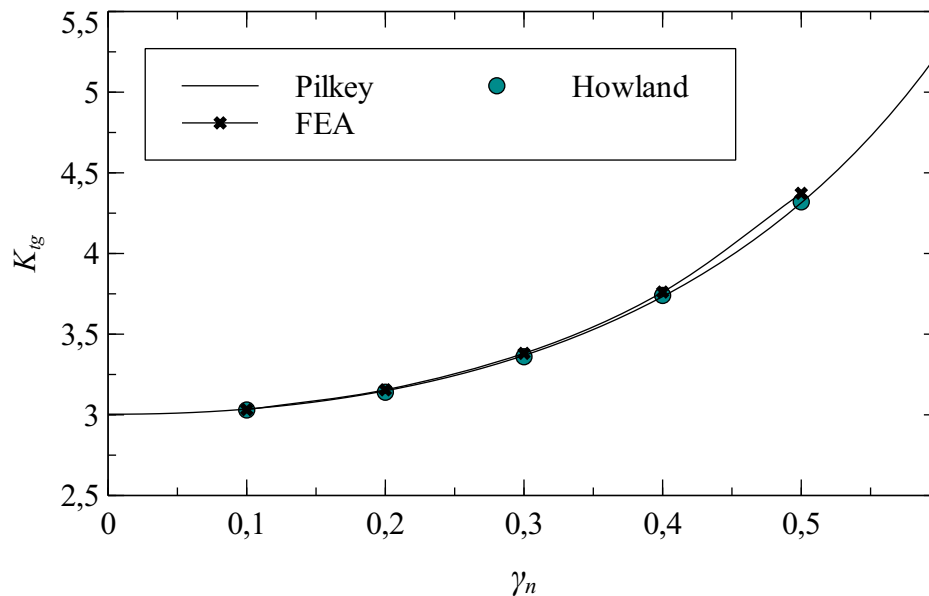


Figura 3.14: Comparação entres os valores obtidos por elementos finitos e valores de referência, [14, 11].

3.4 Discussão de resultados

No âmbito deste capítulo foi possível explorar o *software* em código aberto Salome-Meca e os seus módulos com vista a analisar o caso de estudo de uma placa semi-infinita com um furo central. O *software* foi capaz de criar vários tipos de geometria e automaticamente defini-la a partir algumas variáveis de entrada. De seguida utilizou-se o módulo de criação de uma malha de elementos finitos, onde foi possível definir vários parâmetros para diferentes tipos de malha. Após este pré-processamento, formulou-se um algoritmo na linguagem Code_Aster que simulou o modelo atribuído e procedeu uma análise estática a partir dos dados impostos, como as solicitações, material e condições de fronteira. Por fim, realizou-se um pós-processamento dos resultados para obter o estado de tensão da placa assim como uma representação gráfica da distribuição destes resultados.

Após a apresentação de alguns conceitos teóricos para definir valores de referência para a análise da placa, comparou-se estes com resultados obtidos na análise por elementos finitos de forma a averiguar vários tipos de parâmetros importantes para uma validação do modelo imposto.

No caso da análise do tamanho do elemento necessário para obter a malha com melhor performance sem comprometer o custo computacional exigido concluiu-se que seriam elementos de 0,1 mm segundo a direção y na zona referida previamente como A_1 . Após esta análise explorou-se uma nova organização da malha de forma a obter resultados tão precisos com menor número de elementos no total que revelou-se bastante útil já que demonstrou melhorias significativas. De seguida, procedeu-se a uma comparação entre vários tipos de elementos, sendo que neste caso, atribuiu-se aos elementos quadráticos referidos como QUAD8 a melhor aproximação aos resultados, reduzindo para metade a diferença relativa ao valor de referência. A partir do teste que demonstrou uma aproximação da tensão máxima mais correta, comparou-se a distribuição da tensão ao longo de uma fronteira com valores teóricos. Nesta secção observou-se que as duas distribuições são bastante semelhantes o que permitiu concluir que o teste indicado encontra-se válido para analisar a distribuição da tensão.

Como em análises anteriores não se tinha considerado a espessura da placa decidiu-se realizar um modelo a três dimensões que neste caso não demonstrou melhoria de resultados. Portanto a aproximação anteriormente considerada permite mais liberdade ao utilizador de realizar um modelo de elementos finitos mais refinado e com resultados mais precisos.

Por fim, foram estudadas outras geometrias de placas com furos de diâmetros diferentes em que se implementaram parâmetros que já tinham sido validados em testes anteriores. Após comparação com valores teóricos concluiu-se que o modelo criado aproximou-se bastante aos resultados pretendidos mesmo quando se aplicou uma geometria que não a inicial.

Esta página foi intencionalmente deixada em branco.

Capítulo 4

Arquitetura de uma Rotina de Análise Estrutural Otimizada

A utilização de um *software* de elementos finitos requer algum tempo e conhecimento para abordar todas as suas especificações. No entanto, o caso de análises repetitivas, em que se variam somente alguns parâmetros, como no caso do capítulo 3, abre a possibilidade de automatizar este processo o que resulta numa maior eficiência. Portanto, neste capítulo irá ser apresentada toda a arquitetura para implementar uma rotina que permite realizar análises estruturais automáticas e com a mínima interação com o utilizador. Será apresentado o processo para obter resultados em que se utiliza o *software* Salome-Meca para realizar a modelação e análise e ainda a linguagem de programação Python para implementar a rotina. Neste capítulo somente será abordada a arquitetura geral para construir este tipo de algoritmos, sendo que exemplos de implementação assim como casos de validação serão expostos em capítulos posteriores.

À semelhança de uma análise por elementos finitos típica, será necessário que esta rotina realize um pré-processamento do modelo, que inclui tratamento da geometria e criação da malha. De seguida, terá de ter capacidade para realizar uma análise estrutural, através de vários parâmetros como condições de fronteira, tipo de modelação ou solicitações. E por fim, se assim for pretendido, realizar um pós-processamento que poderá ser um tratamento dos resultados incluindo uma visualização gráfica dos campos obtidos. Para além destes passos associados à análise de elementos finitos propriamente dita, ainda existe a necessidade de ler os dados de entrada, assim como alguns processos auxiliares como a preparação dos ficheiros próprios para a análise.

Sendo assim, como primeira abordagem à rotina a criar, esta integra os seguintes módulos:

- Leitura de dados de entrada e criação de ficheiros para a análise;
- Preparação de ficheiros;
- Pré-processamento do modelo;
- Cálculo por elementos finitos (*solver*);
- Pós-processamento dos resultados.

Com exceção da inserção dos dados de entrada caso seja necessário, o objetivo da rotina criada será de realizar todos estes processos de forma autónoma e com a mínima interação com o utilizador. Este conceito assemelha-se a sistemas do tipo *black box* em que o utilizador somente terá de fornecer o *input* e após executar o programa receber o *output* sem ter necessidade de conhecer intrinsecamente as rotinas internas, [9].

Os módulos referidos serão chamados a partir de um ficheiro principal (*main*) sendo que os algoritmos para a sua aplicação estarão presentes em ficheiros independentes. Esta execução divide-se ainda em algoritmos que foram criados com a finalidade desta otimização ou a execução do próprio Salome-Meca a par de ficheiros exigidos por este *software* para realizar a análise.

Ao longo deste capítulo irá ser apresentado todo o fluxo de uma rotina deste tipo, assim como todos os algoritmos auxiliares para um bom desempenho. Na figura 4.1 é possível visualizar o fluxograma da rotina principal.

4.1 Ler dados de entrada e criar ficheiros

A leitura dos dados de entrada consiste em definir todas as variáveis de entrada necessárias ao processo e é o primeiro passo desta rotina. Também será o único passo que o utilizador terá contacto com o programa sendo que os subsequentes serão automáticos.

De forma a implementar uma programação orientada a objetos, utilizou-se o conceito de classes para definir estes dados. Uma classe é representada pelos seus atributos e pelos seus métodos. Ao iniciar um objeto através de uma classe, esse objeto recebe todos os atributos da classe e pode executar funções que são os respetivos métodos da mesma classe. Assim estes objetos podem ser usados ao longo

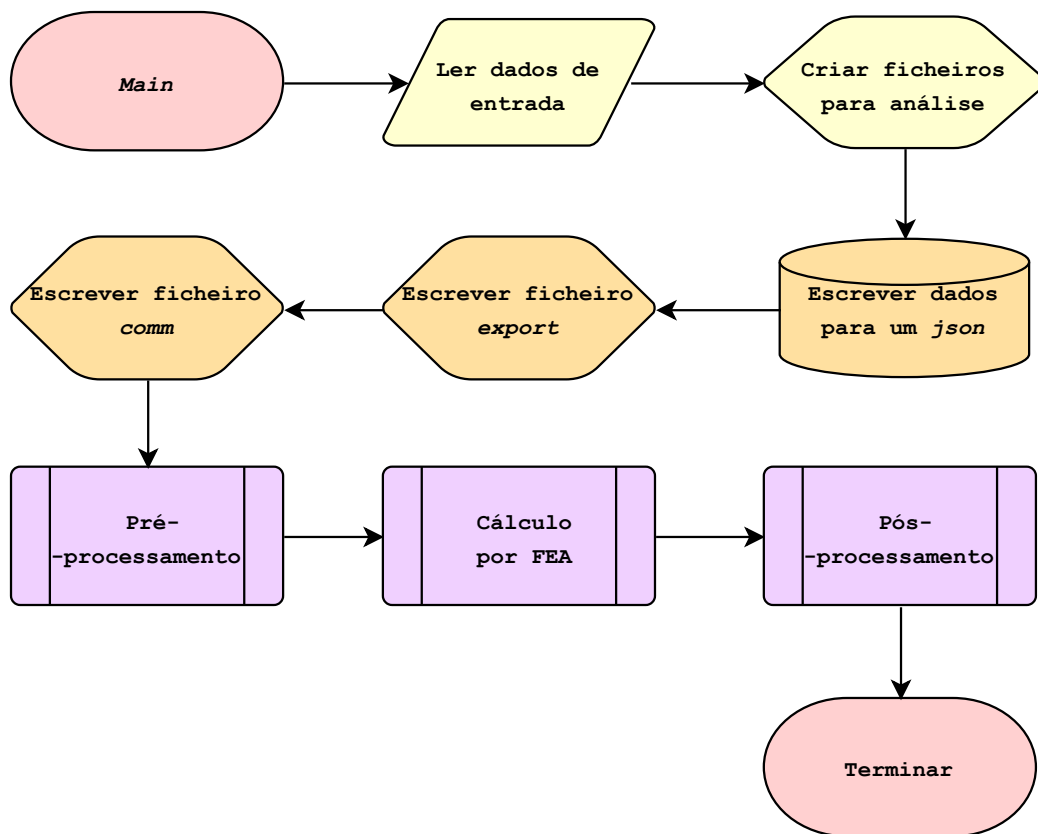


Figura 4.1: Fluxograma da rotina principal.

da rotina para as várias instâncias em que serão necessários. O exemplo do código para definir uma classe segundo a linguagem Python, pode ser observado no código 4.1, [19].

Código 4.1: Classe criada segundo Python para a geração de uma face.

```

class Face:

    def __init__(self, side_horizontal, side_vertical, thickness):
        self.side_horizontal = float(side_horizontal)
        self.side_vertical = float(side_vertical)
        self.thickness = list(thickness)
        self.construction = 'Plane'
        self.bend_length = 0

    def is_curve(self, bend_length):
        self.construction = 'Curve'
        self.bend_length = float(bend_length)

    def thickness_iterable(self, val_min, val_max, interval):
        self.thickness = list(range(val_min, val_max + 1, interval))
  
```

As classes necessárias dependem da finalidade da rotina implementada, mas essencialmente, para uma análise estrutural típica, é importante definir os seguintes *inputs*:

- Dimensões da geometria em estudo;
- Parâmetros para criar a malha de elementos finitos;
- Propriedades do material ou materiais que serão utilizados;
- Condições da análise, como condições de fronteira e solicitações;
- Critérios para definir quais os resultados de interesse, como critérios de tensões/deformações equivalentes.

A leitura de dados pode ser realizada partir de uma interface gráfica, na própria rotina, ou ainda a partir de ficheiros de transferência de dados, como os ficheiros *json*.

O segundo passo para iniciar este programa e algoritmos auxiliares será relativo aos ficheiros ou diretórios necessários ao seu funcionamento. Este ficheiros podem ser criados ao longo da execução ou já existirem previamente e, neste caso, é definido o diretório onde estão localizados. Ainda existe a criação de pastas onde são guardados este ficheiros em que a pasta-mãe de cada estudo criado tem uma identificação própria. Por fim, é necessário igualmente definir os diretórios onde estão presentes os ficheiros executáveis que permitem à rotina lançar os *softwares*, como o Salome-Meca para o pré e pós-processamento e o Code_Aster para o cálculo por elementos finitos. Estes ficheiros ainda podem ser divididos em rotinas que serão executadas ou ficheiros com informação que serão lidos pelos programas.

Assim, os ficheiros que contêm rotinas são:

- Ficheiro com formato *py* com rotinas de pré-processamento;
- Ficheiro com formato *comm* com rotinas de cálculo por elementos finitos;
- Ficheiro com formato *py* com rotinas de pós-processamento;
- Ficheiro com formato *batch* para executar o Salome-Meca;
- Ficheiro com formato *batch* para executar o Code_Aster.

Em relação aos ficheiro que transferem informação, identificam-se os seguintes:

- Ficheiro com formato *json* para transferência de dados;

- Ficheiro com formato *med* com informação da malha para *input*;
- Ficheiro com formato *hdf* que permite usar o ambiente gráfico do Salome-Meca para visualização do modelo;
- Ficheiro com formato *export* com informações para iniciar o estudo;
- Ficheiro com formato *mess* com relatório do estudo realizado;
- Ficheiro com formato *med* com *output* do estudo realizado.

Ainda é necessária a criação de pastas para armazenamento que permitem guardar todos estes ficheiros para facilitar o acesso e ainda rever todo o processo para despistar algum tipo de erro.

Esta dinâmica de criação destes ficheiros é responsável pelo carácter autónomo da rotina implementada, visto que sem eles, o utilizador teria de criar manualmente cada componente do estudo. Assim é possível obter uma eficiência do tempo utilizado para a realização de uma análise por elementos finitos bastante mais elevada. De seguida encontra-se uma explicação da função de cada ficheiro.

4.1.1 Ficheiro com formato *json*

Os dados de entrada relativos a cada estudo são definidos no ficheiro principal (*main*), no entanto, estes dados são necessários para a realização de outras rotinas essenciais, como os dados geométricos para os algoritmos de pré-processamento e ainda, os dados relativos à análise para a realização do cálculo. Como estes algoritmos são ficheiros independentes da rotina principal existe a necessidade de transferir os dados de entrada de forma a não ter que os inicializar mais que uma vez. Assim, existe a criação de um ficheiro *json*, que armazena todas as informações de entrada e é chamado sempre que seja necessário utilizar estes dados.

Um ficheiro do tipo *JavaScript Object Notation* ou *json* assemelha-se na linguagem Python a um dicionário em que existem pares do nome de uma variável e respetivo valor que podem ser respetivos a um objeto. O ficheiro *json* é então escrito na origem, para de seguida ser acedido por outros ficheiros de acordo com a finalidade da sua utilização. Desta forma é possível transferir todos os dados de entrada para os ficheiros em que sejam necessários ao longo da execução, [12].

4.1.2 Ficheiro com formato *py* para pré-processamento

Este ficheiro do tipo Python contém a rotina necessária para realizar o pré-processamento do modelo. Este pré-processamento consiste em gerar uma geometria e uma malha de elementos finitos através de código. Outra componente importante é definir grupos de elementos que permitem na análise subsequente identificá-los para atribuir certas propriedades.

As variáveis iniciais deste algoritmo são as variáveis definidas a partir dos dados de entrada, e os valores são transferidos para este ficheiro através de um *json*. Assim, este algoritmo é estabelecido para um modelo geral paramétrico e quando se executa o programa o modelo é criado com base nos dados de entrada. Portanto, o ficheiro *main* tem de simplesmente receber a localização deste ficheiro no sistema para quando for necessário realizar o pré-processamento executar.

Este ficheiro após executado exporta todas as informações da malha de elementos finitos gerada para um ficheiro do tipo *med* e ainda guarda o modelo gerado para visualização num ficheiro do tipo *hdf*.

4.1.3 Ficheiro com formato *med* para *input*

O formato *med* ou *Modélisation et Échanges de Données* funciona como um ficheiro de transferência de dados desenvolvido pela EDF que é característico da plataforma do Salome-Meca e permite guardar toda a informação de uma malha de elementos finitos. Este ficheiro pode assim ser utilizado subsequentemente no cálculo para ler o modelo em estudo e ainda todas as propriedades da malha, como o tipo de elemento ou ainda os grupos de elementos criados, [23].

Este ficheiro é escrito quando é executada a rotina de pré-processamento, no entanto, terá de ser criado previamente e transferida a localização deste ficheiro para a rotina que gera a malha através do ficheiro *json*.

4.1.4 Ficheiro com formato *hdf* para visualização do modelo

O formato *hdf* ou *Hierarchical Data Format* é um ficheiro para armazenamento e organização lógica de grandes quantidades de informação que pode ser lido por várias aplicações e é o tipo de formato utilizado pelo Salome-Meca. No caso desta rotina, é guardado o modelo gerado neste formato para haver a possibilidade de o utilizador visualizá-lo no ambiente gráfico do *software* assim como toda a informação criada, [10].

É criado um ficheiro vazio no início da rotina e é transferida a localização deste ficheiro para a rotina de pré-processamento que guarda o modelo nesse diretório.

4.1.5 Ficheiro com formato *comm*

O ficheiro *command* identificado pelo seu formato *comm* descreve toda a simulação realizada através do *software* Code_Aster. Este ficheiro é composto pelas funções que dão as instruções necessárias para realizar a análise completa, como ler a malha, atribuir condições de fronteira, atribuir solicitações, entre outros passos, [22].

O *input* da análise serão os ficheiros *med* com as informações das malhas de elementos finitos e após a análise estrutural é possível imprimir os resultados para outros ficheiros *med* para *output*. Este ficheiro é criado e escrito pela rotina implementada em Python, de acordo com os objetivos da simulação e é executado no ambiente do Code_Aster. Como a linguagem utilizada pelo *software* é compatível com a linguagem Python, é possível implementar rotinas próprias em conjunto com as funções destinadas à análise.

Os ficheiros de *input* e *output* são identificados por um *Logical Unit Number* que funciona como um identificador para o ficheiro *comm* utilizar esses ficheiros. A informação com os *Logical Unit Number* de cada ficheiro assim como características do estudo encontram-se presentes no ficheiro *export*.

4.1.6 Ficheiro com formato *export*

O ficheiro *export* contém todos os parâmetros necessários à execução do estudo, como limites de tempo e de memória, e ainda os diretórios dos ficheiros necessários para análise, como ficheiros *med* para *input* e *output*. O ficheiro é escrito com todas as informações pela própria rotina, que transfere as localizações dos ficheiros previamente criados para este ficheiro. Os ficheiros são identificados por um *Logical Unit Number* que mais tarde é lido durante a execução da análise, [33].

Para além dos diretórios já referidos ainda é necessário ao estudo um ficheiro do tipo *mess* e uma pasta para guardar os ficheiros temporários criados durante a análise que é apagada após execução.

4.1.7 Ficheiro com formato *mess*

O ficheiro *mess* ou *message* é um ficheiro criado previamente e escrito durante a execução do estudo que transcreve um relatório da análise realizada como o algoritmo

executado, propriedades do estudo e ainda eventuais erros que podem ter ocorrido. Este ficheiro é importante para apurar se a análise foi bem realizada ou não, [22].

4.1.8 Ficheiro com formato *med* para *output*

O ficheiro *med* para *output* é escrito pela própria análise e tem como objetivo transferir todas as informações dos resultados calculados, como deslocamentos e tensões. Este ficheiro pode ser essencialmente de dois tipos, ou de leitura de resultados ou de visualização dos mesmos a partir de outras ferramentas, como o ParaView. Esta distinção é indicada na própria análise, [23].

Estes ficheiros, são criados pela rotina principal quando se inicia os dados, e a sua localização é fornecida ao ficheiro *export*.

4.1.9 Ficheiro com formato *py* para pós-processamento

Em relação aos ficheiros necessários criar para iniciar a rotina, o ficheiro do tipo Python para realizar o pós-processamento será o último. Este ficheiro só será necessário caso se pretenda visualizar os resultados na ferramenta de pós-processamento do Salome-Meca, o ParaView. Este ficheiro recebe a partir do *json* a localização do ficheiro *med* com os resultados e quando executado no ambiente de visualização permite obter o mapa de cores ou dos deslocamentos ou das tensões.

Ainda existe a possibilidade de criar outras rotinas de pós-processamento no âmbito da leitura e tratamento dos dados para melhor compreensão do utilizador, como, por exemplo, imprimir num ficheiro de texto os resultados máximos ou até avaliar estes resultados em relação a critérios mecânicos definidos.

4.1.10 Pastas para armazenar ficheiros

Todos os ficheiros criados terão de ter uma localização no sistema conhecida pela rotina, por isso mesmo, são criadas pastas pelo próprio algoritmo onde guarda os ficheiros já referidos. Existe uma pasta-mãe respetiva a cada estudo, onde é possível encontrar todos os componentes do processo já referidos.

4.1.11 Ficheiro com formato *batch* para executar o Salome-Meca e Code_Aster

Um ficheiro do tipo *batch* permite executar linhas de código de um programa na sua plenitude. Como este algoritmo tem a necessidade de executar o programa

Salome-Meca para pré e pós-processamento e ainda o Code_Aster para o cálculo por elementos finitos, terá de se localizar o diretório dos ficheiros que executam estes *softwares* para durante a própria rotina utilizá-los. Estes ficheiros são encontrados na pasta onde o programa foi instalado.

4.2 Preparação de ficheiros

Após iniciar os dados de entrada e estabelecida a localização de todos os ficheiros necessários ao estudo, segue-se a preparação de alguns ficheiros que serão utilizados. Estes ficheiros podem já existir previamente ou serem criados durante a execução com o objetivo de modificar o seu conteúdo. Segue-se uma lista dos ficheiros utilizados e respetiva finalidade:

- O ficheiro *json* com todos os dados de entrada;
- O ficheiro *py* para pré-processamento com a rotina que gera a geometria e a malha;
- O ficheiro *comm* com a rotina de análise;
- O ficheiro *export* com as informações do estudo e dos ficheiros que irão ser utilizados;
- O ficheiro *py* para pós-processamento para visualização ou tratamento de dados.

Relativamente aos ficheiros *json*, *comm* e *export*, estes são escritos dentro da própria rotina através de funções criadas para essa finalidade. No caso do ficheiro *comm*, este possibilita alterar e manipular o conteúdo da rotina de análise para implementação de diferentes tipos de análise. Em relação ao ficheiro *export*, ao contrário do ficheiro *comm*, não é uma rotina mas sim um ficheiro leitura pelo próprio *software*, portanto é necessário escrever este ficheiro de acordo com as informações que se pretende fornecer ao Code_Aster.

Os ficheiros Python para pré e pós-processamento, são ficheiros escritos previamente pelo programador quando desenvolve uma rotina deste tipo. Depois são executados dentro do ambiente do Salome-Meca, que conforme os dados de entrada, gera o modelo no caso do ficheiro de pré-processamento, ou mostra os resultados no caso do ficheiro de pós-processamento. De seguida, encontra-se uma explicação da arquitetura de cada ficheiro.

4.2.1 Arquitetura do ficheiro *json*

O ficheiro *json* é escrito através do algoritmo Python, em que dentro de cada classe existem pares variável/valor que podem ser acedidos posteriormente. O exemplo da estrutura de dados de um ficheiro *json* pode ser visualizado no código 4.2, [12].

Código 4.2: Estrutura de um ficheiro *json*

```
{
  "Geometria": [
    {
      "lado_quadrado": 10.0,
      "espessura": 1.0
    }
  ],
  "Malha": [
    {
      "tamanho_elemento": 2.0
    }
  ],
  "Material": [
    {
      "young" : 21000.0,
      "poisson": 0.3
    }
  ],
  "Condicoes": [
    {
      "pressao": -0.1
    }
  ]
}
```

4.2.2 Arquitetura do ficheiro *py* para pré-processamento

A geometria e malha são criadas pelo programador através de parâmetros sendo que, após o *input* das dimensões e parâmetros de malhagem, o modelo é gerado a partir do código. Nesta rotina também podem ser previstas várias situações, como inclusão de acessórios à geometria principal, ou ainda diferentes tipos de construções.

Assim, evita-se a utilização do ambiente gráfico do *software* para realizar o pré-processamento.

Esta rotina é criada segundo a linguagem Python, no entanto as funções utilizadas são de bibliotecas associadas ao Salome-Meca que contêm funções de edição da geometria e de edição da malha. Por esta razão, é necessário que este *script* seja executado no ambiente do próprio Salome-Meca, e para isso somente terá de se abrir o programa na rotina principal a partir do seu ficheiro *batch*.

No fim da execução é necessário exportar toda a informação da malha de elementos finitos, como o próprio modelo, tipo de elemento ou grupos de elementos, para um ficheiro com formato *med* para posterior utilização. Ainda é possível guardar o modelo criado num ficheiro com formato *hdf*.

O exemplo de uma rotina para gerar um quadrado e a respetiva malha de elementos finitos pode ser visualizado no código 4.3. Salientar que também se demonstra neste exemplo a leitura de dados de entrada através do ficheiro *json*.

Código 4.3: Rotina para gerar a malha de elementos finitos de um quadrado

```
import salome
from salome.geom import geomBuilder
from salome.smesh import smeshBuilder
import os
import json
import SMESH

salome.salome_init()
estudo = salome.myStudy
geompy = geomBuilder.New(estudo)
smesh = smeshBuilder.New(estudo)

with open(os.path.join(os.getcwd(), 'Dados.json')) as f:
    dados = json.load(f)
    f.close()
for read in dados['Geometria']:
    lado_quadrado = float(read['lado_quadrado'])
for read in dados['Malha']:
    tamanho_elemento = float(read['tamanho_elemento'])

diretorio_med = os.path.join(os.getcwd(), 'Malha.med')
diretorio_hdf = os.path.join(os.getcwd(), 'Modelo.hdf')

vertice_1 = geompy.MakeVertex(lado_quadrado, lado_quadrado, 0)
vertice_2 = geompy.MakeVertex(-lado_quadrado, lado_quadrado, 0)
vertice_3 = geompy.MakeVertex(-lado_quadrado, -lado_quadrado, 0)
vertice_4 = geompy.MakeVertex(lado_quadrado, -lado_quadrado, 0)
linha_1 = geompy.MakeLineTwoPnt(vertice_2, vertice_1)
linha_2 = geompy.MakeLineTwoPnt(vertice_1, vertice_4)
linha_3 = geompy.MakeLineTwoPnt(vertice_4, vertice_3)
linha_4 = geompy.MakeLineTwoPnt(vertice_3, vertice_2)
lista_de_linhas = [linha_1, linha_2, linha_3, linha_4]
```

```

face = geompy.MakeFaceWires(lista_de_linhas, 1)
for i in range(len(lista_de_linhas)):
    geompy.addToStudy(lista_de_linhas[i], 'linha_{}'.format(i + 1))
geompy.addToStudy(face, 'face')

malha = smesh.Mesh(face)
elemento_1d = malha.Segment()
tamanho_elemento_1d = elemento_1d.LocalLength(tamanho_elemento, None, 1e-007)
elemento_2d = malha.Quadrangle(algo=smeshBuilder.QUADRANGLE)
malha.Compute()
for i in range(len(lista_de_linhas)):
    criterio = [smesh.GetCriterion(SMESH.EDGE, SMESH.FT_BelongToGeom,
                                  SMESH.FT_Undefined, lista_de_linhas[i],
                                  SMESH.FT_Undefined, SMESH.FT_Undefined, 1e-007)]
    filtro = smesh.GetFilterFromCriteria(criterio)
    filtro.SetMesh(malha.GetMesh())
    grupo_linha = malha.GroupOnFilter(SMESH.EDGE, 'grupo_linha', filtro)
    smesh.SetName(grupo_linha, 'grupo_linha_{}'.format(i + 1))
grupo_face = malha.GroupOnGeom(face, 'grupo_face', SMESH.FACE)

smesh.SetName(elemento_1d.GetAlgorithm(), 'elemento_1d')
smesh.SetName(elemento_2d.GetAlgorithm(), 'elemento_2d')
smesh.SetName(tamanho_elemento_1d, 'tamanho_elemento_1d')
smesh.SetName(grupo_face, 'grupo_face')
smesh.SetName(malha.GetMesh(), 'malha')

malha.ExportMED(diretorio_med, 0, SMESH.MED_V2_2, 1, None, 1)
salome.myStudyManager.SaveAs(diretorio_hdf, estudo, False)

```

4.2.3 Arquitetura do ficheiro *comm*

O ficheiro *comm* contém todas as funções necessárias ao cálculo segundo a linguagem utilizada no Code_Aster. Este tipo de algoritmo, devido ao carácter de código aberto do *software*, permite ser completamente desenvolvido pelo programador de forma a implementá-lo em conjunto com a rotina principal. Assim, é possível realizar esta análise a partir de uma rotina, sem necessidade de interação do utilizador com o *solver*. As funções utilizadas são as necessárias para a realização de uma típica análise de elementos finitos e encontram-se presentes numa vasta documentação disponibilizada pela EDF. Este algoritmo ainda permite integrar funções da linguagem Python na sua rotina o que possibilita a leitura de dados a partir de um ficheiro *json* ou ainda a utilização de operadores lógicos, [22].

