

**Otimização de reforços em painéis planos com vigas  
*standard* utilizando métodos de otimização discreta para  
critérios de rigidez**

**José Miguel Candeias Santos Mareco**

Dissertação para obtenção do Grau de Mestre em

**Engenharia Aeroespacial**

Orientadores: Prof. José Arnaldo Pereira Leite Miranda Guedes  
Prof. José Firmino Aguilár Madeira

**Júri**

Presidente: Prof. Fernando José Parracho Lau  
Orientador: Prof. José Firmino Aguilár Madeira  
Vogal: Prof. João Orlando Marques Gameiro Folgado

**Julho 2016**



## **Agradecimentos**

Esta dissertação marca o final do meu percurso académico, e quero expressar a minha sincera e profunda gratidão a todos aqueles que me ajudaram e me deram apoio em toda a minha formação. Em especial, quero agradecer ao meu orientador, o Sr. Professor José Miranda Guedes, e ao meu co-orientador, o Sr. Professor José Aguilár Madeira, aos quais expresso a minha gratidão por toda a disponibilidade e apoio na conceção da minha tese de mestrado.

Por último, quero agradecer a Virgínia Mareco, Luísa Mouro, e a toda a minha família e amigos, por todo o apoio, motivação, carinho e auxílio que me deram desde sempre, mas sobretudo nesta etapa da minha vida académica.



## Resumo

A finalidade deste trabalho consiste no desenvolvimento de um modelo computacional que otimize o reforço de painéis planos com vigas *standard*, utilizando, para isso, métodos de otimização discreta, sujeito a diferentes condições de fronteira e carregamentos, de modo a aumentar a sua rigidez. Para tal, foi necessário desenvolver um modelo computacional que permitisse determinar a localização ótima desses reforços, bem como o tipo de vigas a utilizar, satisfazendo constrangimentos de volume/peso dos reforços. Antes de se conceber o programa, foi necessário definir a formulação do problema de otimização, nomeadamente a função objetivo (energia elástica de deformação total), as variáveis de projeto, bem como os constrangimentos. Utilizou-se o ANSYS, Inc., para a análise de elementos finitos, o GLODS (*global and local optimization using direct search*) como algoritmo de otimização e o MATLAB, The MathWorks, Inc., para criar uma interface entre os dois primeiros programas. Este modelo foi testado para vários conjuntos de vigas de reforço, e determinou as melhores soluções num vasto leque de casos de estudo, que surgiram de dois grupos de configurações principais (configurações em esquemas de 24 e 32 posições de vigas a introduzir). Os resultados concluíram que o programa de otimização discreta de 32 posições foi, em geral, mais eficiente.

**Palavras-chave:** Método de Elementos Finitos, Algoritmo de Otimização, Otimização de Topologia, Análise Estrutural, Teoria de Placas



## **Abstract**

The purpose of this work is to develop a computational model to optimize the reinforcement of plan panels with standard beams, using, for this, discrete optimization methods, subject to different loadings and boundary conditions, in order to increase its stiffness. For this, it was necessary to develop a computational model that allowed determining the optimal location of these reinforcements, as well as the type of beams to be used, satisfying constraints of volume/weight to the reinforcements. Before designing the program, it was necessary to define the formulation of the optimization problem, including the objective function (the total elastic strain energy), design variables, and constraints. It was used the ANSYS, Inc., for finite element analysis, the GLODS (global and local optimization using direct search) as optimization algorithm, and the MATLAB, The MathWorks, Inc., to create an interface between them. This model has been tested for several sets of reinforcement beams, and it determined the best solutions in a wide range of case studies, which arose from two main groups of configurations (configurations in schemes of 24 and 32 positions of beams to be introduced). The results concluded that the discrete optimization program of 32 positions was, in general, more effective.