Um exemplo desta implementação encontra-se presente no código 4.4. Este código é referente a uma análise com elementos de placa, de um modelo de um quadrado de espessura fina, encastrado nos bordos, sujeito a uma pressão uniforme normal à face. O *output* deste exemplo é referente a um ficheiro que permite visualização gráfica do campo de deslocamentos.

Código 4.4: Rotina para cálculo por elementos finitos no Code_Aster

```

DEBUT()

import os
import json

with open(os.path.join(os.getcwd(), 'Dados.json')) as f:
    dados = json.load(f)
    f.close()
for read in dados['Geometria']:
    espessura = float(read['espessura'])
for read in dados['Material']:
    young = float(read['young'])
    poisson = float(read['poisson'])
for read in dados['Condicoes']:
    pressao = float(read['pressao'])

mesh = LIRE_MALLAGE(UNITE=20)

model = AFFE_MODELE(AFFE=_F(MODELISATION=('DKT', ),
                             PHENOMENE='MECANIQUE', TOUT='OUI'),
                   MAILLAGE=mesh)

elemprop = AFFE_CARA_ELEM(COQUE=_F(EPAIS=espessura,
                                   GROUP_MA=('grupo_face', )),
                          MODELE=model)

steel = DEFI_MATERIAU(ELAS=_F(E=young, NU=poisson))

fieldmat = AFFE_MATERIAU(AFFE=_F(MATER=(steel, ), TOUT='OUI'),
                          MODELE=model)

load = AFFE_CHAR_MECA(DDL_IMPO=( _F(GROUP_MA=('grupo_linha_1', 'grupo_linha_2',
                                             'grupo_linha_3', 'grupo_linha_4'),
                                   LIAISON='ENCASTRE')),
                      FORCE_COQUE=_F(GROUP_MA=('grupo_face'), PRES=pressao),
                      MODELE=model)

statics = MECA_STATIQUE(CARA_ELEM=elemprop,
                        CHAM_MATER=fieldmat,
                        EXCIT=_F(CHARGE=load),
                        MODELE=model)

IMPR_RESU(FORMAT='MED',
          RESU=( _F(RESULTAT=statics)),
          UNITE=80)

FIN()

```

4.2.4 Arquitetura do ficheiro *export*

Como já referido, o ficheiro *export* indica ao Code_Aster em primeiro lugar parâmetros do estudo e em segundo, os ficheiros necessários à análise. A construção

deste ficheiro é realizada a partir da rotina, onde é possível manipular os dados que posteriormente serão utilizados. Como este ficheiro é só de leitura, portanto não é uma rotina, não permite qualquer tipo de programação. A solução é escrever este ficheiro como uma variável de texto, onde se substitui os valores que são dependentes do estudo pelos dados pretendidos.

O ficheiro *export* também tem uma nomenclatura própria, presente na documentação do Code_Aster, que identifica um parâmetro pela letra P e um ficheiro pela letra F. A escrita deste ficheiro segue a seguinte caracterização, [33]:

- P nome_do_parâmetro valor_do_parâmetro
- F extensão localização tipo logical_unit_number

No caso dos ficheiros, eles ainda são identificados quanto ao tipo pela letra D caso seja de *input* ou R caso seja de *output*. Os ficheiros de *input* serão o ficheiro *comm* com a rotina de análise e o ficheiro *med* com as informações da malha e os ficheiros de *output* serão o ficheiro *mess* com o relatório da análise e o ficheiro *med* com informações dos resultados. Um exemplo de um ficheiro *export* pode ser visualizado no código 4.5, [33].

Código 4.5: Estrutura de um ficheiro *export*

```
P actions make_etude
P memjob 512000
P memory_limit 32768
P mode interactif
P mpi_nbcpu 1
P ncpus 1
P nomjob RunCase
P time_limit 3600
P tpsjob 60
P version stable
P rep_trav path_folder_temp
F comm path_comm D 1
F libr path_mesh D 20
F libr path_results R 80
F mess path_mess R 6
```

No código 4.5 *path_folder_temp* representa a localização da pasta temporária que aloca ficheiros criados durante a análise. Os diretórios *path_comm*, *path_mesh*,

`path_results` e `path_mess` representam a localização do ficheiro `comm`, do ficheiro `med` com informação da malha, do ficheiro `med` com os resultados e ainda o ficheiro `mess`, respetivamente.

4.2.5 Arquitetura do ficheiro *py* para pós-processamento

O ficheiro Python para pós-processamento, como já referido, será somente necessário se for do interesse do programa visualizar no Salome-Meca os resultados obtidos. Esta rotina, à semelhança do ficheiro para pré-processamento, utiliza livrarias de funções do próprio *software* portanto terá de ser executado nesse ambiente. Este ficheiro já existe previamente e a partir do ficheiro `med`, resultado da análise em `Code_Aster`, permite que o utilizador visualize e analise os resultados através da representação em um mapa de cores. No código 4.6 está presente um exemplo da implementação deste ficheiro.

Código 4.6: Rotina para visualizar resultados no Salome-Meca

```
# -*- coding: utf-8 -*-

import salome
import pvsimple
import os

salome.salome_init()
estudo = salome.myStudy

diretorio_med = os.path.join(os.getcwd(), 'Resultados.med')

pvsimple.ShowParaviewView()
pvsimple._DisableFirstRenderCameraReset()
paraViewrmed = MEDReader(FileName=diretorio_med)
```

4.3 Processos da análise

Após a leitura dos dados de entrada, criação dos ficheiros necessários ao estudo e preparação de algumas rotinas e ficheiros para uma análise completa já é possível executar o pré-processamento, o cálculo por elementos finitos e o pós-processamento. Para isso, terá de se acrescentar ao código desenvolvido na rotina principal, a execução dos *softwares* que permitem realizar todos estes módulos assim como os ficheiros necessários a esta execução. Isto é possível através da livreria do Python `subprocess` e ainda a partir dos ficheiros *batch* de cada *software* assim como os argumentos necessários a cada processo.

No caso do módulo de pré-processamento, ele será executado a partir do ficheiro *batch* do Salome-Meca juntamente com a rotina de pré-processamento, que de seguida é implementada pelo próprio programa.

Assim que o pré-processamento é finalizado, que a rotina identifica quando o modelo é guardado no ficheiro *hdf*, a rotina principal executa o ficheiro *batch* do Code_Aster juntamente com o ficheiro *export*. O ficheiro *export* permite assim que o *software* inicie a análise através da rotina de cálculo estabelecida no ficheiro *comm* e ainda a partir de todos os ficheiros necessários de *input* e *output*.

Por fim, quando a rotina identifica que o ficheiro *mess* se encontra escrito, já é possível executar o pós-processamento dos resultados. Caso se pretenda a visualização é necessário executar novamente o ficheiro *batch* do Salome-Meca assim como a rotina de pós-processamento.

No código 4.7 é possível observar as linhas de código necessárias.

Código 4.7: Rotina para executar os processos da análise

```
import subprocess
import os

# executar pre-processamento
subprocess.call(path_salome_run + '\_ ' + path_pre_process, shell=True)
checker = False
while checker is False:
    checker = os.path.isfile(path_hdf)

# executar solver
subprocess.call(path_code_aster_run + '\_ ' + path_export, shell=True)
checker = False
while checker is False:
    checker = os.path.isfile(path_mess)

# executar pos-processamento
subprocess.call(path_salome_run + '\_ ' + path_pos_process, shell=True)
```

No código 4.7, *path_salome_run* e *path_code_aster_run* correspondem à localização do ficheiro *batch* do Salome-Meca e do Code_Aster respetivamente. Os diretórios *path_pre_process*, *path_export*, *path_pos_process* correspondem à localização do ficheiro *py* para pré-processamento, ficheiro *export* e ficheiro *py* para pós-processamento, respetivamente. E por fim, *path_hdf* e *path_mess* correspondem aos diretórios do ficheiro *hdf* e do ficheiro *mess* e indicam à rotina quando os processos se encontram finalizados para prosseguir.

Nas figuras 4.2, 4.3 e 4.4 encontram-se os fluxogramas de cada processo. Estes processos já se encontravam representados no fluxograma da rotina principal na figura 4.1 como sub-processos, no entanto segue-se uma visualização mais detalhada da rotina de cada módulo.

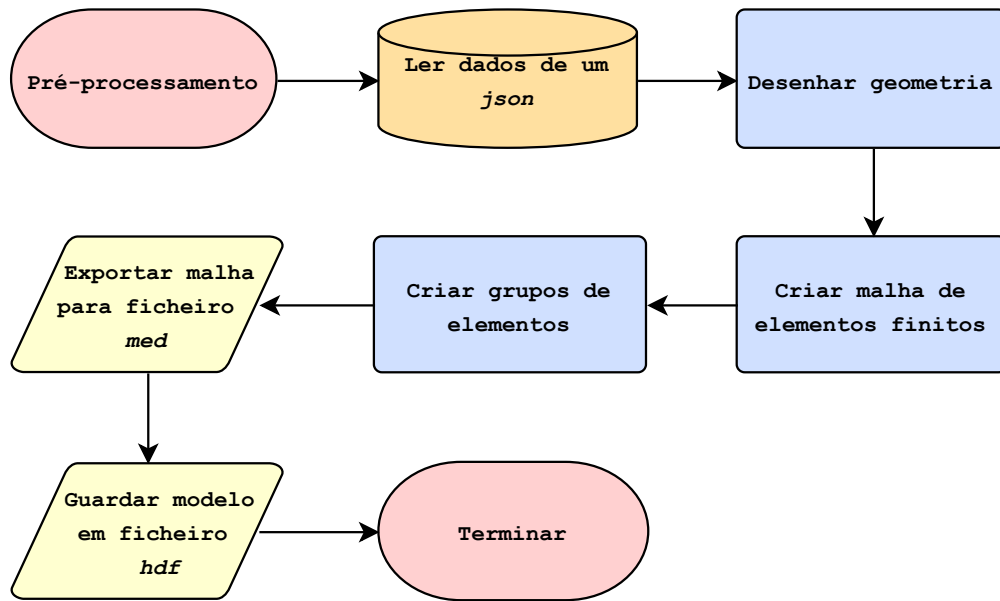


Figura 4.2: Fluxograma da rotina de pré-processamento.

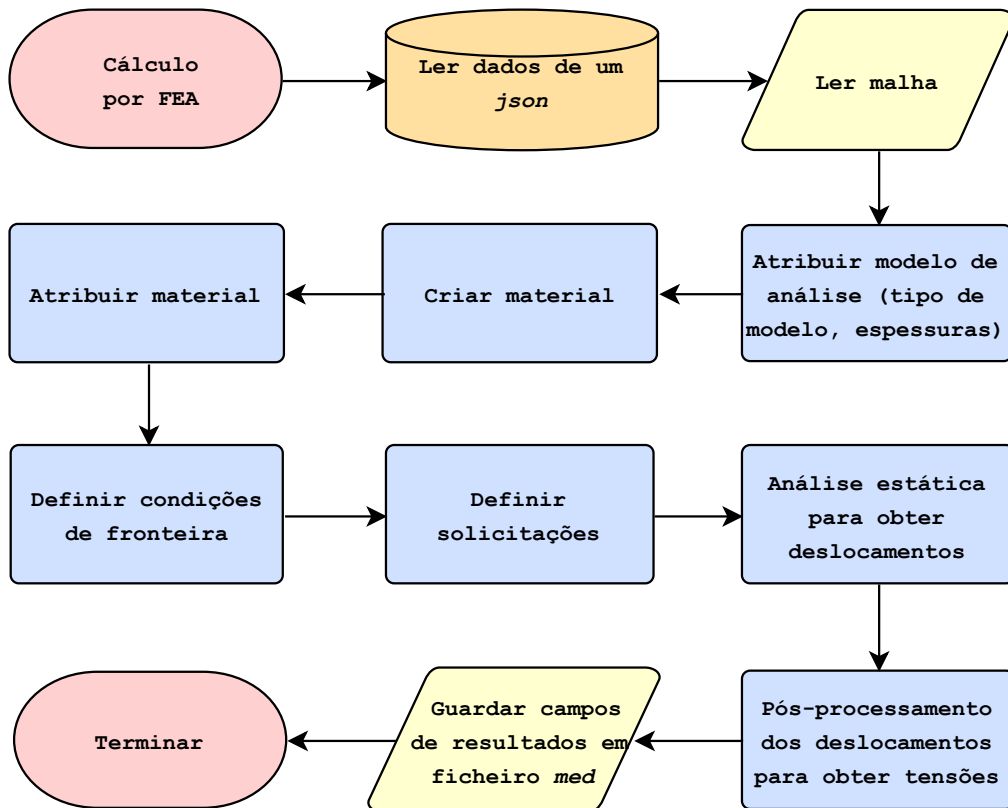


Figura 4.3: Fluxograma da rotina de cálculo.

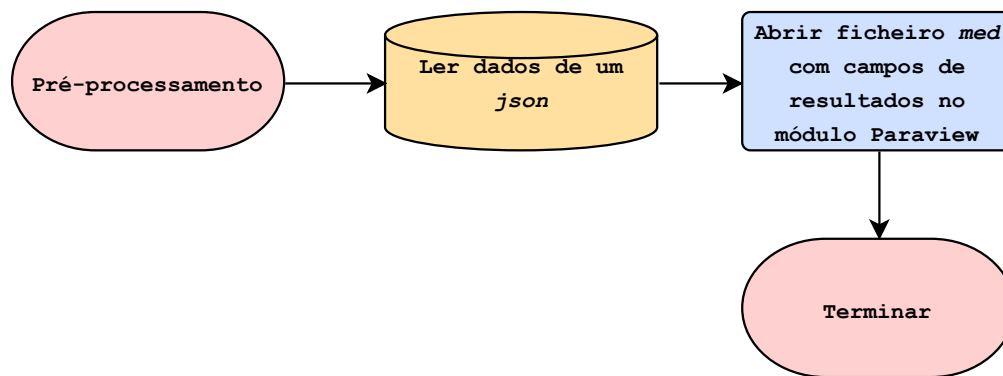


Figura 4.4: Fluxograma da rotina de pós-processamento para visualização.

4.4 Discussão da rotina

Neste capítulo foi apresentada a arquitetura de uma rotina que permite a análise completa por elementos finitos de um modelo estrutural de forma automática e com mínima interação com o utilizador.

O tipo de algoritmo abordado permite realizar análises mais rápidas visto que todo o pré-processamento do modelo, assim como a criação da rotina de cálculo e por fim o pós-processamento é realizado pelo programa. Também é possível alterar a rotina para incluir análises paramétricas mais eficientes, em que se inclui ciclos que permitem iterar dados de entrada para perceber a sua influência nos resultados finais. Por fim, também se apresenta como vantagens deste tipo de rotinas, uma implementação que permite aos utilizadores realizarem vários tipos de cálculo de forma expedita com vista à obtenção de resultados para variadas configurações e condições. Contudo, a implementação deste tipo de rotinas também inclui alguns obstáculos, sendo o principal a pouca versatilidade em relação a mudanças de geometria. Visto que a geometria é criada a partir de código, só será possível gerar a geometria que o programador desenvolveu, portanto, caso seja necessário alterar drasticamente o tipo de construção, ou terá de se desenvolver outra rotina ou alterar a existente.

Em capítulos posteriores serão apresentados exemplos da implementação desta rotina aplicada a casos reais. O primeiro exemplo é em relação à análise de painéis de uma cuba para transformador onde se explora a capacidade de analisar vários modelos paramétricos simultaneamente para perceber qual a construção mais indicada. O segundo exemplo apresentado será a análise de uma cuba completa, onde se abordam geometrias mais complexas e a capacidade de visualização gráfica do modelo.

Capítulo 5

Análise Estrutural de Painéis

Este capítulo aborda a rotina de análise otimizada referida no capítulo 4, tratando o caso de análise estruturais de painéis de cubas de transformadores. Esta rotina permite assim realizar análises de painéis a partir de um algoritmo em Python que integra o *software* de elementos finitos Salome-Meca e desta forma todos os processos relativos a análise completa deste componentes encontram-se automatizados. Também foi possível implementar a análise de vários modelos de painéis simultaneamente e um tratamento dos resultados obtidos de forma a obter qual o modelo preferível de acordo com os critérios aplicados. Desta forma, é possível otimizar tanto a análise como o modelo final, visto que, facilmente e em pouco tempo, se obtém os resultados dos cálculos por elementos finitos assim como a construção que se adequa aos critérios indicados.

Também neste capítulo irão ser analisadas diferentes construções de painéis de cubas disponíveis na rotina implementada. A tipologia destas construções considera, em primeiro lugar, se o painel é curvo ou plano e ainda se o painel inclui reforços verticais ou reforço horizontal. Para cada tipo de construção irão ser apresentadas vários tipos de validação de exemplos, comparando-os ou com bibliografia disponível ou com análises em outros *softwares* de elementos finitos. Também irão ser realizadas análises paramétricas de forma a avaliar a influência de diferentes fatores nos resultados finais.

Por fim, irão ser realizados alguns exemplos de otimização de painéis com uso da rotina criada de forma a validar a sua capacidade de analisar diferentes construções, de acordo com critérios adotados, que minimizam a quantidade de material necessário mantendo um comportamento mecânico dentro de intervalos pretendidos.

5.1 Rotina de análise de painéis otimizada

A partir da rotina abordada no capítulo 4 foi possível implementá-la para obter uma análise por elementos finitos de painéis de vários modelos em simultâneo, em que o utilizador só terá de fornecer os dados de entrada para obter tanto o campo de deslocamentos como o campo de tensões. Este algoritmo permite assim realizar cálculos estruturais mais eficientes e em menor tempo, visto que, a implementação no *software* Salome-Meca de cada modelo separadamente e manualmente implicava que o utilizador realizasse todos os processos da análise desde o princípio. Portanto, a partir desta rotina, todos os passos de pré-processamento, cálculo e pós-processamento são realizados automaticamente através de algoritmos já estabelecidos e uma modelação paramétrica.

O pré-processamento de todos os modelos diz respeito à criação de várias malhas de elementos finitos pelo código desenvolvido, em que se variam propriedades geométricas de cada modelo a partir de iterações e ainda é possível indicar qual o tipo de construção, painéis planos ou painéis curvos. Neste processo, a rotina é capaz de variar parâmetros como o número de reforços verticais, a altura dos reforços verticais e a altura do reforço horizontal. Após o pré-processamento, é realizado o cálculo por elementos finitos de cada modelo, onde novamente é possível realizar novas iterações para obter diferentes análises em que se varia a espessura do painel, a espessura dos reforços verticais e a espessura do reforço horizontal. Também neste cálculo o utilizador pode indicar como dado de entrada quais as condições de fronteira das possíveis que pretende aplicar ao painel. Por fim, é realizado um pós-processamento em que é possível observar os resultados através da ferramenta ParaView. Também se realiza um tratamento dos resultados obtidos em que a rotina lê os valores máximos de cada modelo e de seguida organiza num relatório final um resumo deste resultados. Neste processo também é possível avaliar os valores máximos quanto a critérios definidos, como critério de von Mises, critério da flecha máxima e ainda obter o modelo com menor massa. Assim esta rotina é capaz de indicar ao utilizador, qual o modelo dos analisados que respeita da melhor forma estes critérios.

De seguida encontra-se uma explicação dos dados de entrada necessários, o pré-processamento realizado, o tipo de cálculo e por fim uma explicação do pós-processamento e dos dados de saída.

5.1.1 Dados de entrada

Os dados de entrada deste programa encontram-se organizados em classes em que cada uma tem os respectivos atributos e métodos. Cada classe é respeitante a um componente do estudo sendo que podem ser dados relativamente às dimensões geométricas, à malha, ao material, às condições do cálculo e ainda aos ficheiros criados. Na figura 5.1 é possível observar cada classe com os seus atributos e os métodos.

Os dados desta rotina podem então ser classificados em três grandes tipologias: (i) variáveis protegidas, em que o atributo já é inicializado com o valor que somente pode ser alterado pelos métodos da sua classe; (ii) variáveis de entrada iteráveis, em que o utilizador fornece em forma de lista um conjunto de valores que permite criar diferentes modelos na mesma rotina; (iii) as variáveis de entrada com valor único, em que o utilizador inicia os atributos através de *input* do seu valor.

Em relação às variáveis protegidas identificam-se os seguintes dados e os valores que podem ser utilizados:

- Tipo de construção: painel plano ou painel curvo;
- Condições de fronteira: encastramento, simplesmente apoiado nas arestas, simplesmente apoiado nos vértices ou simplesmente apoiado nas arestas para caso do painel curvo;
- Critérios de validação ativos;
- Diretórios dos ficheiros que irão ser utilizados.

As variáveis de entrada iteráveis permitem criar modelos geométricos com todas as combinações dos valores das listas fornecidas e são responsáveis pela possibilidade de analisar na mesma rotina vários modelos. Os dados que se identificam como este tipo são:

- Espessuras dos painéis;
- Números de reforços verticais;
- Espessuras dos reforços verticais;
- Alturas dos reforços verticais;
- Espessuras dos reforços horizontais;
- Alturas dos reforços horizontais.

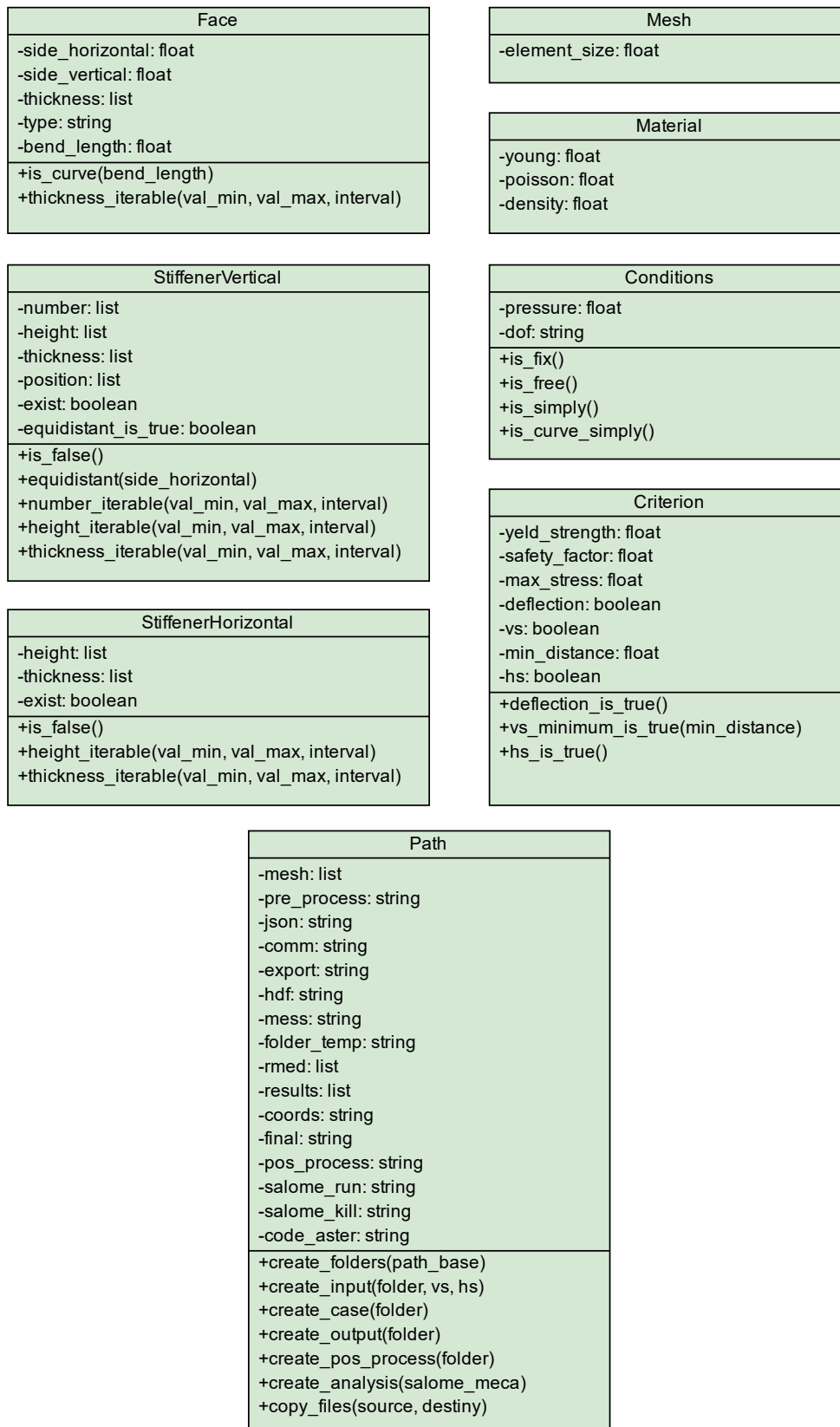


Figura 5.1: Classes da rotina de análise de painéis.

Por fim, as variáveis de entrada com valor único são todas as que são necessárias que o utilizador forneça *input* para serem inicializadas e indicam-se como principais as seguintes:

- Comprimento e largura do painel e flecha de curvatura no caso do painel curvo;
- Posição dos reforços verticais para cada número de reforços caso não sejam equidistantes;
- Tamanho máximo do elemento finito para gerar a malha;
- Propriedades do material: módulo de Young, coeficiente de Poisson e massa volúmica;
- Pressão exercida sobre o painel;
- Critérios mecânicos: tensão de cedência e coeficiente de segurança.

Cada classe também tem métodos próprios que permitem alterar e criar dados de entrada. Os métodos disponíveis nesta rotina serão os seguintes:

- Modificar o tipo de construção para curvo;
- Métodos relativos a cada variável iterável que permite criar uma lista com um valor mínimo, um valor máximo e um incremento;
- Métodos para tanto os reforços verticais como para o reforço horizontal que indicam que estes componentes não existem;
- Modificar a lista de posições para cada número de reforços verticais para valores que tornam os reforços equidistantes;
- Modificar o valor da variável protegida das condições de fronteira a aplicar;
- Métodos que indicam quais os critérios de validação ativos;
- Métodos para a criação de cada ficheiro necessário.

Após o utilizador fornecer cada *input* necessário, o programa já se encontra com a capacidade de começar a rotina para obter os resultados do cálculo por elementos finitos sem que seja necessário mais algum procedimento em que o utilizador tenha contacto com o resto dos processos.

Para além de fornecer os dados de entrada, o algoritmo terá também de inicializar os ficheiros que serão utilizados como abordado no capítulo 4. A criação destes

ficheiros assim como a identificação de alguns diretórios é realizada a partir dos métodos da classe que guarda os diretórios em forma de texto nas variáveis respetivas a cada ficheiro. Todos os ficheiros criados também são guardados numa pasta respetiva ao estudo realizado. Como se irá obter vários modelos e vários resultados na mesma rotina, é necessário criar um ficheiro *med* para *input* no algoritmo de cálculo para cada malha gerada, e ainda um ficheiro *med* para *output* para cada modelo analisado que podem ser mais do que os ficheiros para *input* devido ao facto de uma malha ter a capacidade de gerar vários modelos com diferentes espessuras de cada componente.

O programa cria mais dois ficheiros do tipo texto (*txt*) que permitem escrever um relatório final de todos os resultados obtidos para no fim da análise o utilizador ter uma visão abrangente e resumida do comportamento de cada modelo. Estes ficheiros e o seu objetivo serão os seguintes:

- Ficheiro do tipo *txt* em que se escreve os dados gerais dos modelos e de seguida, para cada modelo, indica-se os parâmetros que se variaram, o valor máximo do deslocamento segundo a normal ao painel, a tensão máxima de von Mises, os nós onde estão localizados cada valor máximo e ainda a massa do modelo. Também é indicado o modelo, dos analisados, que respeita todos os critérios definidos e com menor massa;
- Ficheiro do tipo *txt* em que se escreve para cada modelo as coordenadas dos nós onde ocorrem os valores máximos.

Após a rotina ler os dados de entrada e inicializar o resto dos dados necessários, escreve a informação respeitante a cada classe num ficheiro *json* que permite mais tarde ser lido por rotinas paralelas que serão executadas.

5.1.2 Preparação de ficheiros

Após a leitura de dados e iniciação dos ficheiros, a rotina executa uma preparação dos ficheiros que irão ser necessários nos processos subsequentes. Os ficheiros escritos pela rotina serão o ficheiro *json* com a informação de todos os dados de entrada, o ficheiro *comm* com a rotina de cálculo e o ficheiro *export* com as informações relativas ao estudo que o Code_Aster necessita. Ainda é necessário que o programador desenvolva antes de iniciar a rotina, os algoritmos de pré e pós-processamento.

5.1.2.1 Algoritmo de pré-processamento

O algoritmo de pré-processamento é responsável por gerar os modelos geométricos e as malhas de elementos finitos. Este algoritmo lê os dados geométricos através do ficheiro *json* e subsequentemente, a partir de funções Python respetivas ao módulo geométrico do Salome, cria a face do painel, os reforços verticais para cada altura e número de reforços introduzido, e ainda o reforço horizontal para cada altura introduzida. Por fim, junta todas as combinações destes componentes para criar diferentes modelos geométricos. De salientar que são criadas partições na geometria para a malha gerada ter o maior número de elementos quadrangulares e ainda possibilitar a criação de grupos de elementos sobre essas partições. Na figura 5.2 é possível visualizar um exemplo dos quatro tipos de construção de painéis planos que o algoritmo é capaz de gerar e na figura 5.3 os mesmos tipos para o caso de painéis curvos.