**Keywords:** Finite Element Method, Optimization Algorithm, Topology Optimization, Structural Analysis, Plate Theory





# Conteúdo

Agradecimentos . . . . .	iii
Resumo . . . . .	v
Abstract . . . . .	vii
Lista de Tabelas . . . . .	xi
Lista de Figuras . . . . .	xiv
Lista de Símbolos . . . . .	xvi
Glossário . . . . .	xxi
<b>1 Introdução . . . . .</b>	<b>1</b>
1.1 Motivação . . . . .	1
1.2 <i>State-of-the-Art</i> . . . . .	2
1.3 Objetivos . . . . .	3
<b>2 Problema de Otimização . . . . .</b>	<b>4</b>
2.1 Otimização . . . . .	4
2.2 Otimização Estrutural . . . . .	5
2.2.1 Tipos de Problemas de Otimização Estrutural . . . . .	6
2.2.1.1 Otimização de Dimensão . . . . .	6
2.2.1.2 Otimização de Geometria . . . . .	6
2.2.1.3 Otimização de Topologia . . . . .	7
2.3 Etapas da Formulação do Problema de Otimização . . . . .	8
2.3.1 Objetivos do Problema de Otimização . . . . .	9
2.3.2 Dados e Informações do Problema . . . . .	9
2.3.2.1 Teoria de Placas . . . . .	11
2.3.2.2 Teoria de Placas de Kirchhoff-Love . . . . .	11
2.3.3 Identificação e Definição de Variáveis de Projeto . . . . .	15
2.3.4 Identificação da Função Objetivo . . . . .	17
2.3.5 Identificação dos Constrangimentos . . . . .	18
2.4 Formulação Sintetizada do Problema . . . . .	19
2.5 Algoritmo de Otimização . . . . .	20

<b>3</b>	<b>Modelo Computacional</b>	<b>22</b>
3.1	Conexão MATLAB-ANSYS	23
3.2	Cálculos e Dados Iniciais	23
3.3	Análise de Elementos Finitos	24
3.3.1	Pré-Processamento	24
3.3.1.1	Construção do Modelo do Painel	25
3.3.1.2	Malha	25
3.3.1.3	Seleção dos Tipos de Elemento	25
3.3.1.4	Definir Reforços para o Painel	26
3.3.2	Processamento	28
3.3.2.1	Implementação das Variáveis de Projeto	28
3.3.2.2	Condições de Fronteira	30
3.3.2.3	Cargas Aplicadas	30
3.3.2.4	Análise Estrutural	31
3.3.3	Pós-Processamento	31
3.3.3.1	Energia Elástica de Deformação	31
3.4	Algoritmos de Otimização Aplicados	32
3.4.1	<i>Pattern Search</i>	32
3.4.2	GLODS	33
<b>4</b>	<b>Programas Implementados</b>	<b>34</b>
4.1	Programa de Otimização Contínua	34
4.2	Programa de Otimização Discreta	36
4.2.1	Implementação do Programa de Otimização Discreta	38
4.2.2	Obtenção e Tratamento de Resultados	39
4.2.2.1	Esquemas Simétricos	40
4.2.2.2	Esquemas Normais ( ou Não Simétricos)	42
<b>5</b>	<b>Análise e Discussão de Resultados</b>	<b>43</b>
5.1	Resultados dos Esquemas com 24 Posições	43
5.2	Resultados dos Esquemas com 32 Posições	46
5.2.1	Relação Energia/Peso da Estrutura	48
5.3	Comparação de Resultados Entre os Esquemas de 24 e 32 Posições	50
<b>6</b>	<b>Conclusões e Trabalho Futuro</b>	<b>52</b>
6.1	Conclusões	52
6.2	Trabalho Futuro	53
	<b>Bibliografia</b>	<b>60</b>

# Lista de Tabelas

3.1	Propriedades do material utilizado. . . . .	25
5.1	Configurações das soluções dos esquemas de 24 vigas. . . . .	44
5.2	Configurações das soluções dos esquemas de 32 vigas. . . . .	47
5.3	Tabela de resultados da relação Energia/Peso. . . . .	49