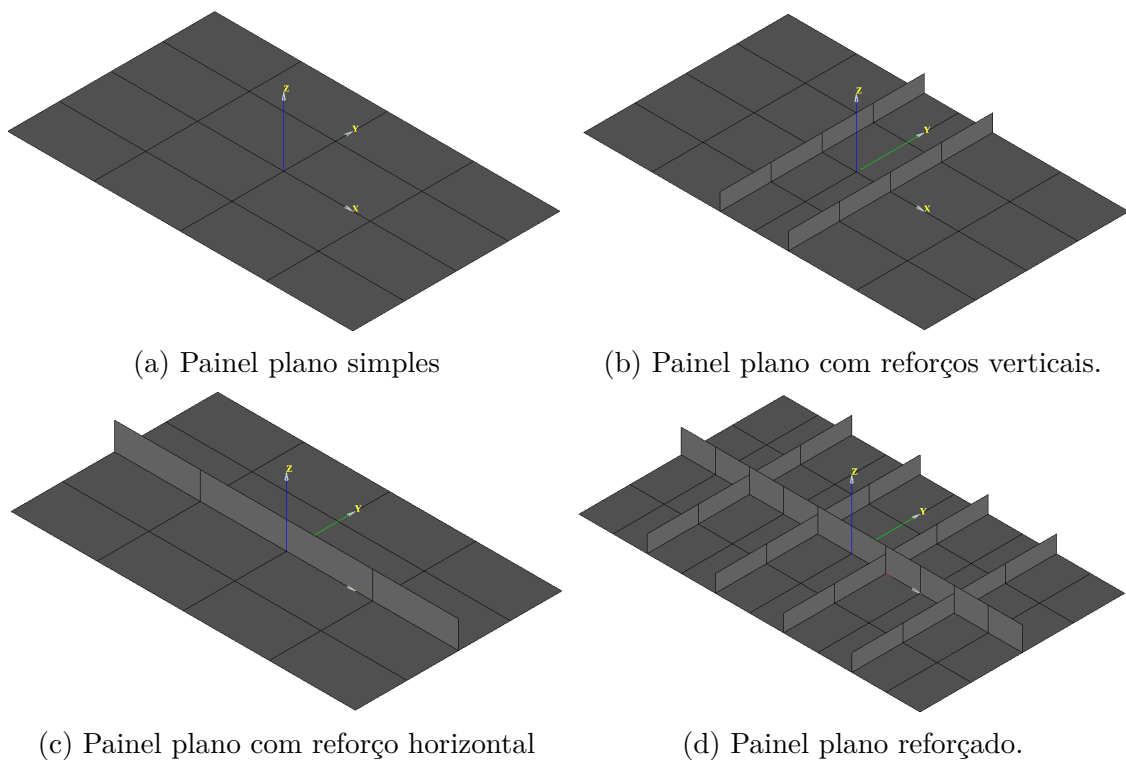


Figura 5.2: Tipos de geometria de painéis planos.

Após cada geometria se encontrar definida o algoritmo prossegue com a geração da malha de elementos finitos para cada modelo. Os elementos utilizados são quadrangulares quadráticos, ou seja com oito nós cada um, designados no âmbito do *software* como QUAD8. Neste processo também é gerado, para cada malha, grupos

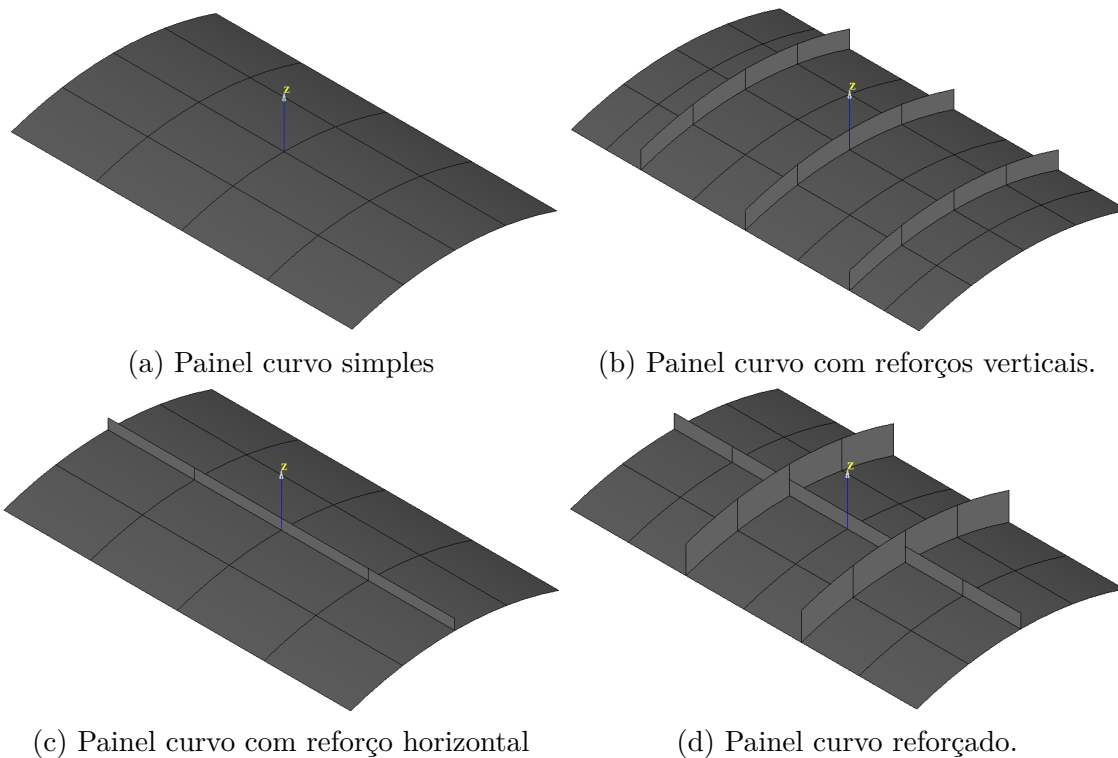


Figura 5.3: Tipos de geometria de painéis curvos.

de elementos de forma a possibilitar no cálculo posterior atribuir propriedades a estes grupos e realizar análises. Os grupos criados são:

- Vértices da face do painel;
- Arestas da face do painel;
- Face do painel;
- Faces dos reforços verticais;
- Face do reforço horizontal;
- Linhas de partição para análise da tensão de von Mises.

O grupo que indica as linhas para análise da tensão de von Mises permite, no pós-processamento, só visualizar os resultados da tensão equivalente nestes elementos de forma a evitar que a rotina indique como valor máximo locais onde ocorra picos de tensão que não representam a realidade. As linhas pertencentes a este grupo foram validadas para diferentes casos como os locais onde a tensão máxima equivalente de von Mises ocorre como é demonstrado em secções posteriores.

Por fim, o algoritmo terá de incluir funções que exportam para cada ficheiro *med* respetivo a cada modelo, a malha de elementos finitos gerada assim como os grupos de elementos que inclui e ainda terá de incluir a função que guarda todos os modelos num único ficheiro *hdf*.

5.1.2.2 Algoritmo de análise por elementos finitos

O algoritmo de análise encontra-se alojado no ficheiro *comm* e este ficheiro é escrito durante a própria rotina com o algoritmo programado previamente. Esta rotina de análise permite calcular por elementos finitos o campo de deslocamentos, o campo de tensões e ainda os campos de tensões segundo critérios, como a tensão equivalente de von Mises.

Inicialmente a rotina lê a partir do ficheiro *json* todos os dados de entrada relativos à análise e a partir de ciclos permite analisar cada modelo gerado, sendo que a mesma malha de elementos finitos de entrada pode ser analisada com diferentes espessuras de cada componente.

Relativamente à análise propriamente dita o algoritmo começa por ler a malha a partir do *Logical Unit Number* atribuído no ficheiro *export* e de seguida cria outra malha em que transforma os elementos de oito nós gerados em elementos de nove nós, designados por QUAD9, visto que a modelação que se irá utilizar somente suporta este tipo de elementos. De seguida atribui-se a modelação que utiliza teoria das cascas, designada como COQUE_3D na linguagem do Code_Aster, à malha de elementos finitos com nove nós cada. No entanto, como esta modelação considera que o nó do meio do elemento QUAD9 só tem graus de liberdade de rotação, é necessário também criar uma modelação por elementos sólidos a partir da malha inicial de oito nós. Este último modelo tem como propósito somente projetar os resultados obtidos a partir da modelação por cascas de forma a que modelo visualizado no pós-processamento não apresente deslocamentos nulos no nó do meio de cada elemento. Assim, os resultados obtidos relativamente ao nó do meio serão desprezados, no entanto, este terá influência nos nós da fronteira do elemento devido às funções de forma utilizadas. Este tratamento dos resultados é proveniente do desenvolvimento da modelação COQUE_3D pelo próprio Code_Aster e é indicada como uma das principais desvantagens da utilização destes modelos, no entanto, é necessário utilizar esta modelação devido aos painéis curvos, já que desta forma é possível contabilizar a curvatura do painel, [24].

Após definir o modelo, é necessário definir as espessuras de cada componente

visto serem elementos casca, definir as propriedades do material utilizado e atribuí-lo ao modelo e ainda atribuir as condições de fronteira e as solicitações. A solicitação a que os painéis estão sujeitos é uma pressão aplicada à superfície do painel com direção positiva contrária à normal de cada elemento, definida pela a orientação da malha. Esta pressão é aplicada a partir da função designada no Code_Aster como `FORCE_COQUE` e o seu valor é indicado a partir dos dados de entrada, [31].

As condições de fronteira a aplicar também são indicadas pelos dados de entrada por uma variável de texto em que o algoritmo é capaz de identificar através de uma condição *if* qual as restrições a aplicar. As condições de fronteira disponíveis na rotina encontram-se explicitas na figura 5.4.

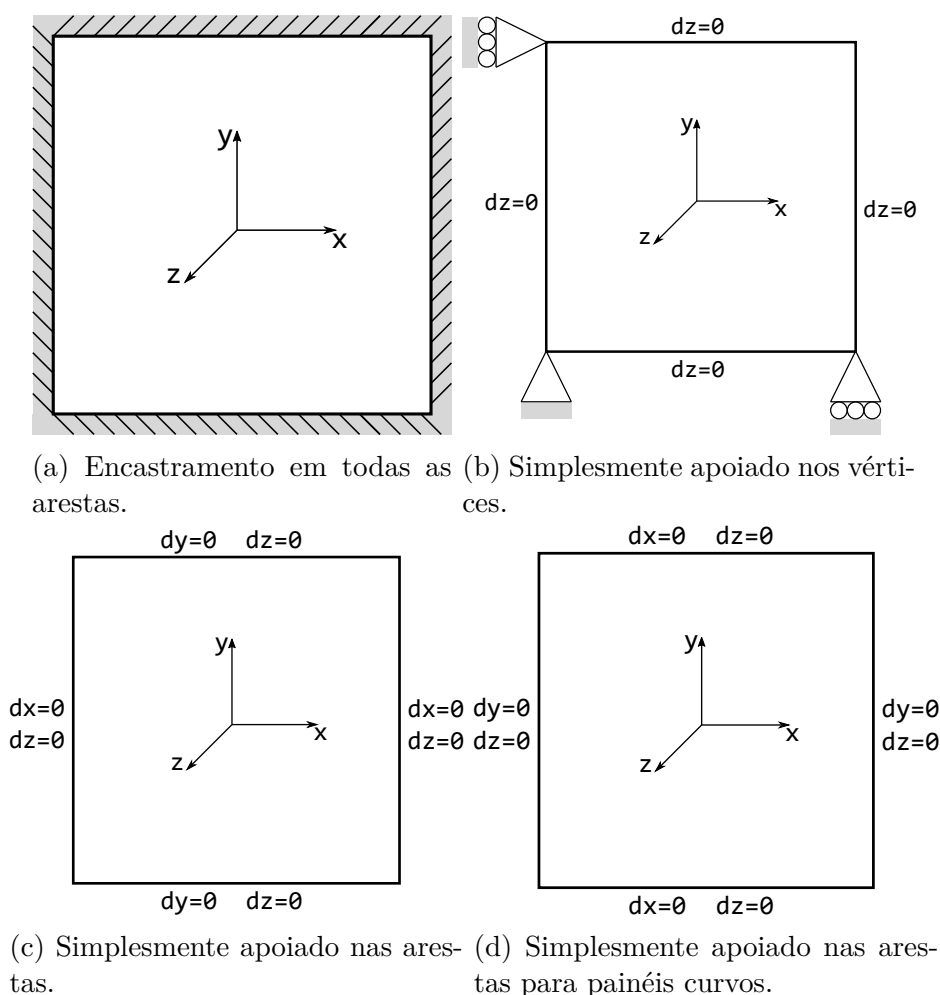


Figura 5.4: Tipos de condições de fronteira presentes na rotina.

A partir deste ponto, já é possível calcular os deslocamentos a partir de uma análise estática e de seguida a rotina projeta estes resultados no modelo com oito nós por elemento. Com os campos de deslocamentos, o algoritmo calcula as tensões

para o nó de cada elemento através da extrapolação das tensões obtidas nos pontos de Gauss de cada elemento. De seguida, extrai as tensões de cada nó de cada elemento para a camada superior da casca e realiza o cálculo deste campo para todos os nós do modelo através de uma média aritmética dos valores respetivos a diferentes elementos para o mesmo nó. À semelhança do campo de deslocamentos, o campo de tensões final é projetado no modelo com oito nós por elemento, [26, 27, 30].

Por fim, o algoritmo imprime dois ficheiros para cada modelo. Um deles é ficheiro do tipo *med*, designado pela extensão também possível de utilizar como ficheiro *rmed*, que permite transferir todos os campos de resultados obtidos para de seguida visualizar na ferramenta ParaView o modelo com mapas de cores. O outro ficheiro, também do tipo *med*, permite visualizar em forma de texto o valor máximo dos deslocamentos segundo a normal ao painel e o nó onde ocorre, o valor máximo da tensão equivalente de von Mises no grupo já identificado e o nó onde ocorre e por fim, imprime também as coordenadas de todos os nós do modelo. Este último ficheiro irá permitir que através de uma rotina posterior o programa leia os valores máximos para cada modelo e realize um relatório final de todas as análises. Estes ficheiros são identificados pela rotina de cálculo através dos seus *Logical Unite Number* presentes no ficheiro *export*, [32].

Esta rotina como referido é escrita no ficheiro *comm* e após ser realizado o pré-processamento é executada.

5.1.2.3 Ficheiro *export*

O ficheiro *export* é responsável pela ligação entre os ficheiros *input*, os ficheiros *output*, o ficheiro *comm* e o ficheiro *mess* com o Code_Aster através de *Logical Unite Numbers* e ainda fornece vários parâmetros ao estudo. Este ficheiro é escrito durante a execução da rotina principal a partir das variáveis que guardam a localização dos ficheiros referidos.

5.1.2.4 Algoritmo de pós-processamento

Nesta rotina, o pós-processamento divide-se em dois sub-processos, sendo que um será o tratamento dos valores máximos presentes no ficheiro *med* de cada modelo, e outro será a visualização dos campos de resultados através do ficheiro *rmed* de cada modelo.

O relatório final da rotina principal indica ao utilizador os dados gerais de todos os modelos, os dados específicos de cada modelo, os resultados máximos e ainda o modelo que melhor respeita os critérios indicados. Este ficheiro do tipo *txt* é escrito

pela rotina principal após a rotina de cálculo, no entanto terão de ser criadas previamente pelo programador algoritmos que tratam os dados provenientes da análise. Na figura 5.5 é possível visualizar um exemplo de um relatório para o caso de análise de painéis curvos reforçados na situação de encastramento, onde se varia o número de reforços verticais e a altura dos mesmos. No relatório também é indicado os nós onde ocorrem os valores máximos, sendo que as coordenadas destes nós podem ser verificadas num ficheiro de texto à parte também criado pela rotina com esta informação.

De forma a verificar qual o modelo mais indicado dos analisados estabelece-se alguns critérios, sendo que a rotina principal inclui algoritmos que são capazes de avaliar automaticamente para todos os modelos qual o mais otimizado. Os critérios que a rotina é capaz de verificar são os seguintes:

- Critério de cedência de von Mises;
- Critério da flecha máxima;
- Distancia mínima entre reforços verticais;
- Relação entre a altura e espessura dos reforços para não ocorrer encurvadura;
- Existência de reforço horizontal;
- Modelo com menor massa.

O critério de cedência de von Mises é válido para materiais dúcteis e indica que a tensão de von Mises (σ_{vmis}) terá de ser menor que a tensão admissível (σ_{adm}). Sendo que a tensão admissível será a tensão de cedência do material (σ_Y) a dividir pelo coeficiente de segurança (n). Assim, no presente algoritmo estabelece-se que, [6]:

$$\sigma_{vmis} \leq \sigma_{adm} = \frac{\sigma_Y}{n} \quad (5.1)$$

A tensão de von Mises é proveniente da análise por elementos finitos de cada modelo, e a tensão de cedência e coeficiente de segurança são dados de entrada do utilizador.

O critério da flecha máxima é um pressuposto que o deslocamento normal ao painel nunca poderá ser maior do que a espessura do painel. No entanto, este critério, em alguns casos, é mais conservador que o critério de cedência de von Mises então admite-se que este último tem maior importância. Portanto nos dados de entrada o critério de flecha máxima poderá ser desativado.

RELATÓRIO FINAL

Painel:

Tipo de painel: Curvo
 Comprimento do lado horizontal [mm]: 3500.0
 Comprimento do lado vertical [mm]: 2000.0
 Flecha de arco [mm]: 50.0
 Reforços Verticais (RV):
 Os reforços verticais são equidistantes.
 Malha de elementos finitos:
 Tamanho de elemento [mm]: 20.0

Material:

Módulo de Young [MPa]: 210000.0
 Coeficiente de Poisson: 0.3
 Densidade [kg/m³]: 7850.0

Análise:

Solicitação [MPa]: -0.1
 Condições de fronteira: Encastramento

Critérios:

Tensão de cedência [MPa]: 355.0
 Coeficiente de segurança: 1.2
 A flecha terá de ser menor que a espessura.
 Distância mínima entre reforços [mm]: 200.0
 O modelo ótimo terá de ter reforço horizontal.
 O modelo ótimo será o modelo com menor massa.
 Tempo de execução [s]: 392.9

Modelo ótimo:

Alt. RH [mm]	Alt. RV [mm]	Nº RV	Esp. RH [mm]	Esp. RV [mm]	Esp. Pain. [mm]	DZ [mm]	NÓ	VMIS [MPa]	NÓ	Massa [kg]
140	140	04	8	8	8	-4.736	N24468	289.825	N24939	541.557

Resultados:

Alt. RH [mm]	Alt. RV [mm]	Nº RV	Esp. RH [mm]	Esp. RV [mm]	Esp. Pain. [mm]	DZ [mm]	NÓ	VMIS [MPa]	NÓ	Massa [kg]
140	120	03	8	8	8	-5.919	N47538	360.532	N23175	516.396
140	120	04	8	8	8	-5.122	N2356	326.506	N25396	531.493
140	120	05	8	8	8	-4.405	N44910	313.153	N26242	546.59
140	140	03	8	8	8	-5.739	N47096	321.671	N23289	523.944
140	140	04	8	8	8	-4.736	N24468	289.825	N24939	541.557
140	140	05	8	8	8	-3.9	N25113	273.859	N25559	559.171

Figura 5.5: Relatório final de uma análise de vários modelos de painéis curvos reforçados.

Em relação à distância mínima entre reforços verticais e verificação da existência de reforço horizontal estes são critérios construtivos que permitem ao utilizador filtrar modelos que não os respeitem, no entanto, igualmente poderão ser desativados antes de se iniciar o programa. Ainda é definido um critério que verifica o rácio entre a altura e a espessura dos reforços de modo a que não ocorra o fenómeno de encurvadura e o seu valor máximo aconselhado é 20. Por fim, a partir dos modelos que verificam os critérios indicados, o algoritmo admite que o modelo mais otimizado será aquele com menor massa, visto ser um aspeto diferenciador tanto por razões construtivas como por razões económicas.

No âmbito do algoritmo de pós-processamento, estabelece-se também uma rotina de leitura dos ficheiros *rmed* que permite visualizar no módulo ParaView do Salome-Meca o mapa de cores dos campos de resultados obtidos para todos os modelos analisados. A informação da localização dos ficheiros *rmed* encontra-se no ficheiro *json* que a rotina lê e identifica os respetivos ficheiros. Para o exemplo ótimo presente na figura 5.5 é possível visualizar o mapa de cores da distribuição da tensão de von Mises na figura 5.6.

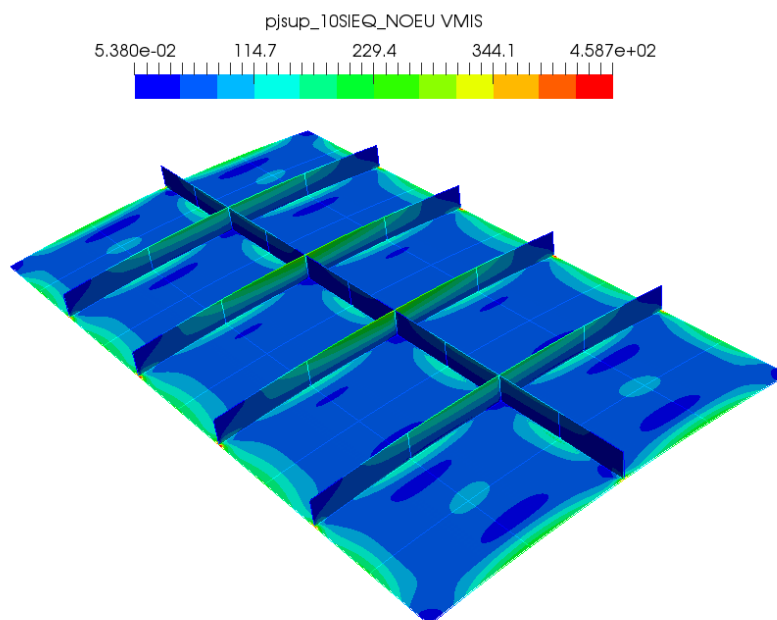


Figura 5.6: Exemplo da representação gráfica da distribuição da tensão de von Mises [MPa].

5.1.3 Processos da análise

Após ler os dados de entrada, iniciar os ficheiros necessários e estabelecer as rotinas que irão ser utilizadas o programa já se encontra capaz de executar os processos responsáveis por cada componente do estudo. Assim, em primeiro lugar executa o algoritmo de pré-processamento no ambiente do Salome-Meca, de seguida executa o ficheiro *export* no ambiente do Code_Aster e por fim executa o algoritmo para visualização de resultados no Salome-Meca assim como o algoritmo para redigir o relatório final.

A rotina de análise de painéis otimizada fica assim estabelecida, indicando-se como principais vantagens uma análise por elementos finitos bastante mais rápida sendo que existe a possibilidade de obter uma análise paramétrica mais eficiente.

5.2 Painéis planos simples

Um caso particular dos painéis planos é quando não existem reforços, sendo que este exemplo encontra-se bem documentado na literatura para vários casos. Na figura 5.7 encontra-se uma representação geométrica de um painel plano simples assim como a solicitação a que está sujeito.

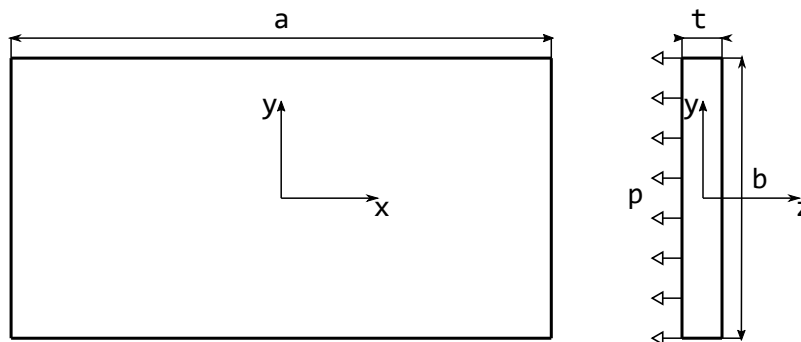


Figura 5.7: Representação geométrica de um painel plano simples.

Na figura 5.7, a será o comprimento horizontal do painel, b o comprimento vertical, t a espessura do painel e p a pressão a que está sujeito. A pressão será a situação quando o painel está sujeito a vácuo, portanto terá o valor de 0,1 MPa com a mesma direção que a representada na figura 5.7.

5.2.1 Fundamentos teóricos

A análise teórica deste tipo de exemplos pode ser abordada pela Teoria das Placas. Quando existe uma solicitação que não seja no plano da placa irá resultar numa situação de flexão e correspondente deslocamento de todos os pontos do painel. A Teoria das Placas fundamenta-se nas hipóteses de Kirchhoff que se encontram a seguir explicitadas, [18]:

- O deslocamento do plano médio é pequeno quando comparado com a espessura da placa.
- As deformação do plano médio serão nulas durante a flexão.
- Secções planas normais ao plano neutro permanecem planas após a flexão. Isto resulta em esforços de corte verticais (γ_{xz} e γ_{yz}) nulos.
- A tensão normal à placa (σ_z) será bastante pequena quando comparadas com as outras tensões normais, portanto negligenciável.

Portanto no caso de flexão de uma placa fina, elástica, homogênea e isotrópica sendo os deslocamentos pequenos é possível caracterizar este comportamento através das equações respectivas. Um exemplo desta geometria pode ser visualizado em:

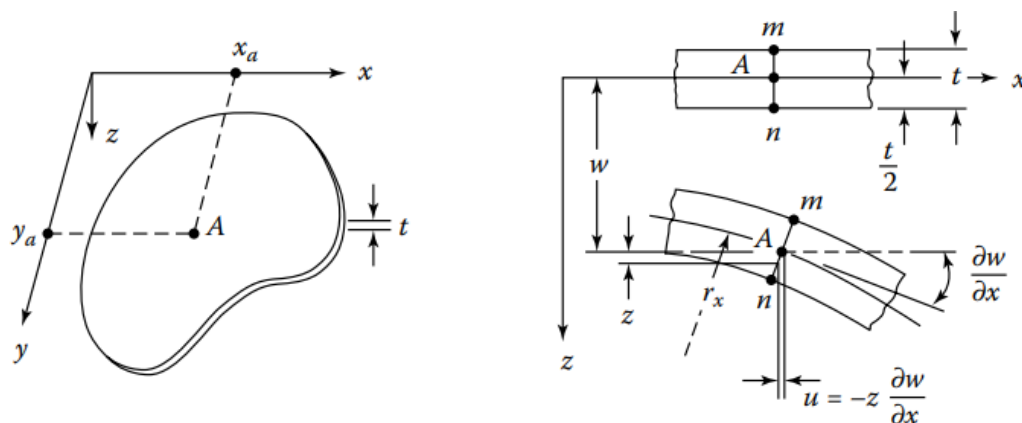


Figura 5.8: Representação de uma placa e respetivo referencial (esquerda) e representação de uma secção lateral antes e depois da solicitação (direita), [18].

O campo de deslocamentos (u) de uma placa é definido pelo deslocamento na direção horizontal, segundo x , (u), pelo deslocamento na direção vertical, segundo

y , (v) e ainda pelo deslocamento na direção normal à placa, segundo z , (w). Este vetor é caracterizado pelos seguintes componentes:

$$\mathbf{u} = \begin{Bmatrix} u(x, y, z) \\ v(x, y, z) \\ w(x, y, z) \end{Bmatrix} = \begin{Bmatrix} -z \frac{\partial w}{\partial x} \\ -z \frac{\partial w}{\partial y} \\ w(x, y) \end{Bmatrix} \quad (5.2)$$

O campo de deformações (ε) é definido pelas deformações normais, segundo a direção x (ε_x), segundo a direção y (ε_y) e segundo a direção z (ε_z) e ainda pelas deformações de corte segundo todas as direções ($\gamma_{xz} = \gamma_{zx}$, $\gamma_{xy} = \gamma_{yx}$, $\gamma_{yz} = \gamma_{zy}$). Como já referido pelas hipóteses de Kirchhoff tanto a deformação normal ε_z , como as deformações de corte γ_{yz} e γ_{xz} serão nulas, portanto o campo de deformações resulta nos seguintes componentes:

$$\varepsilon = \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} = \begin{Bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \end{Bmatrix} \quad (5.3)$$

Substituindo as equações de 5.2 nas equações 5.3 é possível reescrever o campo de deformações como:

$$\varepsilon = \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} = \begin{Bmatrix} -z \frac{\partial^2 w}{\partial x^2} \\ -z \frac{\partial^2 w}{\partial y^2} \\ -2z \frac{\partial^2 w}{\partial x \partial y} \end{Bmatrix} \quad (5.4)$$

As leis constitutivas do problema são representadas pelas leis de Hooke que para um material isotrópico homogêneo são as seguintes, [18]:

$$\begin{aligned} \varepsilon_x &= \frac{1}{E} [\sigma_x - \nu (\sigma_y + \sigma_z)] \\ \varepsilon_y &= \frac{1}{E} [\sigma_y - \nu (\sigma_x + \sigma_z)] \\ \gamma_{xy} &= \frac{\tau_{xy}}{G} \end{aligned} \quad (5.5)$$

Na equação 5.5, E , ν e G representam o módulo de Young, coeficiente de Poisson e o módulo de cisalhamento, respetivamente. O módulo de cisalhamento relaciona-se

com as outras duas propriedades do material a partir da seguinte equação:

$$G = \frac{E}{2(1 + \nu)} \quad (5.6)$$

Isolando as tensões normais (σ_x e σ_y) e ainda a tensão de corte (τ_{xy}) e sabendo que a tensão normal segundo z é nula, é possível recorrer às expressões das equações descritas em 5.4 e 5.6 para reescrever as equações em 5.5. Assim o estado de tensão da placa fica definido como:

$$\begin{aligned} \sigma_x &= -\frac{Ez}{1 - \nu^2} \left(\frac{\partial^2 w}{\partial x^2} + \nu \frac{\partial^2 w}{\partial y^2} \right) \\ \sigma_y &= -\frac{Ez}{1 - \nu^2} \left(\frac{\partial^2 w}{\partial y^2} + \nu \frac{\partial^2 w}{\partial x^2} \right) \\ \tau_{xy} &= -\frac{Ez}{1 + \nu} \frac{\partial^2 w}{\partial x \partial y} \end{aligned} \quad (5.7)$$

A partir do estado de tensão é possível integrar cada componente ao longo da espessura da placa para descobrir os momentos fletores segundo o eixo x e segundo o eixo y (M_x e M_y , respetivamente) e ainda o momento torsor no plano xy (M_{xy}). Ficam assim definidos por, [18]:

$$\begin{aligned} M_x &= -D \left(\frac{\partial^2 w}{\partial x^2} + \nu \frac{\partial^2 w}{\partial y^2} \right) \\ M_y &= -D \left(\frac{\partial^2 w}{\partial y^2} + \nu \frac{\partial^2 w}{\partial x^2} \right) \\ M_{xy} &= -D(1 - \nu) \frac{\partial^2 w}{\partial x \partial y} \end{aligned} \quad (5.8)$$

Na equação 5.8, D é o módulo de rigidez à flexão e é definido pela seguinte relação:

$$D = \frac{Et^3}{12(1 - \nu^2)} \quad (5.9)$$

Com os conceitos já explícitos relaciona-se diretamente os momentos com as tensões respetivas, [18]:

$$\sigma_x = \frac{12 M_x z}{t^3} \quad ; \quad \sigma_y = \frac{12 M_y z}{t^3} \quad ; \quad \tau_{xy} = \frac{12 M_{xy} z}{t^3} \quad (5.10)$$

Por fim, obtém-se a tensão equivalente de von Mises através da seguintes relação com as tensões normais e de corte, [6]:

$$\sigma_{vmis} = \sqrt{\frac{(\sigma_x - \sigma_y)^2 + (\sigma_y - \sigma_z)^2 + (\sigma_z - \sigma_x)^2 + 6(\tau_{xy}^2 + \tau_{yz}^2 + \tau_{zx}^2)}{2}} \quad (5.11)$$

5.2.1.1 Equações diferenciais de equilíbrio

Os momentos já anteriormente definidos variam ao longo da placa segundo a direção x e y sendo que esta variação é determinada pelas equações de equilíbrio estático da estrutura. Sendo assim pode-se representar todos os esforços a que a placa está sujeita como se pode observar pela figura 5.9.