# Lista de Figuras

2.1	Problema de otimização estrutural. Encontrar a melhor estrutura que transmita a carga $F$ ao apoio. . . . .	6
2.2	Otimização de dimensão. . . . .	7
2.3	Otimização de geometria. . . . .	7
2.4	Otimização de topologia. . . . .	8
2.5	Otimização de topologia bidimensional. . . . .	8
2.6	Painel/modelo base da estrutura. . . . .	9
2.7	Sistema de coordenadas e superfície média de uma placa. . . . .	12
2.8	Placa plana e com deformação sob as hipóteses de Kirchhoff-Love. . . . .	13
2.9	Forma das vigas/variáveis de projeto. . . . .	16
2.10	Fluxograma representativo de um algoritmo de otimização. . . . .	21
3.1	Esquema representativo da interação MATLAB-ANSYS . . . . .	22
3.2	Esquema do programa computacional . . . . .	24
3.3	Esquema representativo dos elementos: (1) <i>SHELL93</i> e (2) <i>BEAM189</i> . . . . .	26
3.4	Esquemas de 24 (à esquerda) e 32 variáveis (à direita) . . . . .	27
3.5	Esquemas de 12 (à esquerda) e 16 variáveis (à direita) . . . . .	27
3.6	Imagem de um pormenor da estrutura sem adição de <i>offsets</i> . . . . .	29
3.7	Esquema representativo do deslocamento vertical atribuído nas vigas (offset-z). . . . .	29
3.8	Esquema representativo do deslocamento vertical e horizontal atribuído às vigas do bordo do painel (offset-x e offset-z). . . . .	30
3.9	Condições de fronteira do painel. . . . .	30
3.10	Cargamentos aplicados no painel. . . . .	31
4.1	Duas soluções iguais, mas com variáveis invertidas relativamente ao eixo de simetria. . . . .	35
4.2	Dados do ficheiro <b>Constante.txt</b> correspondentes a 25% de reforço máximo aplicado no esquema de 24 posições com carga $F_1$ . . . . .	35
4.3	Diagrama dos esquemas a analisar . . . . .	36
4.4	Diagrama representativo das análises efetuadas em cada esquema . . . . .	37
4.5	Esquemas representativos da melhor avaliação. . . . .	41
4.6	Esquemas representativos da pior avaliação. . . . .	41

4.7	Soluções representativas de esquemas normais (não simétricos). . . . .	42
5.1	Esquema simétrico e esquema normal de 24 posições com cargas representadas. . . . .	43
5.2	Esquema simétrico e esquema normal de 32 posições com cargas representadas. . . . .	46
5.3	Soluções com mesma função objetivo mas com configurações invertidas relativamente ao eixo de simetria do painel. . . . .	51

# Lista de Símbolos

## Lista de Símbolos

**C** – compliance

**d** – vetor dos deslocamentos nodais do elemento; vetor dos graus de liberdade.

**E** – módulo de Young; módulo elasticidade.

**f** – função objetivo; energia elástica de deformação da estrutura, U.

**F** – vetor de força aplicada.

**G** – módulo de cisalhamento; módulo de rigidez; módulo de torção.

**g<sub>k</sub>** – constrangimentos de desigualdade.

**h<sub>j</sub>** – constrangimentos de igualdade.

**I** – matriz identidade.

**m** – metro(s).

**M** – momentum.

**N** – newton; unidade de força do Sistema Internacional de Unidades; kg.m/s<sup>2</sup>.

**Pa** – pascal(s) (unidade de pressão).

**P<sub>z</sub>** – carga aplicada na direcção do eixo dos z.

**s** – segundo.

**S<sub>e</sub>** – superfície do elemento.

**tr(σ)** – traço do tensor de tensões.

**u** – vetor campo deslocamentos da estrutura.

**U** – energia elástica de deformação total da estrutura.

**ν** – coeficiente de Poisson; razão de Poisson.

**V** – volume da estrutura.

**x** – coordenada cartesiana.

**y** – coordenada cartesiana

**z** – coordenada cartesiana

**γ<sub>ij</sub>** – deformação de corte; deformação de cisalhamento.

**δ<sub>ij</sub>** – delta de Kronecker.

**Δ** – operador diferencial.

**ε** – tensor extensão; tensor de deformações; tensor de deformação Green-Lagrange.

**λ** – primeiro parâmetro de Lamé.

**μ** – segundo parâmetro de Lamé.

**σ** – tensor tensão; tensor tensão de Cauchy.

## Lista de Índices

**e** – elemento finito.

**T** – definições de transposição de uma matriz (equação 2.25).

## Lista de Abreviaturas

**ANSYS** – ANalysis SYStem (inglês); sistema de análise.

**APDL** – ANSYS Parametric Design Language (inglês); linguagem de projeto paramétrico ANSYS.

**DOS** – disk operating system (inglês); sistema operacional de disco.

**e.g.** – exempli gratia (latim); por exemplo.

**ETABLE** – elemento table (inglês), ANSYS, Inc.; tabela de elemento.

**et al.** – et alia (latim); e outros.

**GLODS**: Global and Local Optimization using Direct Search (inglês); otimização global e local utilizando pesquisa direta.

**IBM Corp.** – International Business Machines Corporation (inglês); companhia das máquinas de negócio internacionais.

**Inc.** – incorporated (inglês); incorporado/associado.

**MATLAB** – MATrix LABoratory (inglês); laboratório de matriz.

**MEF** – método de elementos finitos.

**SENE** – strain energy (inglês), ANSYS, Inc.; energia de deformação.

**TOC** – table of contents (inglês); tabela de conteúdos; índice remissivo.

**.m** – a publicação de um ficheiro de código de MATLAB (.m) cria um documento formatado que inclui o código pessoal, comentários e saída.

**.txt** – É uma extensão para arquivos de texto que geralmente contêm muito pouca formatação (e.g., sem negrito ou itálico).

**\*USE** – executa um macroficheiro, ANSYS, Inc.



# Glossário

**Anisotrópico** – Com propriedades físicas desiguais ao longo de diferentes eixos [1].

**Batch mode (ou batch processing)** – Um sistema pelo qual os programas de computador de um número de usuários individuais são submetidos a um computador de um só lote. Uma técnica que usa um único carregamento de programa para processar muitos empregos, tarefas ou solicitações de serviço individuais [2].

**Cisalhamento** – Deformação de um corpo com deslocamento em planos diferentes, mantendo volume constante [3].

**Coefficiente de Poisson ( $\nu$ ) (ou razão de Poisson)** – Relação entre deformação elástica transversal e deformação axial (longitudinal) [4]. Mede a deformação transversal (em relação à direção longitudinal de aplicação da carga) de um material homogêneo e isotrópico. A relação estabelecida é entre deformações ortogonais [5].

**Compliance (C) (ou complacência)** – A deformação de um corpo elástico expressa como uma função da força de a produzir. Um coeficiente que exprime a capacidade de resposta de um sistema mecânico a uma força [6]. O inverso de rigidez [7].

**Constantes elásticas** – Em ciência dos materiais, são os números que quantificam a resposta de um determinado material à deformação elástica ou inelástica quando uma carga é aplicada a esse material. São o módulo de Young (E), o módulo de volume (K), o módulo de cisalhamento (G) e o coeficiente de Poisson ( $\nu$ ) [8].

**Deflexão ( $\omega$ )** – Em engenharia, a deflexão é o grau em que um elemento estrutural é deslocado sob a ação de uma carga. Pode referir-se a um ângulo ou a uma distância [9].

**Deformação** – A extensão na qual um corpo é distorcido quando submetido a uma força de deformação, como quando sob tensão. A distorção pode envolver uma mudança tanto na forma como no tamanho. Todas as medidas de deformação são adimensionais (não têm nenhuma unidade de medida) [10]. A mudança relativa na posição de pontos no interior de um corpo que tenha sido submetido a tensão. O exemplo clássico em duas dimensões é o do quadrado que é deformado para um paralelepípedo [11].

**Deformação de corte ( $\gamma_{ij}$ )** – A verdadeira deformação de corte é definida como a alteração no ângulo (em radianos) entre dois elementos de linha de material inicialmente perpendiculares um ao outro na configuração não deformada ou inicial. A deformação de corte de engenharia é a tangente desse ângulo, e é igual ao comprimento da deformação no seu máximo dividido pelo comprimento perpendicular no plano de aplicação da força que, por vezes, faz com que seja mais fácil de calcular [12].

**Delta de Kronecker ( $\delta_{ij}$ )** – É uma função descontínua de duas variáveis, geralmente apenas números inteiros positivos. A função é 1 se as variáveis são iguais, e zero, caso contrário [13].

**Disco** – Uma superfície que é plana e redonda.

**Elasticidade linear** – É o estudo matemático de como objetos sólidos se deformam e se tornam

internamente stressados devido a condições de carga prescritas. É uma simplificação da teoria não-linear mais geral da elasticidade e é um ramo da mecânica do contínuo. Os pressupostos fundamentais são: deformações infinitesimais ou "pequenas" e relações lineares entre os componentes de tensão e deformação. Além disso, a elasticidade linear é válida apenas para os estados de tensão que não produzem rendimento. Estes pressupostos são razoáveis para muitos materiais de engenharia [14].

**Elástico** – Deformações devido a cargas externas são completamente, e de forma instantânea, reversíveis após a remoção da carga [8]. Diz-se que um material é elástico quando a energia dissipada após um ciclo de carregamento-descarregamento é nula e, assim, as deformações envolvidas no processo são totalmente reversíveis. É importante ressaltar que um material elástico pode apresentar uma curva tensão-deformação linear ou não linear [15].