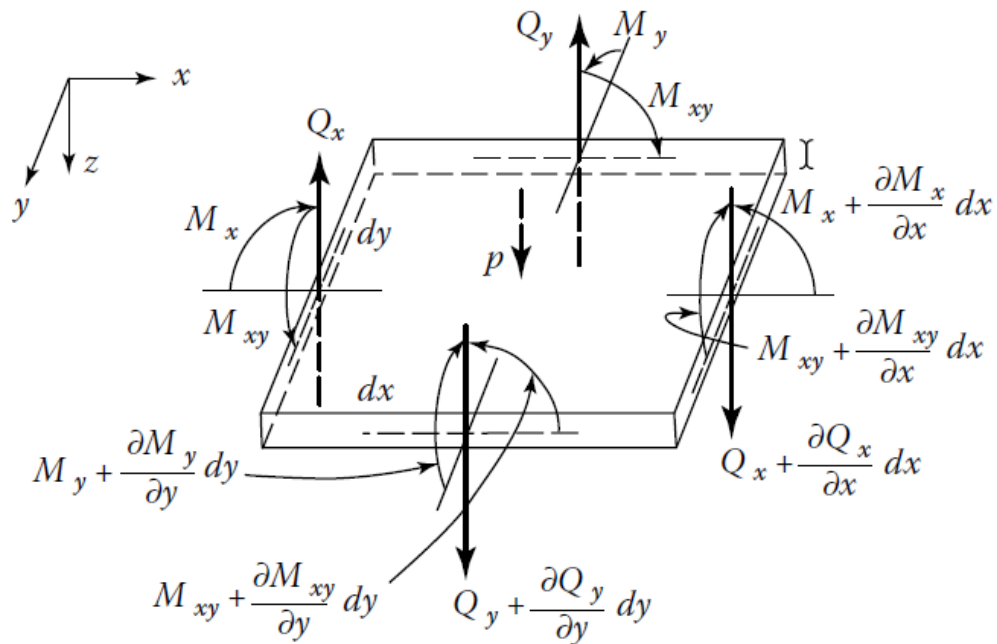


Figura 5.9: Equilíbrio de esforços na placa sujeita a pressão distribuída uniforme, [18].

Os esforços Q_x e Q_y representam os esforços transversos e p a pressão distribuída uniforme a que a placa se encontra sujeita. A partir da representação e dos conceitos

abordados é possível estabelecer as equações de equilíbrio segundo cada direção, [18].

$$\begin{aligned} \frac{\partial M_{xy}}{\partial x} + \frac{\partial M_y}{\partial y} - Q_y &= 0 & (\text{segundo } x) \\ \frac{\partial M_{xy}}{\partial y} + \frac{\partial M_x}{\partial x} - Q_x &= 0 & (\text{segundo } y) \\ \frac{\partial Q_x}{\partial x} + \frac{\partial Q_y}{\partial y} + p &= 0 & (\text{segundo } z) \end{aligned} \quad (5.12)$$

As três equações com um total de cinco incógnitas podem ser reduzidas a uma com três incógnitas através das relações de Q_x e de Q_y com os momentos presentes nos equilíbrios respetivos. Sendo assim a equação que determina o equilíbrio segundo z modifica-se para:

$$\frac{\partial^2 M_x}{\partial x^2} + 2 \frac{\partial^2 M_{xy}}{\partial x \partial y} + \frac{\partial^2 M_y}{\partial y^2} = -p \quad (5.13)$$

No entanto, ainda existe um número maior de incógnitas das que são possíveis resolver, portanto a partir das relações entre os momentos e os deslocamentos normais à placa presentes nas equações em 5.8, obtém-se a equação diferencial de equilíbrio dos deslocamentos:

$$\frac{\partial^4 w}{\partial x^4} + 2 \frac{\partial^4 w}{\partial x^2 \partial y^2} + \frac{\partial^4 w}{\partial y^4} = \frac{p}{D} = \nabla^4 w \quad (5.14)$$

Para obter o valor de w em 5.14 é necessário integrar a equação sendo que as constantes serão definidas pelas condições de fronteira do problema. No seguimento desta secção serão abordadas as condições de fronteira existentes para um painel retangular. Assim consideram-se no âmbito deste análise dois tipos de condições de fronteira e as respetivas restrições:

- Painel simplesmente apoiado em todas as arestas:

$$\begin{aligned} w = 0 \quad \frac{\partial^2 w}{\partial x^2} = 0 & \quad \left(x = -\frac{a}{2}, x = \frac{a}{2} \right) \\ w = 0 \quad \frac{\partial^2 w}{\partial y^2} = 0 & \quad \left(y = -\frac{b}{2}, y = \frac{b}{2} \right) \end{aligned} \quad (5.15)$$

- Painel encastrado em todas as arestas:

$$\begin{aligned} w = 0 \quad \frac{\partial w}{\partial x} = 0 & \quad \left(x = -\frac{a}{2}, x = \frac{a}{2} \right) \\ w = 0 \quad \frac{\partial w}{\partial y} = 0 & \quad \left(y = -\frac{b}{2}, y = \frac{b}{2} \right) \end{aligned} \quad (5.16)$$

A solução exata da equação 5.14 é de alguma complexidade portanto existem na bibliografia algumas soluções aproximadas com diferenças bastante reduzidas. De seguida irão ser apresentadas estas soluções para os dois tipos de condições de fronteira.

5.2.1.2 Solução para um painel simplesmente apoiado

Considerando um painel de dimensões a segundo x e b segundo y , com uma pressão distribuída uniforme ao longo da superfície, será possível estabelecer soluções aproximadas do deslocamento normal segundo z e ainda dos esforços internos para o caso em que se encontra simplesmente apoiada. Esta solução foi apresentada por Urugal baseada nas soluções de Navier e segue séries de Fourier, [18].

Assim o valor do deslocamento w é dado por, [18]:

$$w = \frac{16p}{\pi^6 D} \sum_m^\infty \sum_n^\infty \frac{1}{mn [(m/a)^2 + (n/b)^2]^2} \sin \frac{m\pi(x+a/2)}{a} \sin \frac{n\pi(y+b/2)}{b} \quad (5.17)$$

A partir da solução para o deslocamento pode-se então determinar as soluções para os momentos, [18]:

$$\begin{aligned} M_x &= \frac{16p}{\pi^4} \sum_m^\infty \sum_n^\infty \frac{(m/a)^2 + \nu(n/b)^2}{mn [(m/a)^2 + (n/b)^2]^2} \sin \frac{m\pi(x+a/2)}{a} \sin \frac{n\pi(y+b/2)}{b} \\ M_y &= \frac{16p}{\pi^4} \sum_m^\infty \sum_n^\infty \frac{\nu(m/a)^2 + (n/b)^2}{mn [(m/a)^2 + (n/b)^2]^2} \sin \frac{m\pi(x+a/2)}{a} \sin \frac{n\pi(y+b/2)}{b} \\ M_{xy} &= \frac{16p}{\pi^4 ab} \sum_m^\infty \sum_n^\infty \frac{1-\nu}{mn [(m/a)^2 + (n/b)^2]^2} \cos \frac{m\pi(x+a/2)}{a} \cos \frac{n\pi(y+b/2)}{b} \end{aligned} \quad (5.18)$$

Importante realçar que os termos das séries, m e n , somente terão valores inteiros ímpares e quanto maior o número de termos que se avaliam mais precisos irão ser os resultados. No entanto, como a série é convergente, a partir de um pequeno número de termos o resultado já é considerado válido. Este número será menor nas séries do deslocamento do que nas séries dos momentos, [18].

Sabe-se que tanto o deslocamento w como os momentos fletores M_x e M_y serão máximos no meio da placa isto é em $(x = 0, y = 0)$ e por outro lado o momento torsor M_{xy} será nulo. Como o objetivo deste trabalho é analisar a situação mais crítica será somente do interesse avaliar os componentes nesse ponto. Neste caso, o

produto das funções sinusoidais pode ser simplificado para o seu valor máximo:

$$\sin \frac{m\pi(x + a/2)}{a} \sin \frac{n\pi(y + b/2)}{b} = (-1)^{[(m+n)/2-1]} \quad (x = 0, y = 0) \quad (5.19)$$

Ainda se pode simplificar mais as relações apresentadas a partir do rácio entre os dois lados do painel. Como se considera que a dimensão a será segundo x e b segundo y , define-se assim o coeficiente entre o comprimento dos dois lados como:

$$k = \frac{a}{b} \quad (5.20)$$

Introduzindo este conceitos na equação do deslocamento 5.17 resulta então:

$$w_{max} = \frac{p b^4}{D} \frac{16}{\pi^6} \sum_m^\infty \sum_n^\infty \frac{(-1)^{[(m+n)/2-1]}}{m n [(m/k)^2 + n^2]^2} \quad (5.21)$$

No caso dos momentos descritos em 5.18, no ponto $(x = 0, y = 0)$, reescrevem-se como:

$$M_{x_{max}} = p b^2 \frac{16}{\pi^4} \sum_m^\infty \sum_n^\infty \frac{(m/k)^2 + \nu n^2}{m n [(m/k)^2 + n^2]^2} (-1)^{[(m+n)/2-1]}$$

$$M_{y_{max}} = p b^2 \frac{16}{\pi^4} \sum_m^\infty \sum_n^\infty \frac{\nu (m/k)^2 + n^2}{m n [(m/k)^2 + n^2]^2} (-1)^{[(m+n)/2-1]} \quad (5.22)$$

$$M_{xy} = 0$$

Como se pode observar, as séries necessárias para a solução simplesmente dependem, no caso dos deslocamentos do coeficiente k , e no caso dos momentos, do coeficiente k e do coeficiente de Poisson (ν). Portanto, a partir da utilização de fatores de multiplicação para o deslocamento w , momento segundo x e momento segundo y , (δ_1 , δ_2 e δ_3 , respetivamente), reduz-se bastante as equações apresentadas:

$$w_{max} = \frac{p b^4}{D} \delta_1$$

$$M_{x_{max}} = p b^2 \delta_2 \quad (5.23)$$

$$M_{y_{max}} = p b^2 \delta_3$$

Em que:

$$\begin{aligned}\delta_1(k) &= \frac{16}{\pi^6} \sum_m^{\infty} \sum_n^{\infty} \frac{(-1)^{[(m+n)/2-1]}}{m n [(m/k)^2 + n^2]^2} \\ \delta_2(k, \nu) &= \frac{16}{\pi^4} \sum_m^{\infty} \sum_n^{\infty} \frac{(m/k)^2 + \nu n^2}{m n [(m/k)^2 + n^2]^2} (-1)^{[(m+n)/2-1]} \quad (m, n = 1, 3 \dots) \quad (5.24) \\ \delta_3(k, \nu) &= \frac{16}{\pi^4} \sum_m^{\infty} \sum_n^{\infty} \frac{\nu (m/k)^2 + n^2}{m n [(m/k)^2 + n^2]^2} (-1)^{[(m+n)/2-1]}\end{aligned}$$

Desta forma, foi realizado um algoritmo na ferramenta GNU Octave que permite calcular os fatores de multiplicação através do coeficiente k e do coeficiente de Poisson, e ainda os valores máximos caso sejam fornecidos os dados da pressão, da espessura e módulo de Young. Os intervalos dos valores de m e n para o cálculo também podem ser definidos, no entanto utilizou-se os correspondentes entre 1 e 100. Este número de termos até supera em bastante os necessários, mas visto que o cálculo não resulta num tempo de computação elevado decidiu-se optar por um número que garanta resultados bastante precisos.

Na tabela 5.1, é possível visualizar os fatores de multiplicação para painéis simplesmente apoiados para vários rácios de entrada. Em relação ao coeficiente de Poisson, utilizou-se o valor típico do aço, portanto 0,3.

Tabela 5.1: Fatores de multiplicação para um painel simplesmente apoiado

k	δ_1	δ_2	δ_3
1,0	$4,06 \times 10^{-3}$	$4,79 \times 10^{-2}$	$4,79 \times 10^{-2}$
1,1	$4,87 \times 10^{-3}$	$5,55 \times 10^{-2}$	$4,93 \times 10^{-2}$
1,2	$5,65 \times 10^{-3}$	$6,27 \times 10^{-2}$	$5,00 \times 10^{-2}$
1,3	$6,39 \times 10^{-3}$	$6,94 \times 10^{-2}$	$5,03 \times 10^{-2}$
1,4	$7,08 \times 10^{-3}$	$7,55 \times 10^{-2}$	$5,02 \times 10^{-2}$
1,5	$7,72 \times 10^{-3}$	$8,12 \times 10^{-2}$	$4,98 \times 10^{-2}$
1,6	$8,31 \times 10^{-3}$	$8,62 \times 10^{-2}$	$4,93 \times 10^{-2}$
1,7	$8,84 \times 10^{-3}$	$9,08 \times 10^{-2}$	$4,86 \times 10^{-2}$
1,8	$9,32 \times 10^{-3}$	$9,48 \times 10^{-2}$	$4,79 \times 10^{-2}$
1,9	$9,74 \times 10^{-3}$	$9,85 \times 10^{-2}$	$4,71 \times 10^{-2}$
2,0	$1,01 \times 10^{-2}$	$1,02 \times 10^{-1}$	$4,64 \times 10^{-2}$
∞	$1,30 \times 10^{-2}$	$1,25 \times 10^{-1}$	$3,74 \times 10^{-2}$

A partir dos valores máximos dos momentos, é possível calcular as tensões máximas na direção respetiva utilizando as equações em 5.10 e de seguida a tensão de von

Mises. O caso de uma painel simplesmente apoiado fica assim completamente definido e as relações para os valores máximos teóricos serão utilizadas posteriormente em vários casos de aplicação e comparação com modelos de elementos finitos.

5.2.1.3 Solução para um painel encastrado

No caso de um painel encastrado a solução devido às condições de fronteira apresentadas em 5.16 é de alguma complexidade. No entanto e para efeitos de validação de resultados é possível utilizar os valores indicados pela literatura para vários tipos de geometria. Estes valores assim como apresentados na secção 5.2.1.2 são referentes aos fatores de multiplicação, que a partir das equações em 5.23 permite obter valores tanto de deslocamentos como de momentos, no centro da placa. Para esta solução de igual forma também se admite que a pressão é distribuída uniforme. Assim na tabela 5.2, é possível visualizar os valores dos fatores para um coeficiente de Poisson de 0,3, [17].

Tabela 5.2: Fatores de multiplicação para um painel encastrado (adaptado de [17])

k	δ_1	δ_2	δ_3
1,0	$1,26 \times 10^{-3}$	$2,31 \times 10^{-2}$	$2,31 \times 10^{-2}$
1,1	$1,50 \times 10^{-3}$	$2,64 \times 10^{-2}$	$2,31 \times 10^{-2}$
1,2	$1,72 \times 10^{-3}$	$2,99 \times 10^{-2}$	$2,28 \times 10^{-2}$
1,3	$1,91 \times 10^{-3}$	$3,27 \times 10^{-2}$	$2,22 \times 10^{-2}$
1,4	$2,07 \times 10^{-3}$	$3,49 \times 10^{-2}$	$2,12 \times 10^{-2}$
1,5	$2,20 \times 10^{-3}$	$3,68 \times 10^{-2}$	$2,03 \times 10^{-2}$
1,6	$2,30 \times 10^{-3}$	$3,81 \times 10^{-2}$	$1,93 \times 10^{-2}$
1,7	$2,38 \times 10^{-3}$	$3,92 \times 10^{-2}$	$1,82 \times 10^{-2}$
1,8	$2,45 \times 10^{-3}$	$4,01 \times 10^{-2}$	$1,74 \times 10^{-2}$
1,9	$2,49 \times 10^{-3}$	$4,07 \times 10^{-2}$	$1,65 \times 10^{-2}$
2,0	$2,54 \times 10^{-3}$	$4,12 \times 10^{-2}$	$1,58 \times 10^{-2}$
∞	$2,60 \times 10^{-3}$	$4,17 \times 10^{-2}$	$1,25 \times 10^{-2}$

5.2.2 Aplicação

De forma a comparar e a analisar painéis planos simples foram implementados vários exemplos a partir da rotina criada utilizando o *software* Salome-Meca. Os dados iniciais para o caso do painel plano simples encontram-se presentes na tabela 5.3.

Tabela 5.3: Propriedades de um exemplo de um painel plano simples

Geometria		Material	
a [mm]	1000	E [GPa]	210
b [mm]	500	ν	0,3
t [mm]	10	Solicitação	
k	2	p [MPa]	-0,1

Os valores teóricos utilizando os conceitos abordados em 5.2.1 para este exemplo para o caso de simplesmente apoiado e para o caso de encastramento no centro da placa, encontram-se presentes na tabela 5.4.

Tabela 5.4: Valores teóricos do exemplo de um painel plano simples

	Simplemente apoiado	Encastrado
w [mm]	-3,292	-0,826
σ_{vmis} [MPa]	132,260	54,003

A partir da rotina criada foi analisada a sensibilidade dos resultados à malha utilizada para o caso de simplesmente apoiado e de seguida, variou-se a geometria para verificar a influência nos resultados para o caso de simplesmente apoiado e encastrado. Todas estas análises são comparadas com os valores teóricos obtidos a partir das equações formuladas.

5.2.3 Análise da sensibilidade da malha de elementos finitos

O tamanho do elemento utilizado tem influência no cálculo elementos finitos, portanto para obter resultados fiáveis é necessário analisar a convergência, isto é, estabelecer um tamanho de elemento a partir do qual a variação dos resultados não é significativa. No entanto, um tamanho do elemento bastante reduzido resulta numa análise demorada sendo considerado uma desvantagem. Na tabela 5.5 encontram-se os resultados obtidos para diferentes tamanhos de elemento para o caso de um painel simplesmente apoiado e também é calculada a diferença relativa ao resultado teórico apresentado na tabela 5.4.

Como é possível observar, em relação aos deslocamentos normais ao painel, a malha praticamente não influencia a diferença relativa ao resultado teórico a partir de 80 mm de tamanho do lado de cada elemento. Em relação à tensão von Mises, os resultados também convergem facilmente e indica-se como um tamanho de lado do elemento de 40 mm suficiente para ter resultados conclusivos. Portanto, para efeitos de análises posteriores para esta geometria irá-se utilizar o tamanho de 40 mm.

Tabela 5.5: Resultados obtidos na análise da sensibilidade da malha

Tamanho do elemento [mm]	Nº nós (QUAD9)	w [mm]	dif (%)	σ_{vmis} [MPa]	dif (%)	Tempo [s]
160	153	-3,338	1,401	140,294	6,075	1,92
80	561	-3,322	0,917	134,697	1,842	2,41
40	1881	-3,322	0,917	133,527	0,958	3,39
20	5985	-3,322	0,917	133,254	0,751	6,69
10	21105	-3,322	0,917	133,162	0,682	13,77
5	80601	-3,322	0,917	133,136	0,662	67,53

Na figura 5.10 é possível observar uma representação desta malha obtida a partir do *software*.

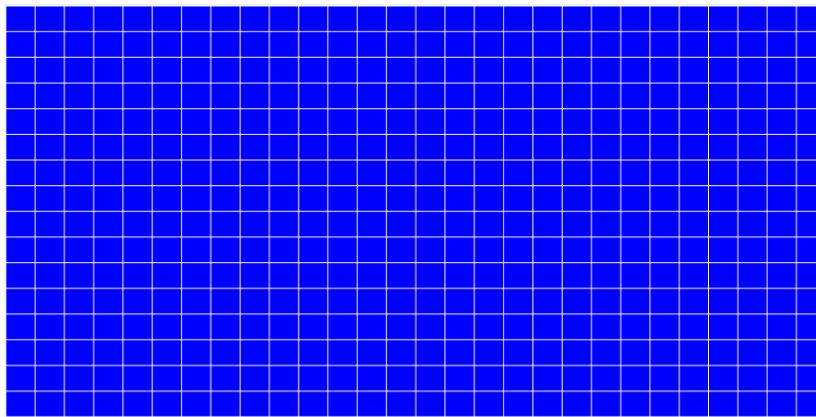


Figura 5.10: Malha de elementos finitos de um painel plano simples com tamanho de elemento igual a 40 mm.

5.2.4 Análise da influência do parâmetro geométrico k

Nesta secção irão ser realizadas análises à variação do parâmetro k , que relaciona o comprimento horizontal com o comprimento vertical, como explicito na equação 5.20. Esta análise irá ser apresentada para o caso de um painel simplesmente apoiado e para o caso de um painel encastrado. Para ambas as situações irão ser calculadas as diferenças relativas ao resultado teórico obtido através dos fatores de multiplicação respetivos a cada k presentes nas tabelas 5.1 e 5.2.

Para o cálculo do resultado teórico utiliza-se as equações estabelecidas em 5.23 em que somente se irá variar o valor de k , para isso mantém-se o valor de a e calcula-se o novo valor de b . O valor do módulo de rigidez à flexão, D , depende do módulo de Young, coeficiente de Poisson e ainda da espessura t que se encontram todos

definidos na tabela 5.3, sendo que o deslocamento é então calculado através do fator de multiplicação respetivo (δ_1). Para o cálculo da tensão equivalente de von Mises, é necessário calcular os momentos fletores a partir de δ_2 e δ_3 e de seguida as tensões normais contabilizando a espessura e com z igual à espessura sobre dois visto que se calcula na camada superior. A partir da equação 5.11 calcula-se então a tensão de von Mises para cada k .

Na tabela 5.6 encontra-se os resultados obtidos para cada rácio entre o comprimento horizontal e vertical para o caso de simplesmente apoiado sendo que os valores obtidos pela análise utilizando a teoria das placas são identificados com uma barra (\bar{w} e $\overline{\sigma_{vmis}}$). Nos cálculos por elementos finitos utiliza-se o tamanho de elemento já indicado na secção anterior de 40 mm.

Tabela 5.6: Resultados obtidos para diferentes rácios geométricos de um painel simplesmente apoiado

k	w [mm]	\bar{w} [mm]	$diff$ (%)	σ_{vmis} [MPa]	$\overline{\sigma_{vmis}}$ [MPa]	$diff$ (%)
1,0	-21,316	-21,124	0,910	289,557	287,320	0,779
1,1	-17,453	-17,289	0,949	263,334	261,160	0,832
1,2	-14,294	-14,156	0,976	241,229	239,180	0,857
1,3	-11,744	-11,659	0,988	222,389	220,380	0,912
1,4	-9,675	-9,580	0,990	205,670	203,790	0,922
1,5	-8,023	-7,945	0,986	190,905	186,160	0,922
1,6	-6,657	-6,592	0,977	177,384	175,620	1,004
1,7	-5,549	-5,496	0,963	164,908	163,280	0,997
1,8	-4,670	-4,626	0,948	153,773	152,270	0,987
1,9	-3,917	-3,881	0,930	143,043	141,660	0,977
2,0	-3,322	-3,292	0,917	133,136	132,260	0,662

Estes resultados indicam que para diferentes geometrias do painel, a análise por elementos finitos e o cálculo teórico aproximam-se com diferenças relativas menores que 1% tanto para os deslocamentos como para a tensão de von Mises no centro da placa.

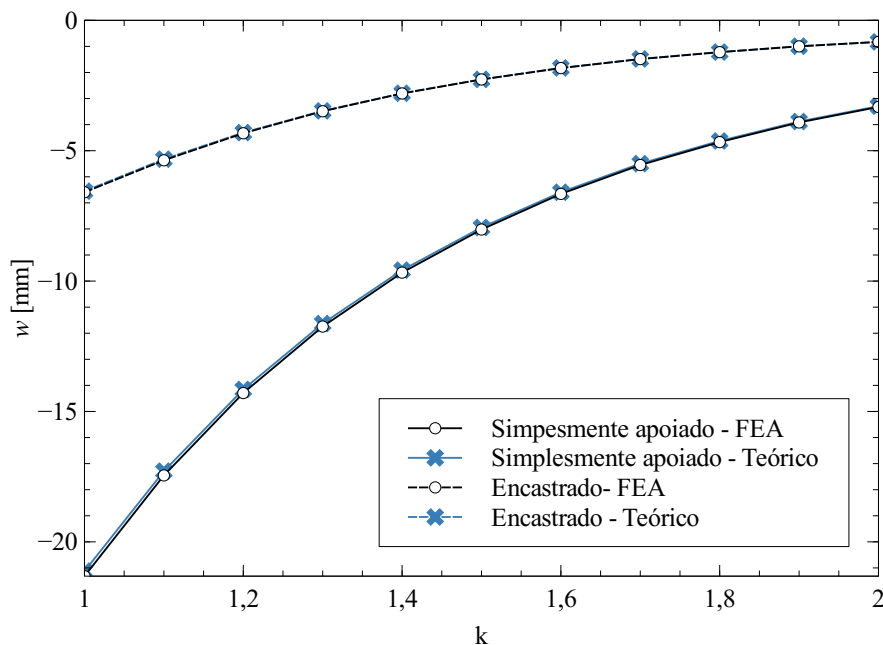
Na tabela 5.7 é possível visualizar os resultados obtidos para o mesmo exemplo no entanto, no caso de encastramento.

No caso de encastramento as diferenças relativas aos valores teóricos calculados também são relativamente baixas portanto pode-se concluir que os resultados da análise de elementos finitos para ambos os casos encontra-se em concordância com a literatura e a rotina permite prever o comportamento de painéis simples com diferentes geometrias e diferentes condições de fronteira.

Tabela 5.7: Resultados obtidos para diferentes ratios geometricos de um painel encastrado

k	w [mm]	\bar{w} [mm]	dif (%)	σ_{vmis} [MPa]	$\bar{\sigma}_{vmis}$ [MPa]	dif (%)
1,0	-6,593	-6,552	0,626	137,851	138,600	0,540
1,1	-5,367	-5,325	0,789	124,935	123,518	1,147
1,2	-4,332	-4,306	0,604	113,367	112,651	0,636
1,3	-3,487	-3,473	0,403	103,181	102,602	0,564
1,4	-2,806	-2,797	0,286	93,871	93,158	0,765
1,5	-2,268	-2,264	0,177	85,575	85,222	0,414
1,6	-1,832	-1,825	0,384	78,107	77,336	0,997
1,7	-1,487	-1,479	0,541	71,180	70,484	0,987
1,8	-1,221	-1,217	0,329	65,097	64,601	0,768
1,9	-0,999	-0,991	0,807	59,353	58,860	0,838
2,0	-0,828	-0,826	0,242	54,376	54,002	0,693

Na figura 5.11 e possivel observar graficamente o deslocamento obtido tanto teoricamente como por elementos finitos em funcao do valor de k para ambas as situacoes de condicoes de fronteira e na figura 5.12 as tensoes de von Mises tambem em funcao do valor de k para os mesmo casos.

Figura 5.11: Distribuicao dos deslocamentos em funcao de k .