**Elemento** – É o bloco de construção básico da análise de elementos finitos. Existem vários tipos básicos de elementos. O tipo de elemento a analisar depende do tipo de objeto a ser modelado na análise por elementos finitos e o tipo de análise que irá ser realizado. É uma relação matemática que define como os graus de liberdade de um nó se referem ao seguinte. Os elementos podem ser linhas (treliças ou vigas), áreas (placas e membranas 2D ou 3D) ou sólidos (tijolos ou tetraedros). O elemento também relata como os desvios criam tensões [16].

**Energia elástica** – A energia armazenada no material elástico deformado (por exemplo, uma mola de relógio). É, por vezes, chamada de energia potencial elástica, porque ela pode ser recuperada quando o objeto retorna à sua forma original [17].

**Estática** – O ramo da mecânica que lida com corpos em repouso ou forças em equilíbrio [18].

**Graus de liberdade** – Os graus de liberdade para determinado ponto (ou nó) representam o possível movimento desse ponto devido à carga da estrutura. Os graus de liberdade, que também representam as forças e os momentos, são transferidos a partir de um elemento para outro. O grau de liberdade de um nó (que é baseado no tipo de elemento) também refere que tipos de forças e restrições são transmitidos através do nó para o elemento. Uma força (axial ou de cisalhamento) é equivalente a um grau de liberdade de translação. Um momento é equivalente a um grau de liberdade rotacional [16].

**Heurística** – Em ciência da computação, é uma técnica concebida para resolver um problema mais rapidamente quando os métodos clássicos são muito lentos, ou para encontrar uma solução aproximada quando os mesmos não conseguem encontrar uma solução exata. A função heurística é uma função que escolhe alternativas em algoritmos de busca em cada etapa de uma ramificação com base em informações disponíveis para decidir qual ramo seguir, assim, podendo aproximar-se da solução exata [19].

**Isotrópico** – De propriedades físicas iguais ao longo de todos os eixos [20].

**Lei de Hooke** – Expressa a força necessária para expandir ou comprimir uma mola [21]. É um princípio da Física que afirma que a força  $F$  necessária para estender ou comprimir uma mola por alguma distância  $X$  é proporcional à distância. Isto é:  $F = kX$ , em que  $k$  é um fator constante característico da mola (isto é, a rigidez), e  $X$  é pequena em comparação com a deformação total possível da mola. É uma aproximação precisa para a maioria dos corpos sólidos, desde que as forças e as deformações sejam suficientemente reduzidas. Permite deduzir a relação entre a deformação e a carga aplicada

para objetos complexos em termos de propriedades intrínsecas dos materiais de que são feitos. Por exemplo, pode-se deduzir que uma barra homogênea com secção transversal constante irá comportar-se como uma mola simples quando esticada, com uma rigidez  $k$  diretamente proporcional à sua área de secção transversal e inversamente proporcional ao seu comprimento [22].

**Malha** – A malha é uma rede de elementos de linha e nós de interligação usada para modelar um sistema estrutural e resolvê-lo numericamente pelo seu comportamento simulado sob uma carga aplicada. Em primeiro lugar, técnicas computacionais criam um modelo analítico povoando o domínio do material com uma malha de elementos finitos em que a cada elemento de linha é atribuído atributos matemáticos (axial, flexão, cisalhamento e rigidez de torção, etc.) que simulam as propriedades geométricas do material e o sistema estrutural. O sistema é então restrito a condições de fronteira e sujeito a esforços mecânicos ou térmicos. A solução numérica pode então resolver tensões, deformações e deslocamentos da estrutura [23].

**Matriz identidade (I)** – É uma matriz diagonal (matriz cujos elementos exteriores à diagonal principal são nulos) [24], cujos elementos da diagonal são todos iguais a 1 [25].

**Método das direções admissíveis** – É uma importante classe de algoritmos para resolver problemas de otimização não-linear restrita [26]. Incorpora a mesma filosofia que as técnicas de minimização irrestrita, mas é projetado para lidar com restrições de desigualdade. Resumidamente, a ideia é escolher um ponto de partida que satisfaça as restrições e encontrar uma direção tal que (i) um pequeno movimento nessa direção permanece admissível e (ii) há otimização da função objetivo. Nesse caso, move-se uma distância finita em determinada direção, obtendo-se um ponto novo e otimizado. O processo é repetido até que nenhuma direção satisfazendo ambos (i) e (ii) possa ser encontrada. Em geral, o ponto final é um mínimo local (mas não necessariamente global) constringido do problema. Uma direção satisfazendo ambos (i) e (ii) é chamada uma direção admissível utilizável [27]. Há muitas maneiras de escolher essas direções, portanto, muitas variações deste método, originalmente descrito por Zoutendijk [28].