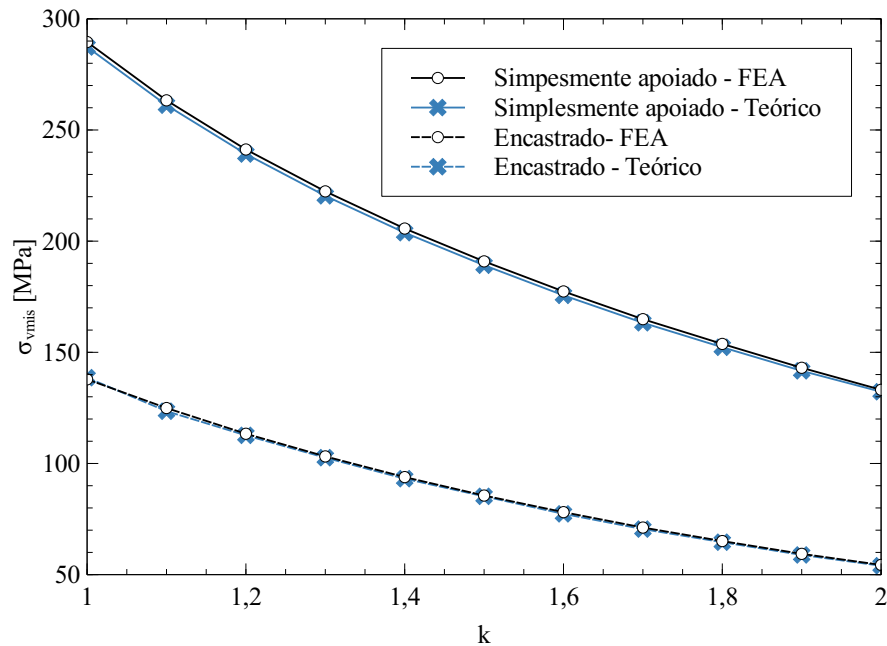


Figura 5.12: Distribuição da tensão de von Mises em função de k .

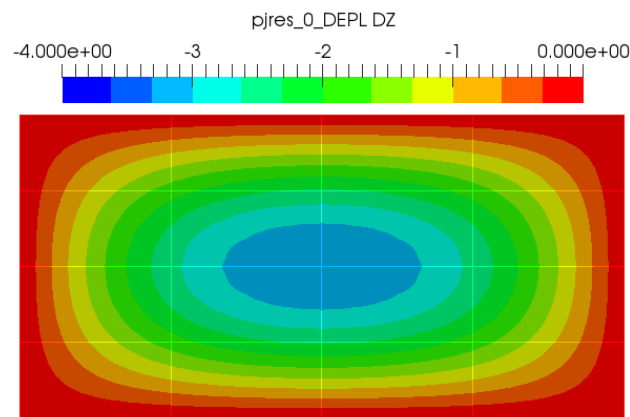
Como se pode observar pelas figuras, todos os resultados obtidos por elementos finitos aproximam-se bastante dos resultados calculados teoricamente. Portanto, a rotina de elementos finitos utilizando o *software* Salome-Meca permite calcular com sucesso os modelos de painéis planos simples.

5.2.5 Comparação entre caso simplesmente apoiado e caso encastrado

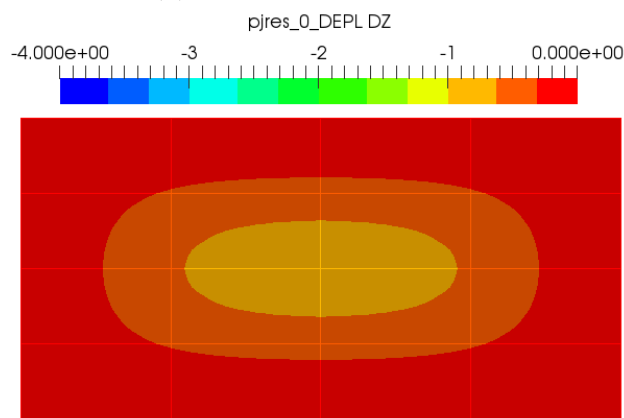
Como se pode observar pelos resultados obtidos, existem grandes diferenças caso o painel se encontre simplesmente apoiado ou encastrado, sendo que no caso de encastramento como se implementa maiores restrições aos deslocamentos e rotações irá apresentar tensões e deslocamentos menores.

Em relação aos deslocamentos segundo z , como se pode observar na figura 5.13, em ambos os casos o valor máximo irá ser no centro do painel, sendo que no caso de encastramento este valor será significativamente menor.

A distribuição da tensão equivalente de von Mises entre os dois casos difere bastante tanto em valor como em localização como se pode observar pela figura 5.14.



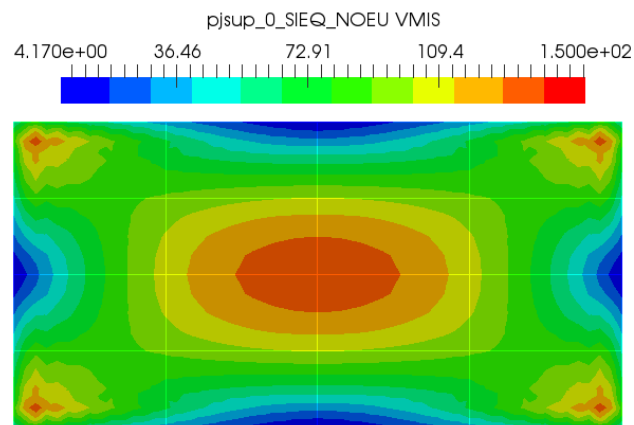
(a) Simplemente apoiado.



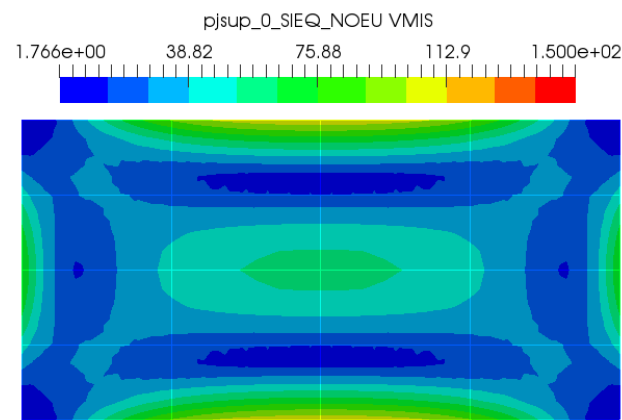
(b) Encastrado

Figura 5.13: Representação do deslocamento segundo z de um painel plano simples.

No caso do painel simplesmente apoiado a tensão von Mises tem maior influência no centro do painel, onde só existem momentos fletores. Na vizinhança dos vértices também existem maiores momentos torsores o que resulta numa influência significativa deste valores nesta localização. Em relação ao painel encastrado, observa-se que existem tensões equivalentes significativas nas fronteiras do painel devido às restrições implementadas que traduzem uma ligação ao exterior mais rígida e que resulta num aparecimento de tensões normais nesta localização. O valor de tensão será maior na aresta mais comprida, e portanto, é possível concluir que a inclusão do maior número de reforços terá de ser perpendicular a esta direção, logo na direção vertical, de forma a diminuir estas tensões. Para efeitos de comparação com o caso simplesmente apoiado utilizou-se o valor no centro do painel no entanto, o valor máximo não será nesta localização contudo, o valor máximo da tensão equivalente será na mesma menor que o valor máximo no caso simplesmente apoiado.



(a) Simplesmente apoiado.



(b) Encastrado

Figura 5.14: Representação da tensão equivalente de von Mises de um painel plano simples.

De forma a reduzir as espessuras utilizadas, mesmo para painéis com dimensões maiores, é necessário incluir reforços que permitem uma redução tanto dos deslocamentos como das tensões.

5.3 Painéis planos reforçados

Em painéis cujo as dimensões assim o exigem, é necessário incluir reforços. Estes componentes viabilizam a utilização de espessuras de painéis menores que equivale a uma menor quantidade de material utilizado. Na figura 5.15 encontra-se uma representação geométrica de um painel plano com reforços verticais e reforço horizontal.

Nesta figura, t_{rv} é a espessura dos reforços verticais, t_{rh} é a espessura do reforço horizontal, h_{rv} é a altura dos reforços verticais, h_{rh} é a altura do reforço horizontal e pos_{rv_i} é a posição de cada reforço vertical em relação ao painel.

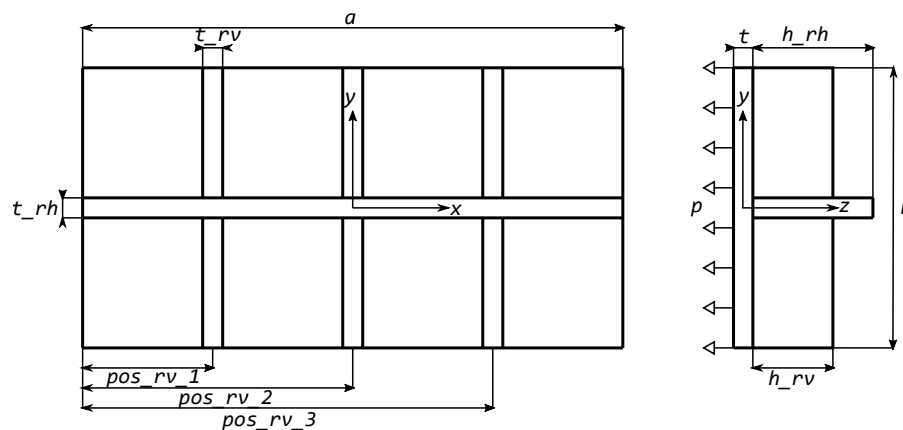


Figura 5.15: Representação geométrica de um painel plano reforçado.

Ainda se define mais uma variável, sendo ela, o número de reforços verticais existentes no painel, n_{rv} . Caso o painel tenha reforços equidistantes:

$$pos_{rv_i} = \frac{a}{n_{rv} + 1} i \quad (i = 1, \dots, n_{rv}) \quad (5.25)$$

Em relação às variáveis relativas aos reforços assim como à variável relativa à espessura do painel é possível introduzir na rotina vários valores de forma a que seja realizado o pré-processamento e o cálculo de vários modelos, cada um com as combinações possíveis destes parâmetros. Assim, esta rotina permite realizar várias análises paramétricas com a finalidade de estudar a influência destas variáveis nos resultados assim como na quantidade de material utilizado.

Como o cálculo analítico que permite obter as tensões e deformações de uma geometria de um painel com reforços é bastante complexo, as comparações dos resultados obtidos serão realizadas a partir de outro *software* de elementos finitos, nomeadamente o ANSYS. Estas validações permitem concluir se os cálculos utilizando o Salome-Meca se aproximam dos cálculos deste *software* reconhecido.

Desta forma, serão realizadas análises da sensibilidade dos resultados ao tamanho de elemento utilizado, análises paramétricas que estudam a influência das variáveis geométricas, estudo e comparação do caso simplesmente apoiado nas arestas e encastrado e ainda uma otimização destes modelos utilizando a rotina implementada.

5.3.1 Aplicação

Para o estudo inicial atribui-se os valores às variáveis para iniciar a rotina que se encontram apresentados na tabela 5.8. Estes valores foram ponderados de forma a sobredimensionar o painel.

Tabela 5.8: Propriedades de um exemplo de um painel plano reforçado

Geometria					
a [mm]	4000	n_{rv}	7	h_{rh} [mm]	200
b [mm]	2000	h_{rv} [mm]	200	t_{rh} [mm]	10
t [mm]	10	t_{rv} [mm]	10	Solicitação	
Material				p [MPa]	-0,1
E [GPa]	210	ρ [kg/m ³]	7850	Massa	
ν	0,3	σ_Y [MPa]	355	m [kg]	910,6

A partir dos dados da geometria a rotina permite criar o modelo geométrico através do algoritmo de pré-processamento. A representação gráfica no módulo de geometria do Salome-Meca encontra-se presente na figura 5.16

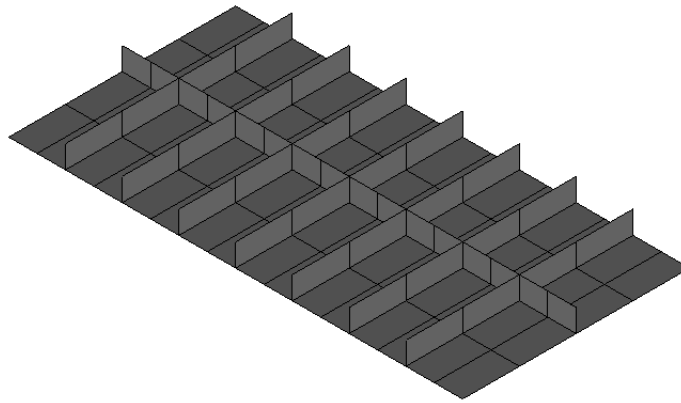


Figura 5.16: Representação gráfica da geometria de um painel plano reforçado no Salome-Meca.

A rotina de seguida cria a malha de elementos finitos, realiza o cálculo e por fim, realiza o pós-processamento. Este pós-processamento como já indicado permite visualizar no módulo ParaView o campo de tensões e campo de deslocamentos assim como produz um relatório final com as informações de todos os modelos analisados.

5.3.2 Análise da sensibilidade da malha de elementos finitos

A partir do modelo geométrico de um painel plano reforçado definido na secção 5.3.1 realizou-se uma análise à malha de elementos finitos para determinar a influência do tamanho de elemento nos resultados fornecidos. Para esta análise optou-se por aplicar a situação de um painel simplesmente apoiado nas arestas.

O refinamento da malha para além de permitir uma convergência de resultados, permite detetar picos de tensão que ocorram devido a métodos numéricos e que não

representam a realidade podendo ser ignorados tendo em conta o local onde ocorram. Estes picos aparecem devido a uma carga concentrada numa área pequena, portanto, à medida que o tamanho de elemento diminui, o efeito aumenta de intensidade a tender para um valor infinito e torna-se cada vez mais local. No caso de um painel reforçado estes valores ocorrem principalmente nos nós de ligação entre os reforços e as arestas do painel, o que pode ser explicado por esses locais se encontrarem extremamente restringidos o que resulta num pico do campo de tensões. Na análise da convergência de resultados será admitido o valor do deslocamento máximo e ainda a tensão de von Mises máxima na aresta que define a ligação entre o reforço mais distante do centro e o painel. Na tabela 5.9 visualiza-se os resultados obtidos para cada tamanho de elemento, o número de nós, diferença entre resultados e o tempo de análise. A diferença entre resultados é calculada relativamente ao resultado obtido com o tamanho de elemento utilizado anteriormente.

Tabela 5.9: Resultados obtidos na análise da sensibilidade da malha de um painel plano reforçado

Tamanho do elemento [mm]	Nº nós	w [mm]	dif (%)	σ_{vmis} [MPa]	dif (%)	Tempo [s]
160	3301	-2,882	-	92,308	-	1,7
80	10479	-2,903	0,729	95,405	3,355	4,3
40	33155	-2,915	0,413	96,338	0,978	16,2
20	120157	-2,926	0,377	96,691	0,366	83,1
10	465241	-2,933	0,239	96,894	0,210	528,4
8	745275	-2,934	0,034	96,941	0,049	1246,6

Como se pode observar, com um tamanho de elemento de 40 mm já se obtém somente uma diferença menor que 1% relativamente ao resultado anterior. Por isso, de forma a não obter um tempo de análise excessivo visto que o objetivo desta rotina é calcular o maior número de modelos possível, considera-se que o tamanho de elemento de 40 mm é suficiente para obter resultados satisfatórios. Na figura 5.17 encontra-se uma representação da malha com tamanho de elemento igual a 40 mm, obtida no módulo do Salome-Meca.

5.3.3 Análise da influência dos parâmetros geométricos

No âmbito da rotina implementada é possível introduzir vários valores em forma de lista das variáveis relativas aos reforços e à espessura do painel, portanto, é do interesse deste trabalho estudar a influência de cada variável nos resultados. Para cada análise foi calculado a tensão de von Mises máxima e os deslocamentos

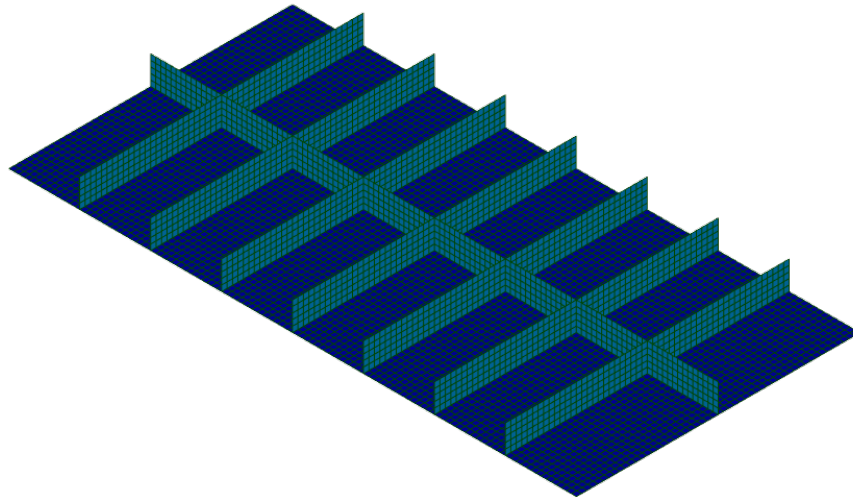


Figura 5.17: Malha de elementos finitos de um painel plano reforçado com tamanho de elemento igual a 40 mm.

e identificado os locais onde ocorrem. Os locais são definidos segundo a seguinte terminologia:

- $painel_j$ - corresponde a cada painel entre-reforços em que j é a localização de cada quando numerados da esquerda para a direita;
- rv_sup_i - corresponde à aresta superior de cada reforço vertical em que i é a localização de cada quando numerados da esquerda para a direita;
- rv_inf_i - corresponde à aresta inferior de cada reforço vertical em que i é a localização de cada quando numerados da esquerda para a direita;
- rh_sup - corresponde à aresta superior do reforço horizontal;
- rh_inf - corresponde à aresta inferior do reforço horizontal;
- $aresta_vertical$ - corresponde à aresta vertical do painel com ligação ao exterior;
- $aresta_horizontal$ - corresponde à aresta horizontal do painel com ligação ao exterior.

De salientar que os modelos são simétricos relativamente à geometria, condições de fronteira e solitação, tanto segundo x como segundo y , portanto os valores das tensões e deslocamentos assim como os locais onde ocorrem também o serão.

Desta forma, para cada modelo foi calculado a massa, que será o fator comum entre todas as análises, visto que o objetivo do dimensionamento destes painéis será obter um comportamento mecânico no regime elástico com o modelo que apresenta a menor massa. Para todas as análises subsequentes foi admitido o material, solicitações e dimensões gerais do painel, (a e b), presentes na tabela 5.8 assim como, a situação de simplesmente apoiado.

5.3.3.1 Espessura do painel

Nesta análise foi admitido um painel plano simples, com as dimensões gerais já referidas, em que se variou a espessura do painel até se obter resultados no regime elástico. Na tabela 5.10 é possível visualizar estes resultados.

Tabela 5.10: Análise da influência da espessura de um painel plano reforçado

t [mm]	w [mm]	Local	σ_{vmis} [MPa]	Local	m [kg]
8	-1648,572	<i>painel_0</i>	3311,785	<i>painel_0</i>	502,4
10	-844,424	<i>painel_0</i>	2120,234	<i>painel_0</i>	628,0
12	-488,879	<i>painel_0</i>	1472,865	<i>painel_0</i>	753,6
14	-307,999	<i>painel_0</i>	1082,457	<i>painel_0</i>	879,2
16	-206,426	<i>painel_0</i>	829,027	<i>painel_0</i>	1004,8
18	-145,044	<i>painel_0</i>	655,247	<i>painel_0</i>	1130,4
20	-105,785	<i>painel_0</i>	530,923	<i>painel_0</i>	1256,0
22	-79,514	<i>painel_0</i>	438,922	<i>painel_0</i>	1381,6
24	-61,274	<i>painel_0</i>	368,936	<i>painel_0</i>	1507,2
26	-48,217	<i>painel_0</i>	314,462	<i>painel_0</i>	1632,8
28	-38,623	<i>painel_0</i>	271,232	<i>painel_0</i>	1758,4
30	-31,417	<i>painel_0</i>	236,350	<i>painel_0</i>	1884,0
32	-25,900	<i>painel_0</i>	207,797	<i>painel_0</i>	2009,6

Como se pode observar para se obter resultados dentro do regime elástico, em relação somente à tensão de cedência do material, a espessura do painel terá de ser bastante elevada. Uma espessura elevada tem como consequência uma massa significativa o que não é do interesse do dimensionamento deste componentes. Portanto, como já referido, é necessário incluir reforços de forma a evitar este fenómeno. Tanto a tensão equivalente como o deslocamento máximo ocorrem no centro do painel, que é referido como *painel_0*, visto que é uma situação de um painel simples simplesmente apoiado.

5.3.3.2 Número de reforços verticais

O número de reforços verticais permite fornecer maior rigidez ao painel em várias localizações. Para esta análise admite-se que os reforços são equidistantes e com a altura e espessura de reforços verticais, assim como a espessura do painel, referidas na tabela 5.8. Na tabela 5.11 encontram-se os resultados para cada número de reforços, mantendo a situação de um painel simplesmente apoiado.

Tabela 5.11: Análise da influência do número de reforços verticais de um painel plano reforçado

n_{rv}	w [mm]	Local	σ_{vmis} [MPa]	Local	m [kg]
1	-242,236	<i>painel_1</i>	1670,081	<i>rv_inf_1</i>	659,4
2	-82,111	<i>painel_1</i>	834,787	<i>rv_inf_1</i>	690,8
3	-32,079	<i>painel_1</i>	497,652	<i>rv_inf_1</i>	722,2
4	-15,240	<i>painel_1</i>	351,155	<i>rv_sup_1</i>	753,6
5	-8,373	<i>painel_1</i>	290,805	<i>rv_sup_1</i>	785,0
6	-5,212	<i>painel_1</i>	247,161	<i>rv_sup_1</i>	816,4
7	-3,588	<i>painel_1</i>	214,244	<i>rv_sup_1</i>	847,8
8	-2,677	<i>painel_1</i>	189,276	<i>rv_sup_1</i>	879,2
9	-2,221	<i>painel_2</i>	168,934	<i>rv_sup_1</i>	910,6
10	-1,970	<i>painel_2</i>	152,027	<i>rv_sup_1</i>	942,0

A partir desta análise foi possível observar que tanto o deslocamento máximo como a tensão equivalente apresentam resultados no regime elástico a partir de número relativamente baixo de reforços, o que significa um menor acréscimo à massa total do painel. Em relação ao deslocamento, o local onde ocorre é praticamente sempre no painel mais distante do centro, no entanto, a partir de um número de reforços o deslocamento será no segundo painel. Isto acontece visto que a partir desse número de reforços, estes têm cada vez menos influência nos resultados portanto ele irá ocorrer em posições cada vez mais perto das anteriores.

A tensão de von Mises máxima será na ligação do reforço com o painel quando o número de reforços verticais é menor que 4. Esta localização deve-se ao facto desta aresta se encontrar bastante restringida e portanto sobrepõe-se os efeitos das tensões nesta zona. A partir de um certo número de reforços a tensão já ocorre na aresta superior do reforço mais distante visto que as tensões no painel irão reduzir com este aumento, portanto a tensão será mais evidente neste local.

5.3.3.3 Altura dos reforços verticais

A altura dos reforços verticais permite criar rigidez sem alterar o número de reforços. Nesta análise foi utilizado o resultado mais favorável da análise anterior, portanto cinco reforços verticais. A espessura do reforço, espessura do painel e dimensões gerais mantêm-se com os valores anteriores.

Tabela 5.12: Análise da influência da altura dos reforços verticais de um painel plano reforçado.

h_{rv}	w [mm]	Local	σ_{vmis} [MPa]	Local	m [kg]
60	-91,534	<i>painel_3</i>	2639,548	<i>rv_sup_2</i>	675,1
80	-43,192	<i>painel_2</i>	1568,969	<i>rv_sup_2</i>	690,8
100	-24,306	<i>painel_2</i>	1024,225	<i>rv_sup_1</i>	706,5
120	-15,342	<i>painel_2</i>	744,497	<i>rv_sup_1</i>	722,2
140	-11,701	<i>painel_1</i>	562,895	<i>rv_sup_1</i>	737,9
160	-10,021	<i>painel_1</i>	440,348	<i>rv_sup_1</i>	753,6
180	-9,018	<i>painel_1</i>	353,906	<i>rv_sup_1</i>	769,3
200	-8,373	<i>painel_1</i>	290,805	<i>rv_sup_1</i>	785,0
220	-7,952	<i>painel_1</i>	243,534	<i>rv_sup_1</i>	800,7
240	-7,654	<i>painel_1</i>	209,108	<i>rv_inf_1</i>	816,4
260	-7,438	<i>painel_1</i>	210,205	<i>rv_inf_1</i>	832,1
280	-7,275	<i>painel_1</i>	211,029	<i>rv_inf_1</i>	847,8
300	-7,155	<i>painel_1</i>	211,923	<i>rv_inf_1</i>	863,5

A partir da tabela 5.12 percebe-se que um aumento da altura de reforços verticais tem um impacto pouco significativo na massa global do painel (neste exemplo é 0,785 kg/mm) e permite obter reduções com bastante impacto tanto nas tensões como no deslocamento máximo. Quanto ao deslocamento, com apenas 180 mm de altura do reforço horizontal é possível obter uma flecha menor que a espessura utilizada no entanto para a tensão equivalente ser menor que a tensão de cedência é necessário uma altura de 200 mm. Quando a altura dos reforços é menor que estes 200 mm, os deslocamentos são significativos, portanto não cumpre os critérios estabelecidos. Em contrapartida, a partir de uma certa altura superior aos 200 mm este aumento não influencia significativamente a redução dos deslocamentos e ainda pode ocorrer um aumento da tensão equivalente devido à criação de uma rigidez elevada na ligação entre os reforços e o painel.

5.3.3.4 Espessura dos reforços verticais

A espessura do reforço vertical, à semelhança da altura, permite criar rigidez. Deste modo variou-se este atributo em painéis com cinco reforços verticais com 200

mm de altura. Os resultados encontram-se expressos na tabela 5.13.

Tabela 5.13: Análise da influência da espessura dos reforços verticais de um painel plano reforçado

t_{rv}	w [mm]	Local	σ_{vmis} [MPa]	Local	m [kg]
2	-17,196	<i>painel_2</i>	1321,888	<i>rv_sup_1</i>	659,4
4	-11,171	<i>painel_1</i>	691,024	<i>rv_sup_1</i>	690,8
6	-9,616	<i>painel_1</i>	470,815	<i>rv_sup_1</i>	722,2
8	-9,616	<i>painel_1</i>	358,689	<i>rv_sup_1</i>	753,6
10	-8,373	<i>painel_1</i>	290,805	<i>rv_sup_1</i>	785,0
12	-8,051	<i>painel_1</i>	245,302	<i>rv_sup_1</i>	816,4
14	-7,808	<i>painel_1</i>	212,645	<i>rv_sup_1</i>	847,8
16	-7,610	<i>painel_1</i>	212,262	<i>rv_inf_1</i>	879,2
18	-7,445	<i>painel_1</i>	214,990	<i>rv_inf_1</i>	910,6
20	-7,303	<i>painel_1</i>	217,594	<i>rv_inf_1</i>	942,0

Quando é aumentado a espessura dos reforços, o comportamento mecânico é semelhante ao comportamento evidenciado quando se aumenta a altura. No entanto, o aumento da espessura demonstra um maior impacto na massa global do painel.

5.3.3.5 Reforço horizontal

O reforço horizontal encontra-se numa direção que não favorece a diminuição da tensão equivalente e ainda dos deslocamentos. No entanto este é utilizado para obter uma diminuição das dimensões dos reforços verticais que têm mais impacto na massa global do que as dimensões do reforço horizontal. O painel plano estudado com as dimensões referidas somente com reforço horizontal não é uma solução viável.

5.3.3.6 Discussão da análise

Os resultados obtidos são relativos às dimensões utilizadas por isso não será possível extrapolar uma previsão exata do comportamento mecânico de um painel reforçado para qualquer valor das variáveis existentes. No entanto um dos objetivos desta análise é concluir um procedimento base para a otimização deste tipo de modelos. De forma, a comparar todos os resultados obtidos é possível visualizar todas as análises representadas, cada uma pela massa e tensão máxima obtida, na figura 5.18. Esta representação inclui também as análises realizadas ao reforço horizontal que permitiram concluir que apesar do impacto da massa deste componente ser bastante menor, o reforço horizontal sozinho nunca poderá reduzir as tensões significativamente. A representação gráfica só inclui valores até 2000 MPa e entre 650 e 950 kg, visto que valores fora dessa gama já não são representativos.

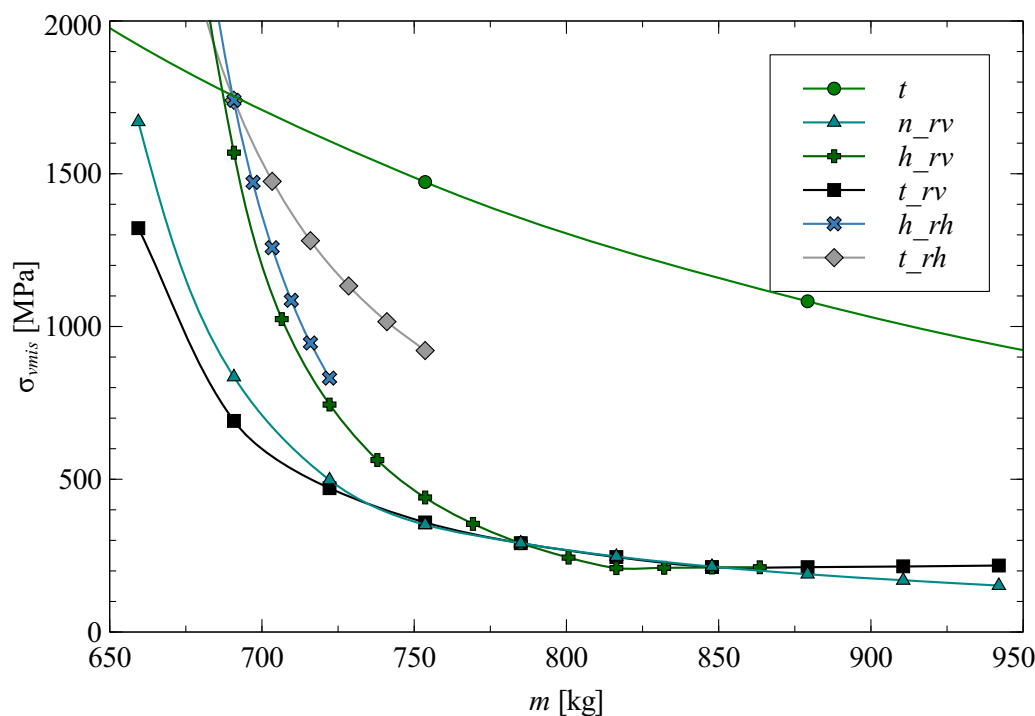


Figura 5.18: Representação da tensão equivalente de von Mises em função da massa para cada análise paramétrica de um painel plano reforçado.

Com vista a obter diretrizes possíveis de retirar das análises realizadas considerou-se os seguintes fundamentos:

- A espessura do painel tem um impacto negativo na massa global sem reduzir significativamente as tensões, portanto será o fator mais importante a reduzir numa otimização;
- O número de reforços verticais permite reduzir as tensões e deslocamentos ao longo do painel sem comprometer a massa global, no entanto será um fator que é necessário ter precaução sendo que será sempre importante começar por definir uma distância mínima entre reforços e uma distância máxima;
- A altura dos reforços verticais foi a variável que demonstrou melhor equilíbrio entre a massa acrescentada e a redução de tensão, portanto o acréscimo deste valor permite obter resultados favoráveis num algoritmo de otimização;
- A espessura dos reforços verticais permite reduzir também significativamente as tensões do painel contudo, das variáveis geométricas dos reforços verticais foi a que demonstrou ter o impacto mais negativo na massa do modelo, portanto, será preferível otimizar a estrutura modificando os outros parâmetros.

- O reforço horizontal permite reduzir na globalidade as dimensões dos parâmetros dos reforços verticais visto que apresenta um impacto menor na massa global do painel. Portanto será necessário estabelecer um equilíbrio entre estes dois componentes.

5.3.4 Otimização de um painel plano reforçado

Após as análises paramétricas realizadas foi do interesse deste trabalho utilizar a rotina implementada, principalmente a funcionalidade de analisar vários modelos simultaneamente. Este cálculo permite obter o modelo, dos analisados, que cumpre o critério da tensão máxima admissível e com menor massa. Os painéis analisados encontram-se todos na situação de simplesmente apoiado.

No âmbito desta análise estabeleceu-se uma variação de vários parâmetros simultaneamente segundo as análises realizadas na secção 5.3.3. Foram analisados 84 modelos diferentes onde se estima uma tempo de análise conjunto de cerca de 30 minutos, incluindo pré-processamento, cálculo e pós-processamento, o que demonstra desde já a viabilidade das rotinas de cálculo automático para realizar várias análises num curto espaço de tempo. Os resultados obtidos para a tensão equivalente de von Mises em função da massa encontram-se expressos na figura 5.19.

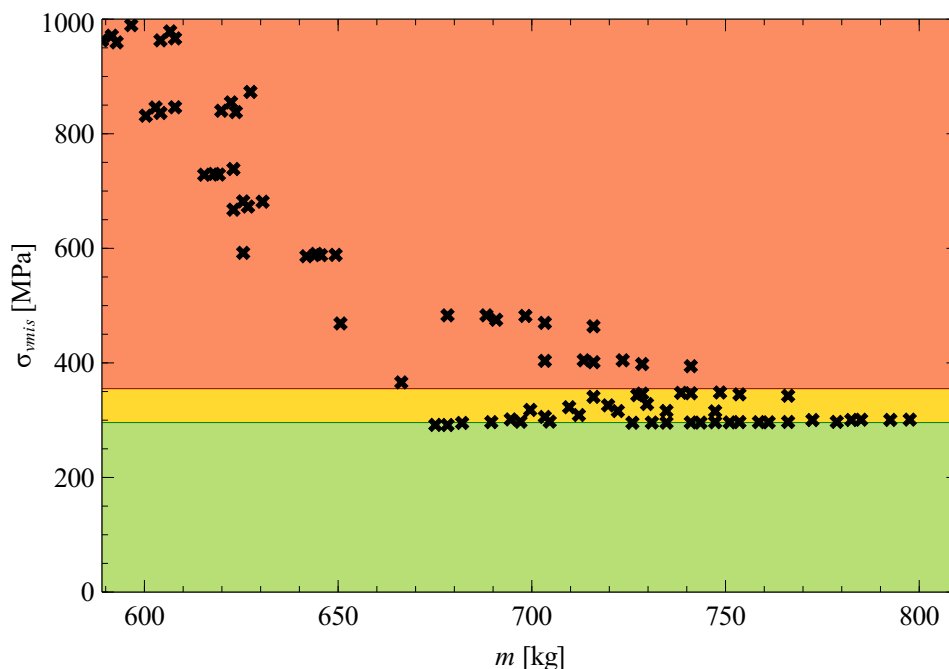


Figura 5.19: Representação da tensão equivalente de von Mises em função da massa para obter um painel plano reforçado otimizado.

No gráfico da figura 5.19, a zona inferior ou verde, corresponde às tensões abaixo da tensão máxima calculada através da equação 5.1 com um coeficiente de segurança de 1,2. A zona intermédia ou amarela corresponde às tensões de von Mises entre a tensão máxima e a tensão de cedência correspondente ao limite máximo da região elástica. A zona superior ou vermelha já serão tensões superiores à tensão de cedência que não respeitam o critério.

De acordo com as análises realizadas foi possível obter o modelo com menor massa com uma tensão de von Mises máxima abaixo da permitida. As propriedades do modelo e resultados encontram-se na tabela 5.14.

Tabela 5.14: Propriedades do modelo obtido após análises paramétricas

t [mm]	n_{rv} [mm]	h_{rv} [mm]	t_{rv} [mm]	h_{rh} [mm]	t_{rh} [mm]
8	5	200	10	100	5
w [mm]	Local	σ_{vmis} [MPa]	Local	m [kg]	
-9,604	<i>painel_1</i>	291,755	<i>rv_sup_1</i>	675.1	

Uma análise de otimização da construção do painel permite assim obter uma diminuição significativa da massa deste componente. Neste exemplo, obteve-se uma flecha maior que a espessura mas justifica-se pelo facto de a análise realizada ter usado como condições de fronteira o caso de simplesmente apoiado, que naturalmente irá permitir deslocamentos maiores enquanto que no caso mais aproximado à realidade, encastrado, estes deslocamentos irão diminuir. Portanto, utilizando o caso de simplesmente apoiado, é possível admitir deslocamentos e tensões maiores. Os campos de resultados do modelo ótimo podem ser visualizados nas figura 5.20 e 5.21.

Com a utilização desta rotina é então possível analisar eficientemente vários modelos por elementos finitos de modo a aproximar-se de uma estrutura inicial de um modelo otimizado.

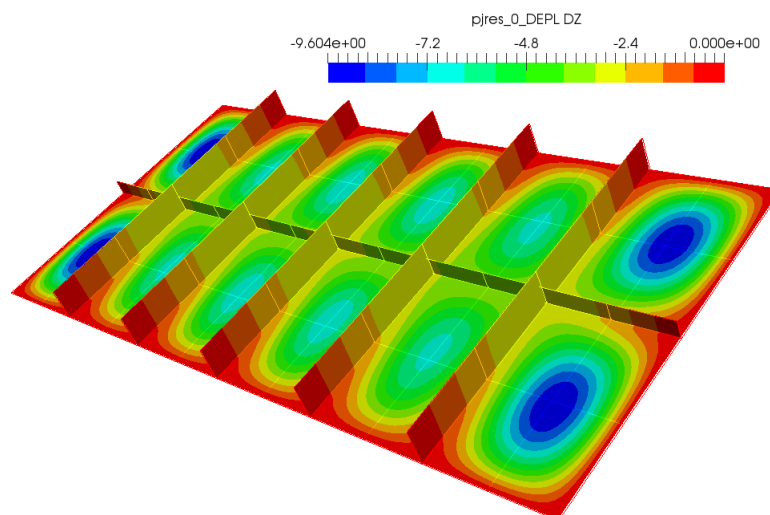


Figura 5.20: Campo de deslocamentos do modelo ótimo caso simplesmente apoiado.

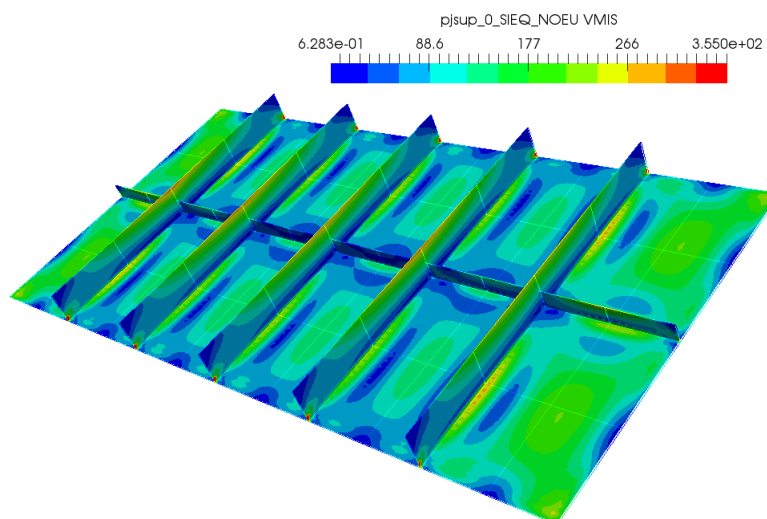


Figura 5.21: Campo das tensões equivalentes de von Mises do modelo ótimo caso simplesmente apoiado.

5.3.5 Comparação entre caso simplesmente apoiado e caso encastrado

O modelo ótimo foi obtido para o caso de simplesmente apoiado portanto será do interesse deste trabalho comparar os resultados com o caso de encastramento. Os valores máximos desta análise encontram-se expressos na tabela 5.15.

Tabela 5.15: Resultados do modelo ótimo para o caso encastrado

w [mm]	Local	σ_{vmis} [MPa]	Local
-6,585	<i>painel_2</i>	295,206	<i>aresta_vertical</i>

Nas figuras 5.22 e 5.23 encontram-se as representações gráficas dos campos de resultados na ferramenta ParaView.

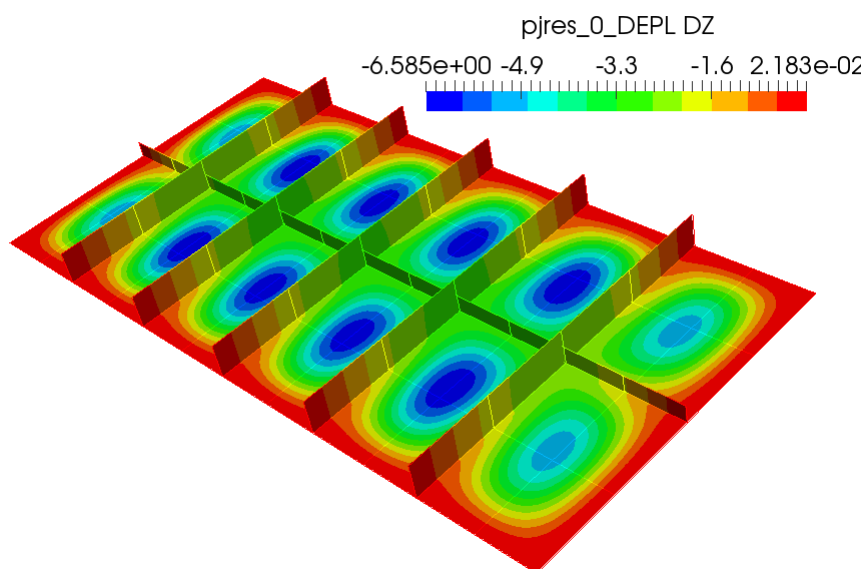


Figura 5.22: Campo de deslocamentos do modelo ótimo caso encastrado.

Em relação aos deslocamentos, o caso simplesmente apoiado apresenta uma maior deformação, sendo que também é possível observar que neste caso os deslocamentos encontram-se na zona mais perto da fronteira. Estes resultados devem-se ao facto de no caso de simplesmente apoiado existirem menos restrições na fronteira o que possibilita maiores deslocamentos. No entanto, no resto do painel, o caso simplesmente apoiado aproxima-se bastante aos deslocamentos obtidos para o caso de encastramento, devido a nestas zonas as condições de fronteira terem menor influência.

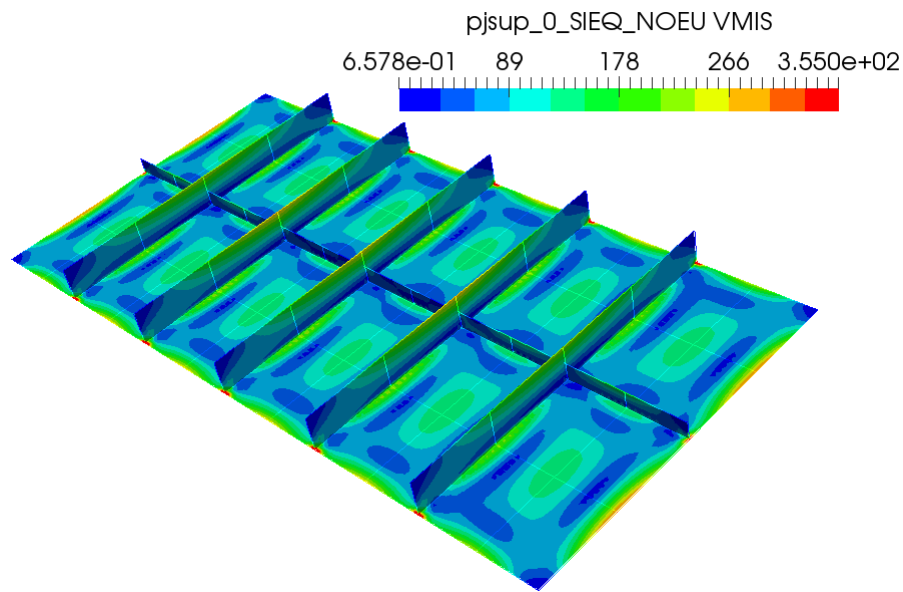


Figura 5.23: Campo das tensões equivalentes de von Mises do modelo ótimo caso encastrado.

Em relação às tensões, na generalidade do painel, a tensão de von Mises é menor no caso de encastrado, com exceção nas fronteiras onde é possível observar tensões elevadas devido às restrições aplicadas nesta zona. Em relação às tensões nos reforços, o caso encastrado apresenta a tensão mais elevada no segundo reforço a contar da esquerda enquanto no caso simplesmente apoiado esta tensão localiza-se no primeiro reforço.

Um painel real de uma cuba de um transformador apresenta uma combinação de resultados entre estes dois casos sendo que este estudo será realizado num capítulo posterior.

5.3.6 Validação de resultados

No âmbito de validar os resultados obtidos para o modelo ótimo, assim como certificar que a rotina implementada apresenta resultados viáveis de serem utilizados procedeu-se a uma validação comparando os campos de resultados obtidos com o *software* de elementos finitos Salome-Meca, com outro *software* de elementos finitos, nomeadamente o ANSYS.

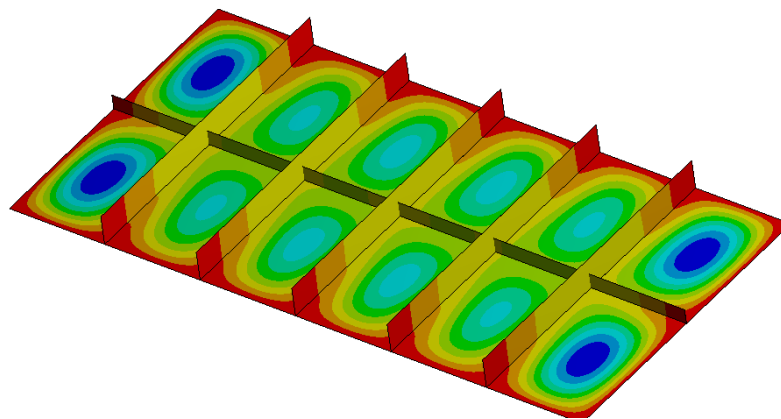
No *software* ANSYS foi implementada a mesma geometria, as mesmas condições de fronteira e a mesma solicitação. Em relação à malha de elementos finitos, foi

utilizado elementos quadriláteros com o mesmo tamanho, no entanto, devido à impossibilidade de utilizar elementos de 9 nós, utilizou-se elementos quadráticos de 8 nós.

Nas figuras 5.24 e 5.25 encontram-se os resultados obtidos para o caso de simplesmente apoiado e encastrado no *software* ANSYS que podem ser respetivamente comparados com os deslocamentos e tensões obtidas para o caso simplesmente apoiado nas figuras 5.20 e 5.21 e para o caso encastrado nas figuras 5.22 e 5.23.

A: Static Structural
Directional Deformation
Type: Directional Deformation(Z Axis)
Unit: mm
Global Coordinate System
Time: 1
19/08/2019 11:47

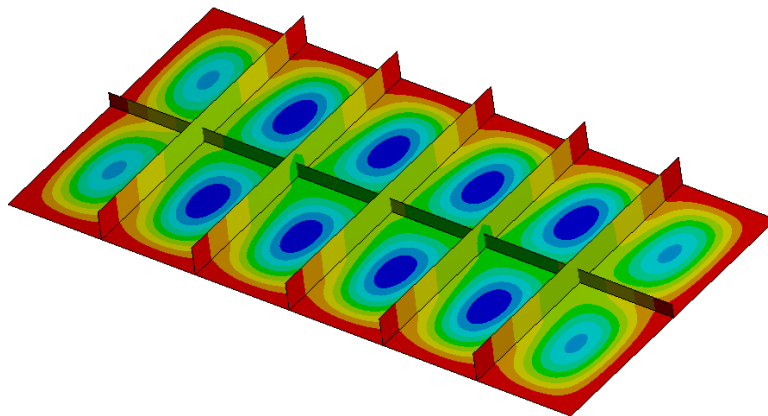
0 Max
-1,0681
-2,1361
-3,2042
-4,2723
-5,3403
-6,4084
-7,4765
-8,5445
-9,6126 Min



(a) Simplesmente apoiado.

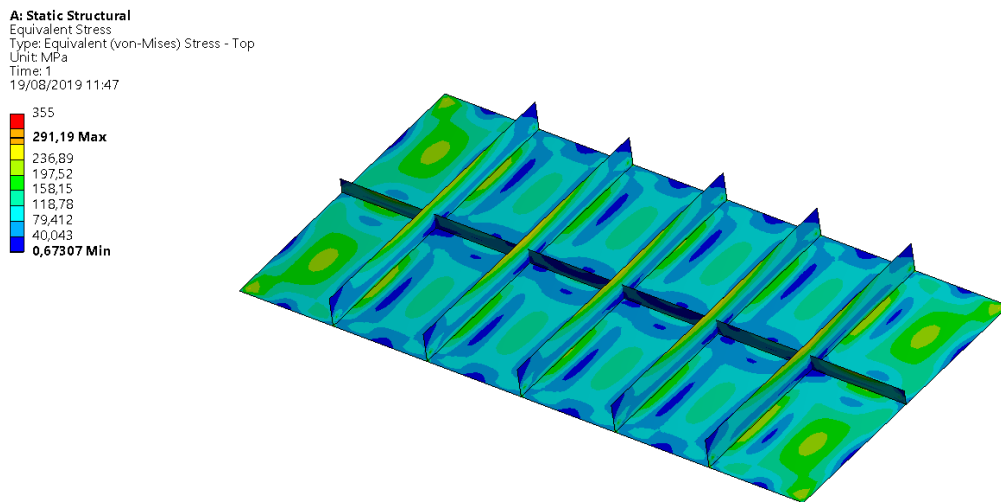
A: Static Structural
Directional Deformation
Type: Directional Deformation(Z Axis)
Unit: mm
Global Coordinate System
Time: 1
19/08/2019 11:45

0,022827 Max
-0,7106
-1,444
-2,1774
-2,9109
-3,6443
-4,3777
-5,1111
-5,8446
-6,578 Min

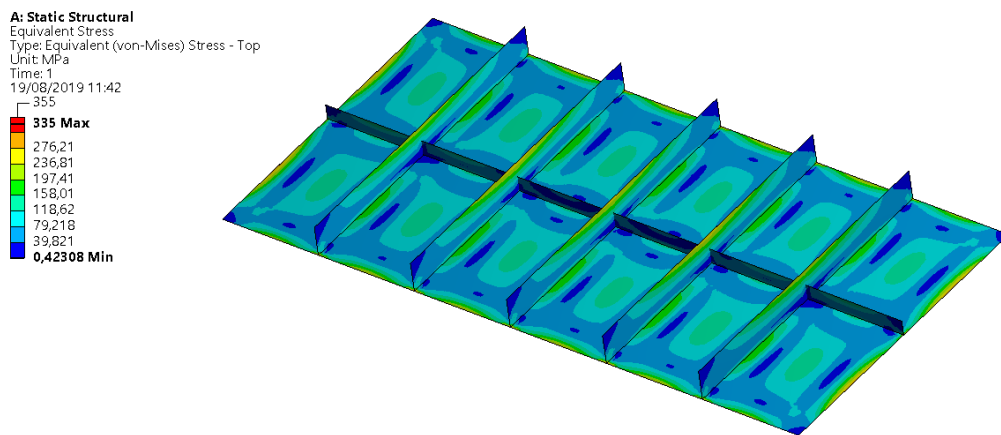


(b) Encastrado

Figura 5.24: Representação do deslocamento segundo z de um painel plano reforçado a partir do ANSYS.



(a) Simplesmente apoiado.



(b) Encastrado

Figura 5.25: Representação da tensão von Mises de um painel plano reforçado a partir do ANSYS.

Como é possível observar, tanto os valores deslocamentos como os valores das tensões von Mises se encontram bastante próximos dos valores obtidos no *software* Salome-Meca. Sendo que no caso dos deslocamentos a diferença do valor máximo obtido utilizando a rotina relativamente ao ANSYS é 0,094% no painel simplesmente apoiado e 0,106% no painel encastrado. Em relação à tensão, encontra-se na figura 5.26 uma representação da distribuição na aresta superior do primeiro reforço, para o caso de simplesmente apoiado e para o caso encastrado, utilizando os resultados obtidos nos dois *softwares*.

A partir do gráfico da figura 5.26 percebe-se que as diferenças relativas entre os dois *softwares* para as tensões são praticamente nulas à semelhança das diferenças obtidas para os deslocamentos. No entanto a partir da representação gráfica nas

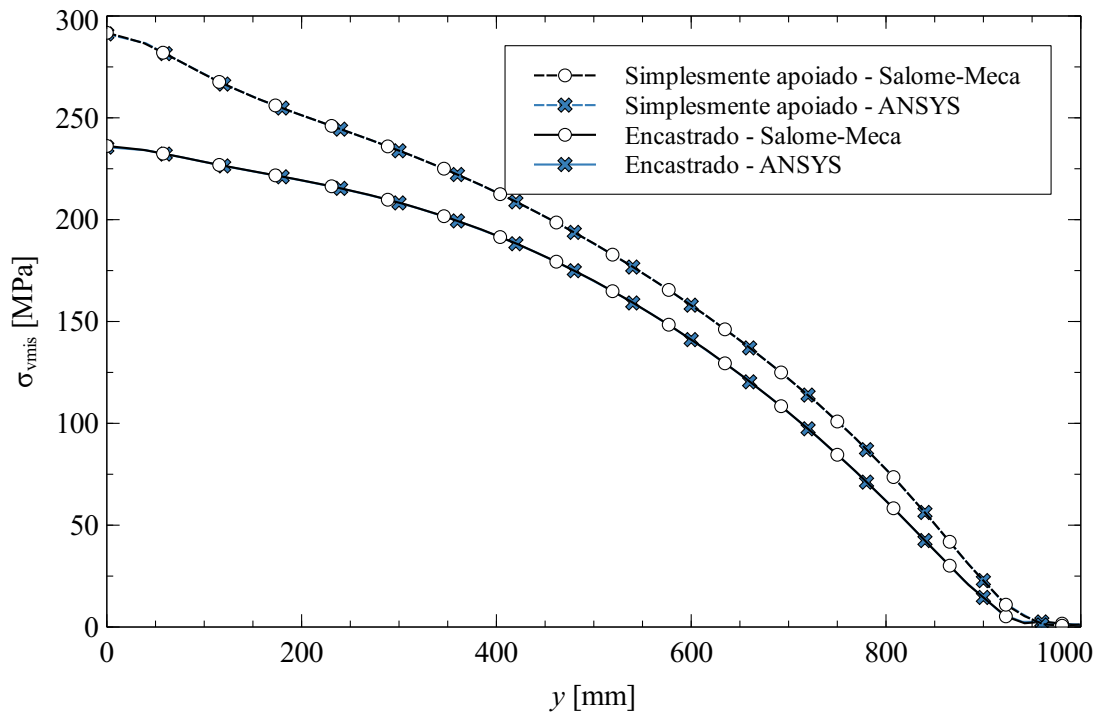


Figura 5.26: Distribuição da tensão de von Mises num reforço vertical.

ferramentas de visualização dos dois *softwares* percebe-se que os picos de tensão são menores ou melhor suavizados no *software* ANSYS, o que leva a concluir que este *software* tem algoritmos de um tratamento destes resultados.

Para além da comparação dos resultados obtidos nos dois *softwares* também foi realizado no *software* ANSYS uma análise à encurvadura dos reforços quando sujeitos à pressão de 0,1 MPa. Esta necessidade advém da utilização de reforços com uma altura bastante superior à espessura, 20 vezes superior, que é também o limite máximo imposto por procedimentos internos da empresa Efacec de forma a não ocorrer este fenómeno. Como este rácio se encontra precisamente no limite, realizou-se uma análise de encurvadura utilizando valores próprios a partir das ferramentas do ANSYS. Este cálculo resulta num multiplicador de carga (*load multiplier*) que permite verificar qual a carga crítica para ocorrer encurvadura. Na tabela 5.16 é possível verificar os multiplicadores de carga calculados para o primeiro modo de encurvadura, tanto para o caso simplesmente apoiado como para o caso encastrado.

Como se verifica que os multiplicadores de carga são maiores que 1, conclui-se que a carga aplicada se encontra abaixo da carga crítica portanto não ocorre encurvadura nos reforços do painel.

Tabela 5.16: Multiplicadores de carga para primeiro modo de encurvadura para o modelo ótimo

	Simplemente apoiado	Encastrado
Multiplicador de carga	2,425	2,576

5.4 Discussão de resultados

No âmbito deste capítulo foi apresentada uma rotina que permite cálculo estrutural automático por elementos finitos de várias tipologias de painéis de cubas de transformador. Foi igualmente demonstrado que esta rotina permite alguma versatilidade relativamente à geometria, tamanho de malha, condições de fronteira, análise de resultados, entre outras funcionalidades implementadas. Ainda relativamente à rotina criada, destaca-se a capacidade de analisar vários modelos de painéis com diferentes parâmetros num curto de espaço tempo e somente com a introdução dos dados de entrada. De forma a validar a rotina criada foram realizados duas aplicações, painéis planos simples e painéis planos reforçados.

Relativamente aos painéis planos simples, verificou-se que os resultados obtidos a partir da análise de elementos finitos demonstravam uma diferença relativa a resultados teóricos bastante baixa, menor que 1%. Ainda foram realizadas análises com diferentes parâmetros geométricos assim como condições de fronteira possíveis de implementar devido ao carácter automático da rotina criada.

No estudo dos painéis planos reforçados, foi analisada a influência dos parâmetros nos resultados obtidos, como as dimensões dos reforços, sendo que, a partir desta análise foi criada uma otimização estrutural do exemplo implementado. Esta demonstração permite observar a capacidade da rotina implementada em otimizar painéis de cubas de transformadores, obtendo assim o componente com a menor massa que permite suportar os esforços que irá estar sujeito. Ainda foram validados os resultados obtidos, comparando-os com o *software* ANSYS, em que se observa diferenças relativas tanto a nível de deslocamentos como de tensões, praticamente nulas.

Esta página foi intencionalmente deixada em branco.

Capítulo 6

Análise Estrutural de Cubas

Este capítulo irá incidir sobre a implementação de uma rotina de análise otimizada para o cálculo por elementos finitos de uma cuba simplificada de um transformador do tipo *core* de painéis planos. O objetivo da utilização de uma rotina para o cálculo de uma estrutura deste tipo é demonstrar a viabilidade de implementar geometrias mais complexas de que os painéis já estudados.

Após a discussão da rotina implementada e demonstração de um exemplo de aplicação, irá-se analisar a sensibilidade dos resultados à malha de elementos finitos utilizada. Também é de interesse comparar os resultados obtidos nos painéis de uma cuba completa com os resultados obtidos no capítulo 5 do cálculo de painéis isolados e por fim, irá se validar os modelos obtidos comparando-os com outro *software* de elementos finitos.

6.1 Rotina de análise de uma cuba otimizada

A arquitetura de uma rotina para o cálculo de uma cuba irá ser semelhante à abordada no capítulo 4, no entanto, esta rotina será mais simples do que a implementada para os painéis isolados, visto que não existe a funcionalidade de análise de vários modelos em simultâneo. Contudo, esta rotina distingue-se devido ao algoritmo de pré-processamento que terá de ter a capacidade de desenhar a cuba completa através do código programado. Após o cálculo por elementos finitos, a rotina mostra os campos de resultados obtidos através de um mapa de cores na ferramenta do Salome-Meca, o ParaView.

6.1.1 Dados de entrada

Como referido, os dados de entrada desta rotina encontram-se organizados por classes que contêm atributos e métodos. Na figura 6.1 é possível verificar as classes criadas e respetivos atributos e métodos.

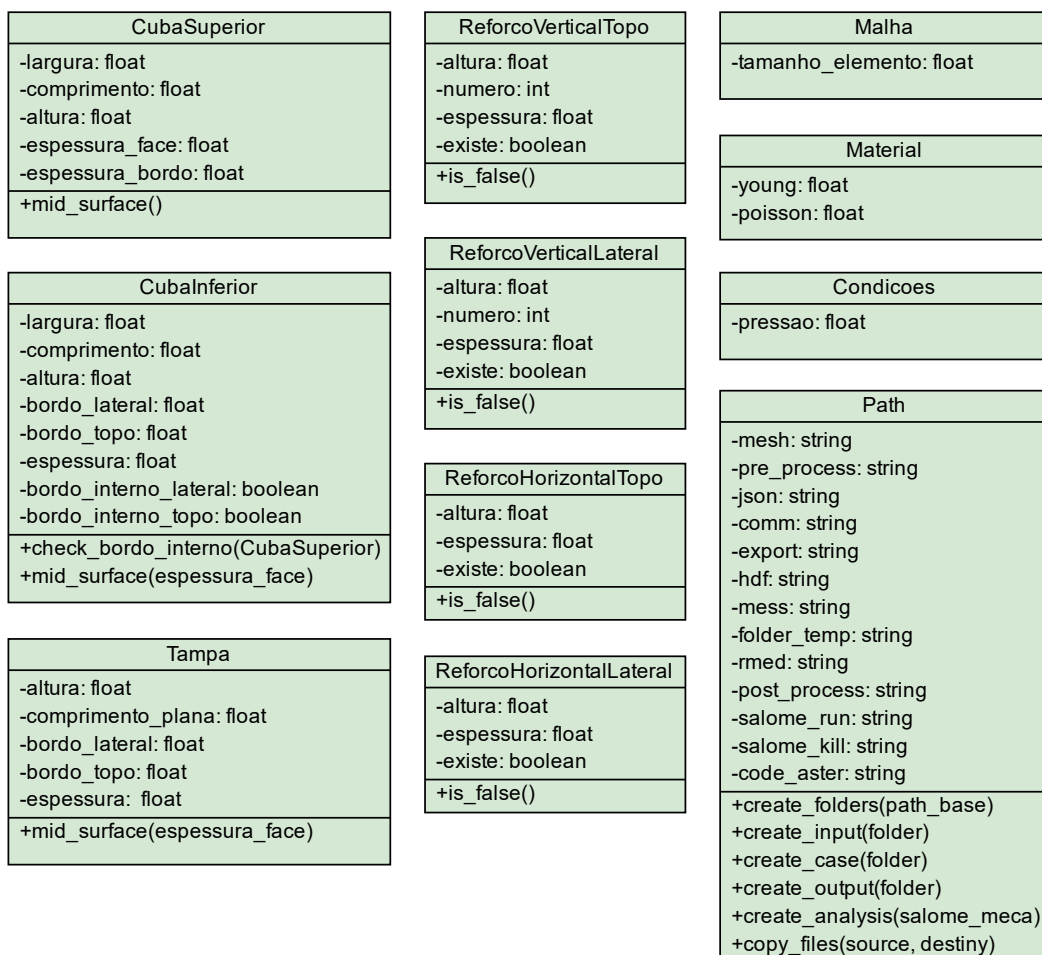


Figura 6.1: Classes da rotina de análise de cubas.

As classes utilizadas são então referentes a dados geométricos, ao tamanho de malha do modelo, ao material e por fim, aos ficheiros utilizados. Existem três grandes métodos que podem ser relativos a uma classe ou a várias, sendo eles:

- `mid_surface`: permite obter as dimensões de forma a construir o modelo por superfícies em vez de sólidos;
- `is_false`: permite retirar os reforços do modelo;

- `check_bordo_interno`: verifica se a cuba inferior tem dimensões menores que a cuba superior, e caso seja verdade, indica à rotina que existirá uma zona de ligação entre estes dois componentes sujeita a pressão.

Após o utilizador fornecer os dados de entrada à rotina, o programa cria todos os ficheiros necessários à análise e de seguida, escreve os dados das classes num ficheiro *json*.

6.1.2 Preparação de ficheiros

A rotina implementada para ser bem-sucedida necessita de um ficheiro *json* que permite guardar os dados de entrada, uma rotina de pré-processamento que cria a geometria e malha de elementos finitos da cuba, o ficheiro *comm* com a rotina de cálculo, o ficheiro *export* com a todas as informações relativas ao estudo como os ficheiros *input* e *output* do cálculo por elementos finitos, e por fim, a rotina de pós-processamento que permite visualizar o modelo na ferramenta ParaView.

6.1.2.1 Algoritmo de pré-processamento

O algoritmo de pré-processamento permite desenhar a cuba através dos dados de entrada pelo código programado. Este algoritmo utiliza os módulos Python inerentes ao *software* Salome-Meca para ser capaz de desenhar cada componente. A rotina implementada cria cada componente separadamente, sendo que no final liga todas as partes para formar a cuba.

Após ser gerada a geometria da cuba, a rotina prossegue com a malhagem do modelo, através de elementos quadrangulares com quatro nós cada elemento, designados no *software* como QUAD4. O tamanho de cada elemento é definido como dado de entrada. Para ser possível na rotina de cálculo atribuir propriedades como espessura, solicitações ou condições de fronteira é necessário criar grupos de elementos. Os grupos criados são:

- Painéis laterais da cuba superior;
- Painéis de topo da cuba superior;
- Painéis laterais da cuba inferior;
- Painéis de topo da cuba inferior;
- Tampa;

- Chapa de fundo da cuba inferior;
- Bordo superior referente à tampa;
- Bordo superior referente à cuba superior;
- Bordo inferior referente à cuba inferior;
- Bordo inferior referente à cuba superior;
- Bordo interno lateral entre a cuba inferior e superior;
- Bordo interno de topo entre a cuba inferior e superior;
- Reforços verticais laterais;
- Reforços verticais de topo;
- Reforço horizontal lateral;
- Reforço horizontal de topo;
- Faces sujeitas à pressão;
- Arestas da chapa de fundo;
- Vértices da chapa de fundo.

A direção lateral é coincidente com o painel de maior comprimento, enquanto a direção de topo é a direção normal a esta.

Após a criação dos grupos, a rotina exporta a malha obtida para um ficheiro *med* e guarda o modelo gerado num ficheiro *hdf*. Na figura 6.2 encontra-se um exemplo do resultado da implementação da rotina de pré-processamento.

6.1.2.2 Algoritmo de análise por elementos finitos

O algoritmo de análise por elementos finitos é redigido no ficheiro *comm* pela própria rotina principal, que mais tarde, durante o processo de cálculo executa este ficheiro no Code_Aster. Este algoritmo permite assim calcular o campo de deslocamentos, campo de tensões normais e ainda os campos de tensões segundo critérios, como a tensão equivalente de von Mises.

O algoritmo começa por ler a partir do ficheiro *json* todos os dados de entrada que o utilizador forneceu que irão ser necessários.

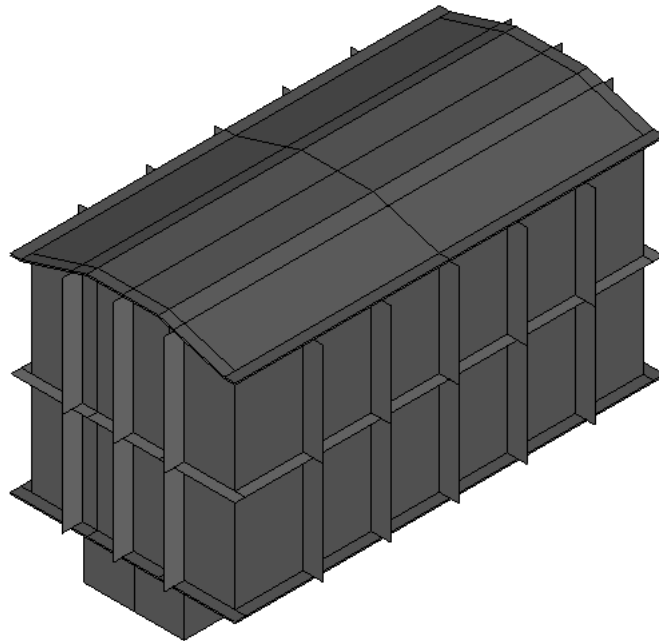


Figura 6.2: Exemplo de geometria gerada através do algoritmo de pré-processamento.

De seguida atribui a modelação relativa à Teoria de Placas, designada no *software* Code_Aster como DKT. Esta modelação utiliza elementos de placa em que cada nó tem três graus de liberdade relativos à translação e mais três graus de liberdade relativos à rotação e somente é possível utilizar elementos com quatro nós cada, ou seja QUAD4. Esta modelação, ao contrário da implementada no capítulo anterior, não contabiliza uma curvatura inicial da superfície, no entanto, é possível de utilizar visto que não existem componentes com as faces curvas. Após atribuído o modelo, indica-se a espessura de cada componente através da função `AFFE_CARA_ELEM`, [29].

A pressão é dada relativa ao grupo de elementos criado no pré-processamento que corresponde às faces da cuba superior e inferior e à tampa. De seguida, atribui-se as condições de fronteira à cuba, em que nesta rotina considera-se as arestas da chapa de fundo numa situação de encastramento. Por fim, de forma a ligar rigidamente a cuba inferior com a cuba superior e a cuba superior com a tampa, utiliza-se o argumento `LIAISON_MAIL` em que para cada ligação indica-se os respetivos bordos a ligar. Esta ligação projeta os nós de um grupo de elementos em outro, e cria relações lineares entre os dois grupos de forma a criar restrições entre esses componentes, [31].

A partir deste ponto a rotina já se encontra capaz de analisar estaticamente o modelo de forma a obter os deslocamentos. Após se obter os deslocamentos a

rotina também calcula as tensões normais e tensão equivalente de von Mises em cada nó. Por fim, exporta-se todos os campos de resultados para um ficheiro *med* identificado pelo seu *Logical Unite Number* que mais tarde permite visualizar os resultados obtidos.

6.1.2.3 Algoritmo de pós-processamento

O algoritmo de pós-processamento possibilita a leitura do ficheiro *med* de *output* para visualizar o campo de resultados na ferramenta ParaView. Na figura 6.3 encontra-se um exemplo do *output* da rotina implementada para visualização dos resultados

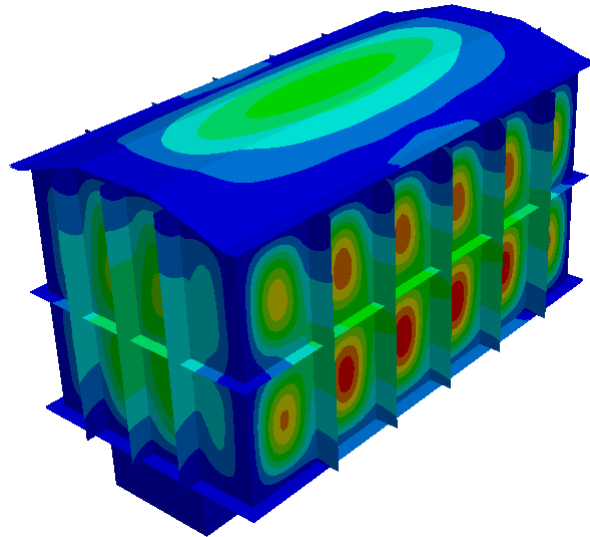


Figura 6.3: Exemplo da visualização de resultados utilizando a rotina implementada para análise estrutural de cubas.

6.2 Aplicação

Após a rotina estar estabelecida é do interesse deste trabalho validar e demonstrar a sua aplicabilidade. De forma a simplificar a nomenclatura utilizada demonstra-se na figura 6.4 uma representação geométrica das dimensões e respectivas variáveis dos modelos implementados por esta rotina.

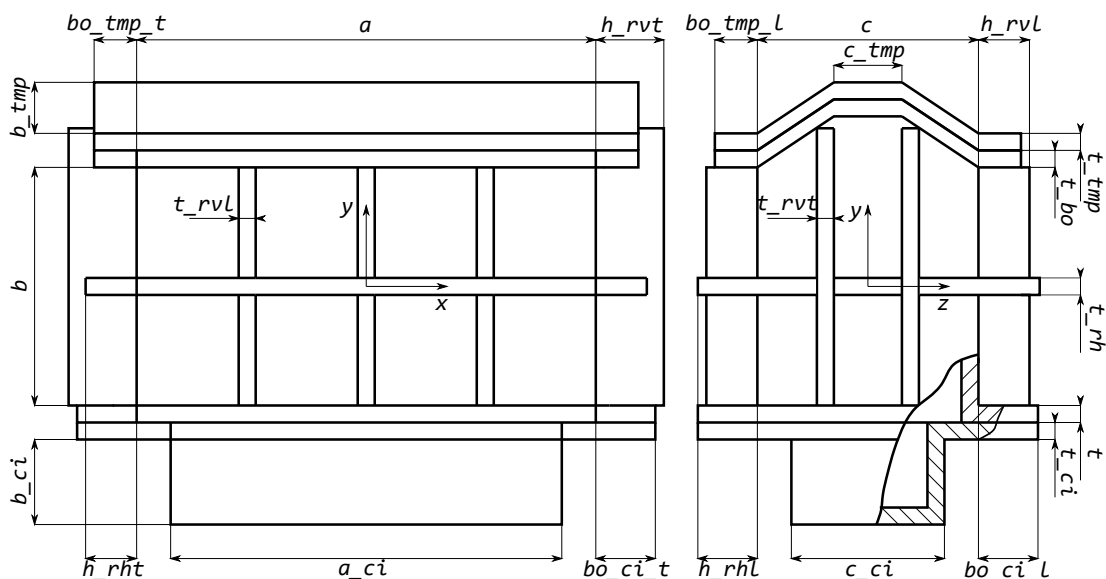


Figura 6.4: Representação geométrica de uma cuba simplificada.

Na tabela 6.1 é possível visualizar o significado das variáveis geométricas representadas na figura 6.4 e que terão de ser fornecidas à rotina como dados de entrada.

Tabela 6.1: Variáveis geométricas de uma cuba simplificada

Cuba Superior		Cuba Inferior		Tampa	
a	largura	a_{ci}	largura	b_{tmp}	altura
b	altura	b_{ci}	altura	c_{tmp}	comprimento
c	comprimento	c_{ci}	comprimento	t_{tmp}	espessura
t	espessura	t_{ci}	espessura	bo_{tmp_t}	bordo topo
t_{bo}	espessura bordo	bo_{ci_t}	bordo topo	bo_{tmp_l}	bordo lateral
		bo_{ci_l}	bordo lateral		
Reforço Vertical Lateral		Reforço Vertical Topo		Reforço Horizontal	
n_{rvl}	número	n_{rvt}	número	h_{rhl}	altura lateral
h_{rvl}	altura	h_{rvt}	altura	h_{rht}	altura topo
t_{rvl}	espessura	t_{rvt}	espessura	t_{rh}	espessura

O exemplo de aplicação será com base nos painéis planos reforçados analisados na secção 5.3, de forma a comparar os resultados obtidos para os painéis de uma cuba e os painéis analisados individualmente. Assim na tabela 6.2 é possível visualizar os valores das variáveis que foi atribuído como exemplo de aplicação.

Tabela 6.2: Propriedades de um exemplo de uma cuba de um transformador

Geometria					
a [mm]	4000	a_{ci} [mm]	4000	b_{tmp} [mm]	200
b [mm]	2000	b_{ci} [mm]	1000	c_{tmp} [mm]	700
c [mm]	2000	c_{ci} [mm]	500	t_{tmp} [mm]	20
t [mm]	8	t_{ci} [mm]	20		
n_{rvl}	5	n_{rvt}	3	h_{rhl} [mm]	100
h_{rvl} [mm]	200	h_{rvt} [mm]	200	h_{rht} [mm]	100
t_{rvl} [mm]	10	t_{rvt} [mm]	10	t_{rh} [mm]	5
bo_{tmp_t} [mm]	100	Material		Solicitação	
bo_{tmp_l} [mm]	100	E [GPa]	210	p [MPa]	0,1
bo_{ci_t} [mm]	100	ν	0,3	Massa	
bo_{ci_l} [mm]	100	ρ [kg/m ³]	7850	m [kg]	6006,1
t_{bo} [mm]	10	σ_Y [MPa]	355		

6.3 Análise da sensibilidade da malha de elementos finitos

De forma a verificar o tamanho de elemento necessário de forma a garantir que os resultados obtidos são fiáveis e não dependem deste fator procedeu-se com uma análise em que se varia o tamanho do elemento e verifica-se a diferença entre os resultados obtidos. A geometria utilizada é a já referida na tabela 6.2.

Como referência utiliza-se o deslocamento máximo obtido que será correspondente à deformação dos painéis laterais da cuba, mais precisamente a superfície mais próxima do reforço a meio do painel. Ainda se utiliza para o cálculo das tensões, a tensão de von Mises máxima obtida na aresta de ligação do reforço do meio com o painel. Esta análise também permite verificar eventuais zonas onde ocorram picos de tensão, portanto, com o refinamento da malha de elementos finitos, estes fenómenos irão intensificar-se e tornar-se cada vez mais locais.

O critério para obter um modelo em que se considera que tamanho de elemento já não tem uma influência considerável será quando a diferença relativa dos resultados

obtidos entre refinamentos subsequentes for menor que 1%. Na tabela 6.3 é possível verificar os resultados obtidos para cada tamanho.

Tabela 6.3: Resultados obtidos na análise da sensibilidade da malha de uma cuba

Tamanho do elemento [mm]	Nº nós	w [mm]	dif (%)	σ_{vmis} [MPa]	dif (%)	Tempo [s]
160	3526	7,043	-	219,906	-	4,6
80	11794	7,234	2,712	220,59	0,311	11,0
40	40764	7,288	0,746	221,951	0,617	49,5
20	154084	7,316	0,384	222,336	0,173	152,7
10	596358	7,315	0,014	222,447	0,059	1249,25

Após um refinamento gradual da malha de elementos finitos, observa-se que o tamanho de elemento não demonstra uma influência grande nos resultados obtidos a partir de 40 mm, visto que a diferença relativa dos resultados é menor que 1%. Ainda se demonstra que um tamanho de elemento excessivamente pequeno resulta num tempo de cálculo bastante elevado o que não é desejável. Na figura 6.5 demonstra-se a malha com o tamanho de elemento 40 mm assim como uma ampliação para uma melhor compreensão.

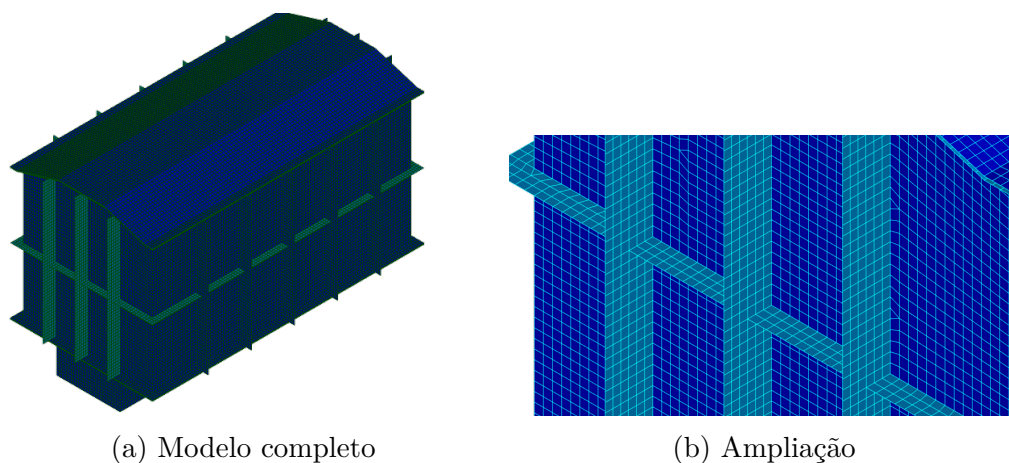


Figura 6.5: Malha de elementos finitos de uma cuba com tamanho de elemento igual a 40 mm.

A partir desta análise também se verificou que em certos locais, a tensão de von Mises aumenta exponencialmente com o refinamento sendo que pode ser resultado de diversas razões no entanto indica-se neste modelo da cuba que ocorre devido a nós demasiado restringidos como pontos de ligação entre diferentes geometrias. Estes picos de tensão devem ser tratados com cuidado, mas principalmente devem

ser identificados, sendo que após esta análise verificou-se que estes picos ocorrem nos seguinte locais:

- Ponto de ligação entre painel lateral, painel de topo e reforço horizontal;
- Pontos de ligação entre reforços verticais, painéis da cuba superior e bordo da tampa ou bordo da cuba inferior;
- Pontos de ligação entre painel lateral cuba inferior, painel de topo da cuba inferior e bordo da cuba inferior.

O local onde existe a ligação entre os painéis da cuba superior e o reforço horizontal verifica-se como o que apresenta picos de tensão mais elevados. Um exemplo deste fenómeno para o tamanho de elemento de 20 mm pode ser observado na figura 6.6. Nesta figura compreende-se que no local de ligação ocorre uma tensão mais elevada sendo que na vizinhança, por comparação, a tensão diminui drasticamente

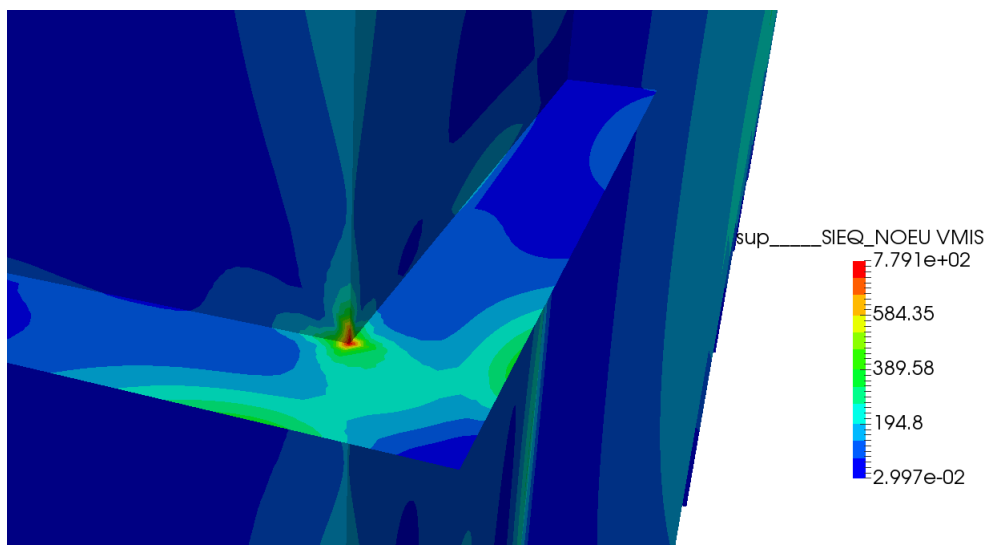


Figura 6.6: Exemplo de local onde ocorre tipicamente picos de tensão na análise de uma cuba.

6.4 Resultados e validação

Após verificar a malha adequada para o exemplo de aplicação, é necessário realizar uma análise aos resultados calculados através da rotina implementada. De igual forma foi calculado os resultados no *software* ANSYS, utilizando a mesma geometria, tamanho de elemento e número de nós por elemento assim como as mesmas

condições de análise e ligações entre componentes. Estes resultados para efeitos de comparação encontram-se na figura 6.7 relativamente aos mapas de cores dos deslocamentos totais calculados e na figura 6.8 relativamente aos mapas de cores do campo de tensões von Mises. O intervalo de valores assim como a escala de cores é mesma nos dois *softwares* para cada exemplo, portanto é possível observar as diferenças entre os dois.

Como primeira observação a partir das figuras indicadas salienta-se o facto de entre ambos os *softwares* as diferenças serem praticamente nulas devido à semelhança das distribuições de cores por toda a estrutura. Ainda se indica que é possível perceber que o modelo se encontra bastante sobre dimensionado devido à tensão von Mises em certas zonas, como a cuba inferior e a tampa, ser bastante menor que a tensão máxima admissível, portanto este modelo seria passível de uma otimização da geometria.

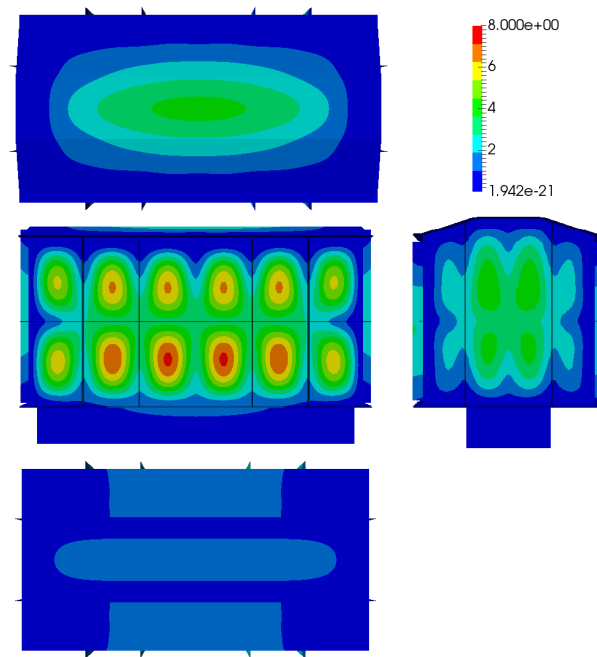
De forma a melhor perceber as diferenças entre os resultados obtidos nos dois *softwares* comparou-se os valores em zonas específicas. Em relação aos deslocamentos, na tabela 6.4 encontram-se os deslocamentos máximos nas três direções obtidos a partir de cada *software* e calculou-se a diferença do resultado utilizando o Salome-Meca relativamente ao resultado utilizando o ANSYS.

Tabela 6.4: Deslocamentos máximos obtidos para cada direção em diferentes *softwares*

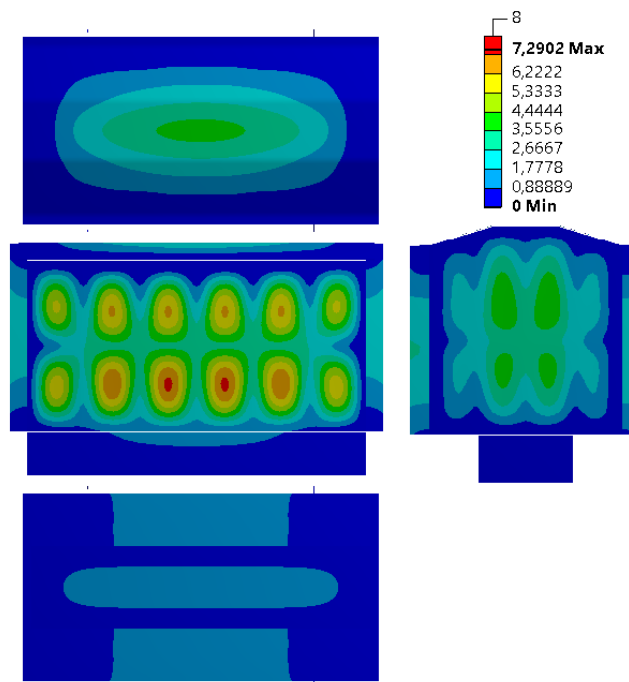
	ANSYS	Salome-Meca	<i>dif</i> (%)	Local
<i>u</i> [mm]	7,290	7,288	0,027	<i>painel_lateral</i>
<i>v</i> [mm]	4,327	4,331	0,092	<i>painel_topo</i>
<i>w</i> [mm]	3,788	3,809	0,554	<i>tampa</i>

A tabela verifica que os deslocamentos máximos obtidos para cada direção são praticamente iguais nos dois *softwares*.

No que respeita à comparação das tensões obtidas utilizou-se a distribuição dos resultados obtidos nos dois *softwares* em diferentes zonas da cuba. As zonas de referência serão a aresta superior do reforço vertical a meio do painel lateral, a aresta inferior de ligação entre o reforço vertical a meio do painel e o painel lateral e ainda uma linha pertencente ao painel lateral entre o reforço vertical a meio e o reforço imediatamente a seguir. Na figura 6.9 é possível observar estas distribuições para cada zona e para cada *software*.

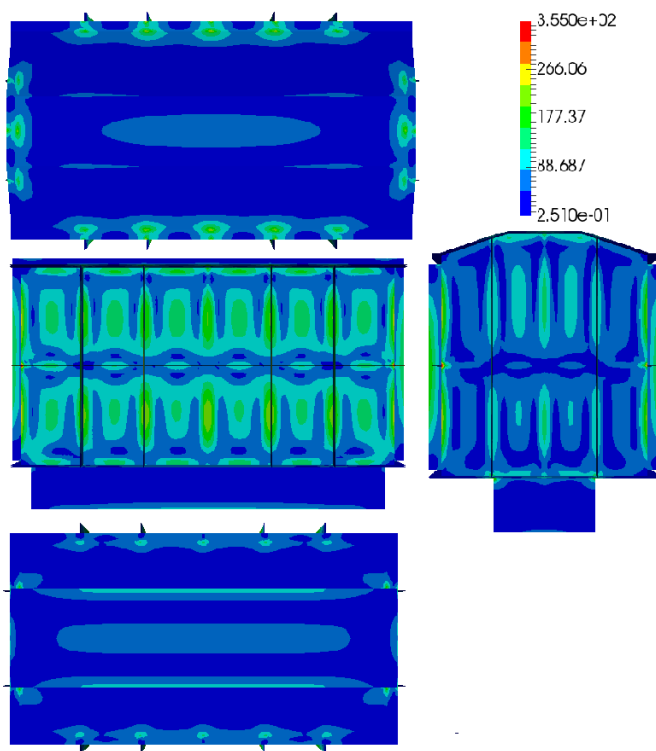


(a) Salome-Meca

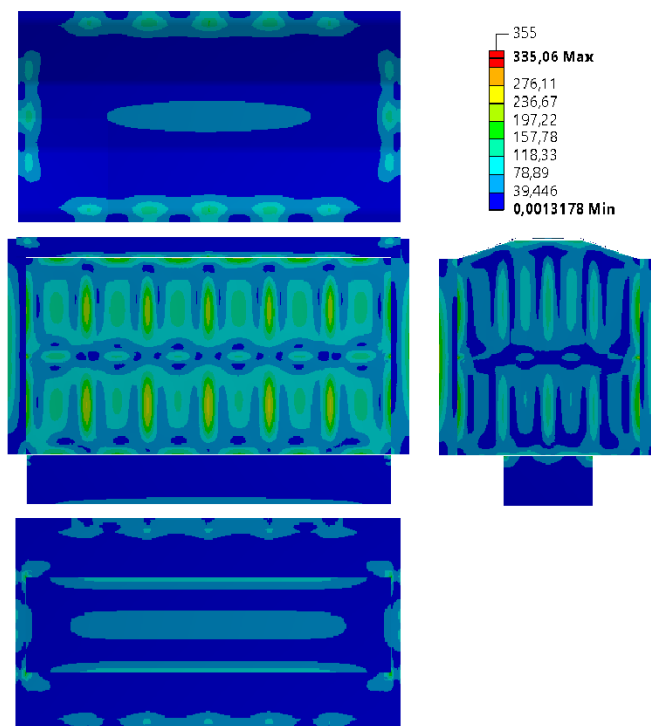


(b) ANSYS

Figura 6.7: Campo de resultados dos deslocamentos para exemplo de aplicação de análise de uma cuba.



(a) Salome-Meca



(b) ANSYS

Figura 6.8: Campo de resultados da tensão de von Mises para exemplo de aplicação de análise de uma cuba.

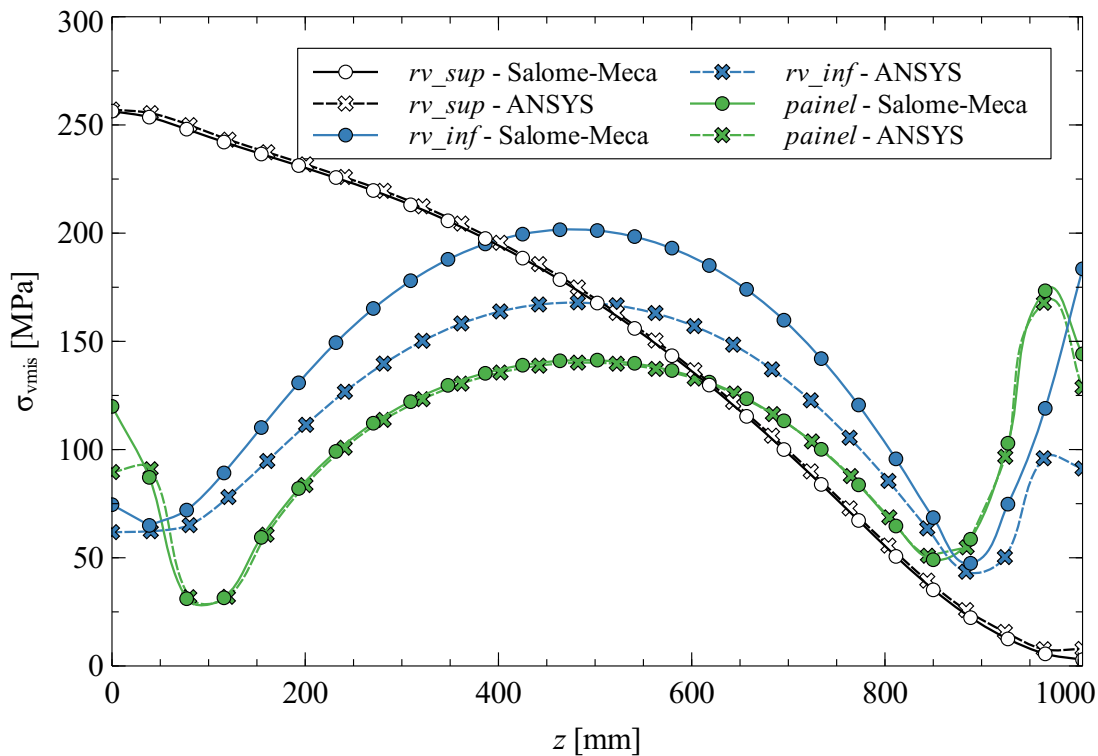


Figura 6.9: Distribuição da tensão de von Mises obtida por diferentes *softwares*.

Os resultados apresentados na figura 6.9 indicam que a diferença entre as tensões obtidas é praticamente nula na aresta superior do reforço vertical e ainda na linha pertencente ao painel, o que desde já demonstra a primeira observação realizada que os dois *softwares* descrevem o comportamento mecânico da estrutura de forma bastante semelhante. No entanto, também é possível observar uma diferença relativamente grande do resultado do Salome-Meca em relação ao ANSYS (cerca de 20%) na distribuição da tensão na aresta de ligação entre o reforço vertical e o painel lateral. Como nas outras zonas analisadas a tensão é praticamente igual, conclui-se que esta divergência deve-se aos resultados obtidos especificamente nas ligações. Diversas razões poderão resultar na diferença de resultados como uma diferença entre os dois *softwares* na restrição dos graus de liberdade entre os dois corpos ou ainda algoritmos diferentes para realizar a assemblagem das tensões relativas a cada corpo. Esta diferença apesar de excessiva poderá ser contornada tentando implementar as mesmas rotina de cálculo ou pós-processamento em cada *software*. Apesar da discrepância nestas zonas específicas de ligação considera-se que no resto da cuba os dois *softwares* calculam resultados tanto de deslocamentos, como de tensões, praticamente iguais.

6.5 Comparação de resultados com a análise painel plano reforçado

No âmbito desta dissertação foram implementadas duas rotinas de análise de forma automática, uma relativa a painéis e outra relativa a uma cuba simplificada. Será do interesse deste trabalho comparar os resultados obtidos para a análise de um painel plano reforçado para o caso em que se encontra simplesmente apoiado e para o caso em que se encontra encastrado com os resultados obtidos para um painel na análise da cuba completa.

Em relação aos deslocamentos, o comportamento do painel assemelha-se mais ao painéis encastrados do que aos painéis simplesmente apoiados, visto que o valor máximo ocorre entre os reforços do meio em ambos os casos, enquanto no simplesmente apoiado ocorre nas extremidades do painel. Em termos de valores, encontra-se organizado na tabela 6.5 o valor obtido para cada caso.

Tabela 6.5: Deslocamento máximo obtidos para um painel com diferentes condições de fronteira

	Cuba	Encastrado	Simplesmente apoiado
w [mm]	7,288	6,585	9,604

Como se pode observar, o valor máximo aproxima-se mais do caso encastrado. No entanto, o deslocamento no caso de simplesmente apoiado, no mesmo local onde ocorre o encastrado e o do painel da cuba, tem um valor de 6,966 mm, o que demonstra que se considerar-se somente um painel entre reforços do caso simplesmente apoiado, este terá um comportamento semelhante aos outros a nível de deslocamentos.

Em relação às tensões de von Mises, o painel da cuba terá um comportamento bastante semelhante ao painel encastrado nas suas extremidades devido às restrições impostas pelo contacto com outros componentes. No resto do painel da cuba, este será uma situação intermédia dos dois casos. Na figura 6.10 é possível observar a distribuição da tensão em vários locais, para os três casos.

Como se pode observar, a análise de painel de uma cuba aproxima-se bastante dos resultados obtidos utilizando a análise de um painel isolado com diferentes condições de fronteira. Sendo que as distribuições da situação do painel encastrado assemelham-se às distribuições da cuba completa de uma forma mais aproximada que no caso simplesmente apoiado.

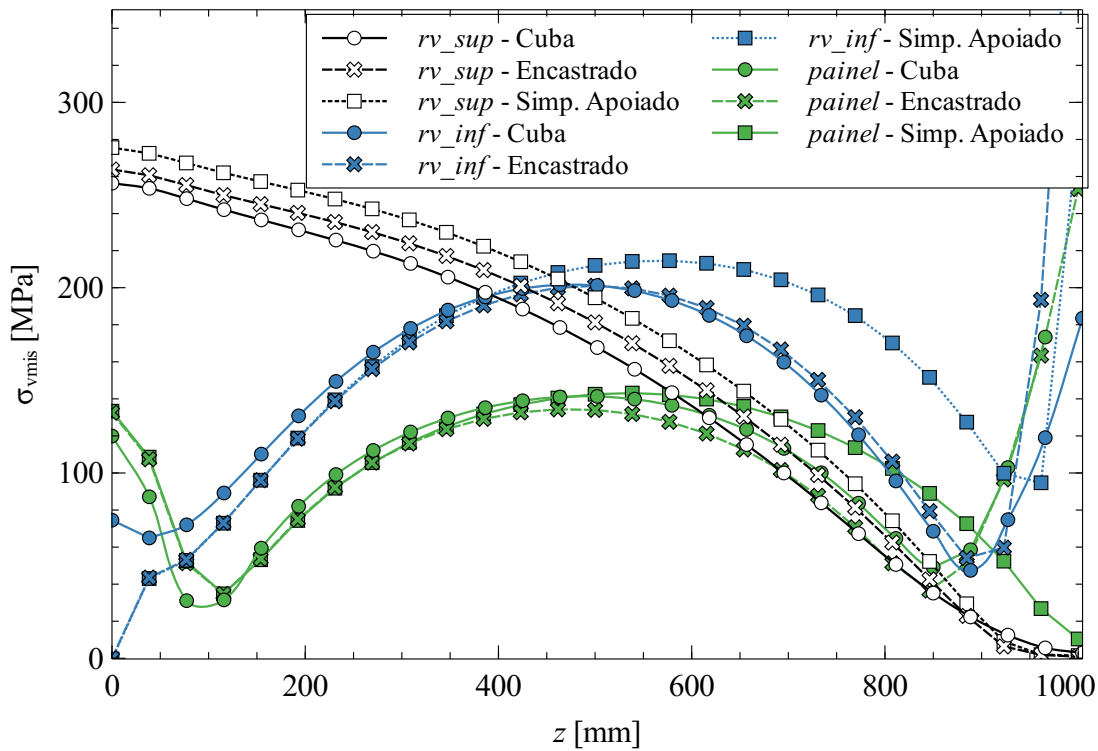


Figura 6.10: Distribuição da tensão de von Mises obtida com diferentes condições de fronteira.

6.6 Comparação com painéis simples

Nesta secção irá ser utilizada a contribuição das duas rotinas para traçar comparações entre um painel simples plano isolado e um painel simples plano de uma cuba completa. A escolha de painéis simples deve-se à complexidade de painéis reforçados já que existe um maior número de variáveis que poderão influenciar os resultados.

Nesta análise paramétrica somente se irá variar a relação entre os dois lados do painel lateral, à semelhança da análise realizada na secção 5.2.4. O resto dos valores encontram-se na tabela 6.6. As propriedades do material e solicitação serão iguais às referidas na tabela 6.2.

Tabela 6.6: Propriedades da cuba para comparação com painel simples

a [mm]	1000	a_{ci} [mm]	1000	b_{tmp} [mm]	100
c [mm]	500	c_{ci} [mm]	500	c_{tmp} [mm]	200
t [mm]	10	b_{ci} [mm]	200	t_{tmp} [mm]	10
bo_{tmp_t} [mm]	50	bo_{ci_t} [mm]	50	t_{ci} [mm]	10
bo_{tmp_l} [mm]	50	bo_{tmp_l} [mm]	50	t_{bo} [mm]	10

A altura da cuba superior, b , será definida a partir do comprimento do painel lateral, a , e o rácio entre estas duas dimensões.

Nesta análise irá ser utilizado o deslocamento e a tensão de von Mises no ponto central do painel, sendo que ainda serão calculados três coeficientes de forma a avaliar a relação entre estes valores. O coeficiente que determina a relação entre os valores da cuba e o painel simplesmente apoiado, x_{sap} , um coeficiente que determina uma relação entre os valores da cuba e o painel encastrado, x_{enc} , e ainda um coeficiente que relaciona os três valores obtidos, x_{tot} .

$$x_{sap} = \frac{p_{sap} - p_{cuba}}{p_{cuba}} \quad (6.1)$$

$$x_{enc} = \frac{p_{cuba} - p_{enc}}{p_{cuba}} \quad (6.2)$$

$$\begin{aligned} p_{cuba} &= (1 - x_{tot}) p_{enc} + x_{tot} p_{sap} \equiv \\ &\equiv x_{tot} = \frac{p_{cuba} - p_{enc}}{p_{sap} - p_{enc}} \end{aligned} \quad (6.3)$$

Na equação 6.1, p_{sap} , p_{enc} e p_{cuba} são os valores obtidos para o caso simplesmente apoiado, caso encastrado e para o painel lateral da cuba. Na tabela 6.7 e 6.8 encontram-se os valores obtidos para o deslocamento e tensão de von Mises, respetivamente. Para cada rácio de painel foi calculado os três coeficientes.

Tabela 6.7: Deslocamentos para diferentes dimensões de painéis simples com diferentes condições de fronteira

k	w_{sap} [mm]	w_{enc} [mm]	w_{cuba} [mm]	x_{sap} (%)	x_{enc} (%)	x_{tot} (%)
1,0	21,316	6,593	10,217	108,633	35,470	24,615
1,1	17,453	5,367	8,505	105,209	36,896	25,964
1,2	14,294	4,332	7,098	101,381	38,969	27,766
1,3	11,744	3,487	5,956	97,179	41,454	29,902
1,4	9,675	2,806	5,025	92,537	44,159	32,305
1,5	8,023	2,268	4,277	87,585	46,972	34,909
1,6	6,657	1,832	3,653	82,234	49,849	37,741
1,7	5,549	1,487	3,141	76,664	52,658	40,719
1,8	4,670	1,221	2,729	71,125	55,258	43,723
1,9	3,917	0,999	2,371	65,205	57,866	47,019
2,0	3,322	0,828	2,082	59,558	60,231	50,281

Tabela 6.8: Tensão von Mises para diferentes dimensões de painéis simples com diferentes condições de fronteira

k	σ_{sap} [MPa]	σ_{enc} [MPa]	σ_{cuba} [MPa]	x_{sap} (%)	x_{enc} (%)	x_{tot} (%)
1,0	289,557	137,851	173,457	66,933	20,527	23,470
1,1	263,334	124,935	157,680	67,005	20,767	23,660
1,2	241,229	113,367	144,252	67,227	21,410	24,155
1,3	222,389	103,181	132,801	67,460	22,304	24,847
1,4	205,670	93,871	122,831	67,441	23,577	25,904
1,5	190,905	85,575	114,217	67,142	25,077	27,193
1,6	177,384	78,107	106,411	66,697	26,599	28,510
1,7	164,908	71,180	99,448	65,823	28,425	30,160
1,8	153,773	65,097	93,340	64,745	30,258	31,850
1,9	143,043	59,353	87,569	63,349	32,221	33,715
2,0	133,136	54,376	82,530	61,318	34,114	35,747

No caso dos deslocamentos, o valor obtido para a cuba no centro do painel lateral aproxima-se praticamente para todas dimensões da situação de encastramento (x_{tot} mais próximo dos 0%). No entanto, as dimensões do painel irão influenciar bastante os resultados obtidos nos três casos, sendo que, observa-se que nenhum dos coeficientes se manteve constante, portanto a relação entre a tensão das cubas e as outras duas não é linear.

Em relação à tensão von Mises os coeficientes calculados permanecem dentro de uma gama menor do que no caso deslocamentos. O comportamento no centro do painel lateral da cuba aproxima-se bastante do caso encastrado em todas as dimensões, com uma diferença relativa a este valor de cerca de 25% (x_{enc}). O coeficiente que relaciona a tensão no centro com o caso simplesmente apoiado demonstra que a diferença relativa ao painel lateral da cuba apesar de relativamente mais alta mantém-se mais constante com estudo dos outros casos.

Importante referir que este estudo foi só para o caso de um painel simples, sendo que num caso painel reforçado existirão um maior número de parâmetros a influenciar o comportamento do painel. Também se salienta que a cuba tem bastantes componentes que poderão surgir efeito nestes resultados, como as dimensões do painel de topo, ou ainda as espessuras da tampa e da cuba inferior. Será do interesse de trabalhos posteriores estudar a influência de cada componentes de forma a perceber se é possível criar valores tabelados que permitisse obter o comportamento do painel de uma cuba utilizando somente análise de painéis isolados. As análises de painéis isolados permite maior rapidez e fácil implementação.

6.7 Discussão de resultados

Neste capítulo abordou-se a implementação de uma rotina de cálculo automático de cuba simplificada utilizando o método dos elementos finitos. Demonstrou-se que com esta rotina será possível analisar de forma expedita cubas de diferentes dimensões. Este tipo de rotina permite então realizar cálculos de deformações e tensões, assim como subsequentes otimizações estruturais, com maior eficiência e maior rapidez comparando com outros métodos

Foram implementadas aplicações desta rotina e devidamente validadas comparando-as com o *software* ANSYS. As conclusões possíveis de retirar desta análise são que os dois *softwares* em praticamente toda a estrutura apresentam diferenças, tanto em deslocamentos como em tensões praticamente nulas. No entanto, em ligações entre dois componentes diferentes a diferença é relativamente maior (cerca de 20%), devido a diferentes algoritmos de computação de resultados pertencentes a nós de ligação.

Neste capítulo foi também comparado painéis isolados tanto em simplesmente apoiado como no caso encastrado. Os resultados demonstraram que o painel de uma cuba, aproxima-se mais do caso encastrado, no entanto, será sempre uma situação intermédia dos dois casos. Também se procedeu à obtenção de fatores que relacionam um painel simples isolado com o painel lateral simples de uma cuba, sendo que, nesta análise concluiu-se que a relação não é linear e depende de vários parâmetros. No entanto, só será possível obter uma previsão mais próxima da realidade com um maior número de exemplos e assim criar um modelo que permite a partir da análise de um painel prever o comportamento da cuba. Isto é alcançável através de uma rotina de cálculo automático visto permitir a análise de um grande número de modelos.

Esta página foi intencionalmente deixada em branco.

Capítulo 7

Conclusões e Trabalhos Futuros

7.1 Conclusões

No âmbito desta dissertação foram explorados os *softwares* de código aberto Code_Aster e Salome-Meca, sendo possível confirmar as suas funcionalidades e potencial para cálculo estrutural em projetos industriais, permitindo a automatização de procedimentos de elaboração dos modelos e reduzir a dependência de licenciamento para software comercial.

Como primeira abordagem, implementou-se um exemplo de uma placa com furo central de forma a criar bases de utilização do *software* e validar este exemplo. O modelo implementado foi devidamente validado e comparado com resultados teóricos. Os resultados apresentaram diferenças bastante baixas portanto é possível concluir que o *software* permite a análise deste tipo de modelos. De igual forma foram realizadas várias análises para efeitos de comparação paramétrica de diferentes modelos.

Devido ao carácter repetitivo de análises estruturais a modelos geométricos, em que somente se varia certas condições ou aspetos da geometria, decidiu-se criar rotinas de cálculo automático em que o utilizador só teria de fornecer os dados de entrada. O Code_Aster e o Salome-Meca revelaram-se *softwares* que permitem com alguma facilidade incluir rotinas que executam o pré-processamento, cálculo por elementos finitos e o pós-processamento. No âmbito desta implementação foi criada uma arquitetura generalista de uma rotina deste tipo que permite implementar diferentes geometrias, diferentes malhas, diferentes condições de análises para cálculo, como formulações, tipos de análises, solicitações, e ainda diferentes condições de fronteira. No entanto, a mesma rotina de cálculo automático estará sempre limitada às funcionalidades incluídas nela, portanto, existe sempre necessidade de criar um

algoritmo novo caso o utilizador pretenda fazer análises de modelos geométricos ou condições que não estejam previstos nessa rotina.

De seguida foi implementada uma rotina de cálculo automático para painéis de cubas de transformadores, em que se observou que é possível analisar estes modelos de forma mais eficiente e com bastante maior rapidez. Como esta rotina permite a análise simultânea de vários modelos geométricos é possível criar análises paramétricas e otimizações estruturais em pouco tempo. Foram validados dois exemplos desta rotina, os painéis planos simples e os painéis planos reforçados. As validações realizadas comparando tanto com resultados teóricos, no caso dos painéis planos simples, como com resultados de outro *software* de elementos finitos, no caso de painéis planos reforçados, demonstraram diferenças praticamente nulas. Portanto conclui-se que este *software*, e por consequência, as rotinas criadas, permitem uma análise com resultados fidedignos. Utilizando a rotina criada foi de igual forma possível realizar várias análises paramétricas, em que se observou a influência nos resultados de certos parâmetros, como geométricos e condições de fronteira. No caso dos painéis planos reforçados, variou-se os parâmetros dos reforços verticais e reforço horizontal e a partir desta análise foi possível proceder uma otimização em que se obteve um modelo com menor massa mantendo um comportamento mecânico dentro do admissível.

Por fim, abordou-se a implementação da rotina de cálculo automático para análise de uma cuba simplificada. Os resultados apresentados foram comparados com o *software* ANSYS, onde se observou uma boa correspondência em quase toda a cuba. No entanto, verificou-se que elementos de ligação, como entre reforços e painéis, as tensões apresentadas diferiam nos dois *softwares*. Contudo, devido a haver diferenças praticamente nulas nos deslocamentos e tensões no resto da cuba, concluí-se que esta discrepância deve-se a uma diferença na computação de resultados nestes nós de ligação. Como no caso do Code_Aster estas tensões na ligação são superiores, justifica-se o seu uso como válido visto que um dimensionamento utilizando estes valores resultava num dimensionamento mais seguro, para além de a cuba apresentar diferenças baixas em relação ao ANSYS em outros locais.

No âmbito da rotina de cálculo automático da cuba, ainda foram realizadas análises paramétricas em que se comparou diversos resultados com a análise de painéis isolado. Foi possível verificar que o painel de uma cuba encontra-se numa situação intermédia entre o caso simplesmente apoiado e encastrado, no entanto, na generalidade mais próximo do encastrado. Também foi realizada uma análise de forma a verificar a existência de alguma relação entre os resultados obtidos para cada caso. Nesta análise concluiu-se que não é possível tirar ilações diretas devido

ao número de parâmetros que influenciam os resultados, no entanto, com uma rotina de cálculo deste gênero é possível obter grandes quantidades de dados facilmente que permite criar previsões deste tipo de comportamento.

Esta dissertação permitiu uma nova abordagem às análises estruturais por elementos finitos completamente automatizada utilizando recursos de código aberto e usando o potencial da programação neste tipo de processos. Com o desenvolvimento de computadores que permitem análises com maior rapidez, estas rotinas poderão fazer parte do cotidiano industrial e permitir assim que análises de elementos finitos sejam procedimentos frequentes no projeto de algum componente. Esta implementação demonstra maior eficiência, menores custos e maior otimização estrutural de componentes mecânicos. Com este tipo de rotinas ainda é possível integrar algoritmos de análise de dados e por consequência estabelecer modelos de previsão de resultados.

7.2 Trabalhos futuros

A área em que este projeto se inclui permite grandes desenvolvimentos e contribuições, principalmente devido à constante evolução tecnológica da indústria. No âmbito desta dissertação indicam-se nos seguintes tópicos possíveis trabalhos futuros:

- Desenvolvimento de rotinas para modelos geométricos mais complexos;
- Desenvolvimento de rotinas para outro tipo de análises, como análises não-lineares ou análises vibro-acústicas;
- Integração de algoritmos de geração e análise de bases dados resultantes de análises passadas que permitam criar modelos de bases estatísticas para previsão de resultados;
- Realização de análises estruturais remotamente através de servidores e, consequentemente, visualizar os resultados obtidos através de ferramentas em ambiente *web*.

Esta página foi intencionalmente deixada em branco.

Referências

- [1] Electricity explained. <https://www.eia.gov/energyexplained/electricity/delivery-to-consumers.php>. Acedido em 10-08-2019.
- [2] The open source definition. <https://opensource.org/osd>. Acedido em 13-08-2019.
- [3] Salome-meca 2018 for windows. <https://code-aster-windows.com>. Acedido em 13-08-2019.
- [4] M. Akbari, M. Allahbakhshi, e R. Mahmoodian. Heat analysis of the power transformer bushings in the transient and steady states considering the load variations. *Applied Thermal Engineering*, 121:999 – 1010, Maio 2017.
- [5] K. Bathe. *Finite Element Procedures*. Prentice Hall, Pearson Education, Inc., Segunda edição, 2016.
- [6] R. Budynas e K. Nisbett. *Shigley's Mechanical Engineering Design*. McGraw-Hill Education, 2010.
- [7] OPEN CASCADE. Salome version 8.3.0, 2017. Disponível em <https://www.salome-platform.org>.
- [8] C. Durand. Free software for computational mechanics: EDF's choice. <https://www.code-aster.org>.
- [9] M. Ehmer e F. Khan. A comparative study of white box, black box and grey box testing techniques. *International Journal of Advanced Computer Science and Applications*, 3:12–15, Junho 2012.
- [10] The HDF Group. Introduction to hdf5, 2019. Disponível em www.hdfgroup.org.
- [11] R. C. J. Howland. On the stresses in the neighbourhood of a circular hole in a strip under tension. *Philosophical Transactions of the Royal Society of London*, 229:49–86, Janeiro 1930.
- [12] ECMA International. *Standard ECMA-404, The JSON Data Interchange Syntax*. ECMA International, Segunda edição, Dezembro 2017.
- [13] S. V. Kulkarni e S. A. Khaparde. *Transformer Engineering: Design, Technology, and Diagnostics*. CRC Press, Segunda edição, 2017.

-
- [14] W.D. Pilkey e D.F. Pilkey. *Peterson's Stress Concentration Factors*. John Wiley & Sons, Inc., Terceira edição, 2008.
- [15] J. N. Reddy. *An Introduction to Finite Element Method*. McGraw-Hill Book Company, Inc., Terceira edição, Janeiro 2005.
- [16] S. Timoshenko e J.N. Goodier. *Theory of Elasticity*. McGraw-Hill Book Company, Inc., Segunda edição, 1951.
- [17] S. Timoshenko e S. Woinowsky-Krieger. *Theory of plates and shells*. Engineering societies monographs. McGraw-Hill, 1959.
- [18] A. C. Ugural. *Plates and shells: Theory and analysis*. Quarta edição, Janeiro 2017.
- [19] A. Urdhwareshe. Object-Oriented Programming and its Concepts. *International Journal of Innovation and Scientific Research*, 26:1–6, Agosto 2016.
- [20] R.M.D. Vecchio, B. Poulin, P.T. Feghali, R. Ahuja, e D.M. Shah. *Transformer Design Principles*. CRC Press, 2017.
- [21] Électricité de France. Modelizations axis, d_plan, c_plan, 2012. Disponível em www.code-aster.org, U3.13.01.
- [22] Électricité de France. A simple example of use, 2012. Disponível em www.code-aster.org, U1.05.00.
- [23] Électricité de France. Procedure impr_resu (format “med”), 2013. Disponível em www.code-aster.org, U7.05.21.
- [24] Électricité de France. Modeling coque_3d, 2015. Disponível em www.code-aster.org, U3.12.03.
- [25] Électricité de France. Modelings 3d and 3d_si mechanics, 2016. Disponível em www.code-aster.org, U3.14.01.
- [26] Électricité de France. Operator post_champ, 2017. Disponível em www.code-aster.org, U4.81.05.
- [27] Électricité de France. Operator proj_champ, 2017. Disponível em www.code-aster.org, U4.72.05.
- [28] Électricité de France. Presentation of code_aster and salome-meca, 2017. Disponível em www.code-aster.org.
- [29] Électricité de France. Elements of plate: modellings dkt, dst, dktg and q4g, 2018. Disponível em www.code-aster.org, R3.07.03.
- [30] Électricité de France. Operator calc_champ, u4.81.04, 2018. Disponível em www.code-aster.org.

-
- [31] Électricité de France. Operators `affe_char_meca`, `affe_char_meca_c`, `affe_char_meca_f`, 2018. Disponível em www.code-aster.org, U4.44.01.
- [32] Électricité de France. Procedure `impr_resu` with the formats ‘result’ and ‘aster’, 2018. Disponível em www.code-aster.org, U4.91.01.
- [33] Électricité de France. Various syntaxes: files `.export`, 2018. Disponível em www.code-aster.org, D1.02.05.