**Método de elementos finitos** – É uma técnica matemática utilizada para prever a resposta de estruturas e materiais a factores ambientais. É uma poderosa ferramenta de engenharia para simular numericamente o mundo real sem necessidade de testar protótipos em laboratório. O processo começa com a criação de um modelo geométrico. Em seguida, o modelo é subdividido em pedaços pequenos (elementos) de formas/formatos simples ligados em pontos de nó específicos. Dentro de cada elemento, a variação do deslocamento é assumida para ser determinada por funções de forma polinomial simples e deslocamentos nodais ( $d$ ). Equações para as deformações e tensões são desenvolvidas em termos dos deslocamentos nodais desconhecidos. A partir daí, as equações de equilíbrio são montadas numa matriz ( $N$ ) que pode ser facilmente programada e resolvida num computador. Depois de aplicar as condições de fronteira adequadas, os deslocamentos nodais são encontrados através da resolução da matriz da equação de rigidez. Uma vez conhecidos os deslocamentos nodais, as tensões e as deformações dos elementos podem ser calculadas [29].

**Método de homogeneização** – A ideia principal é substituir o problema difícil da disposição da distribuição, ou geometria, do material por um problema de dimensionamento, ou topologia variável,

muito mais fácil para as propriedades de densidade e eficácia de um material compósito perfurado, obtido perfurando pequenos orifícios no material homogêneo original [30]. Muitos, ou infinitos, orifícios são introduzidos para formar um meio poroso que produza uma estrutura linear elástica. Um problema de otimização para a disposição da geometria é definido de modo a alcançar a porosidade ótima do meio identificado com um domínio de design. Se uma porção do meio poroso consiste em apenas vazios, a estrutura não é colocada nessa localização. Por outro lado, se nenhuma porosidade for detectada noutro local, uma estrutura “sólida” é aí colocada. Se a porosidade não for os valores-limite, um meio poroso é gerado. Neste caso, o material “sólido”, ou estrutura, é distribuído otimamente numa região especificada de modo a que uma determinada função objetivo seja minimizada sob um conjunto de restrições. Matematicamente, este problema de distribuição ótima de orifícios em microescala (ou material “sólido”) pode ser definido como minimizar a função objetivo de tal modo que (i) a conformidade média se submeta às equações de equilíbrio, ao volume de vazios (ou orifícios) e aos constrangimentos de tensão/deslocamento, ou (ii) o volume de vazios (ou orifícios) se submeta às equações de equilíbrio, à conformidade média e aos constrangimentos de tensão/deslocamento [31].

**Módulo de cisalhamento (G) (ou módulo de rigidez/torção)** – A razão entre a tensão de cisalhamento ( $F$ ) aplicada ao corpo e a sua deformação específica [32]. É o coeficiente de elasticidade para uma força de cisalhamento ( $F$ ). É a razão da tensão de cisalhamento ( $F$ ) até ao deslocamento por unidade de comprimento da amostra (deformação de corte/cisalhamento) [33].

**Modelo de estrutura pseudocontínuo** – Uma técnica de projeção ótima de um esquema de reforço de placas em que uma conformidade/compliance mínima é desenvolvida. Uma placa fina com reforços é tratada como uma placa de estrutura anisotrópica pseudocontínua, e é discretizada em elementos finitos. O design de conformidade mínimo submetido a um volume constante, em que as distribuições de ângulos da disposição do reforço e as densidades do reforço são variados, é determinado pela técnica de programação quadrática recursiva. Ao aplicar esta técnica de design, é obtido o esquema de reforço ótimo da placa retangular submetida a determinada carga típica e condições de suporte [34].

**Módulo de compressibilidade (K) (ou módulo volumétrico)** – É um parâmetro que descreve a elasticidade volumétrica, ou seja, a tendência de um material em se deformar em todas as direções quando uniformemente carregado em todas elas (hidrostaticamente). É definido como a razão entre a tensão volumétrica e a deformação volumétrica, e é o inverso da compressibilidade [35].

**Módulo de Young (E) (ou módulo elástico)** – É uma propriedade mecânica de materiais sólidos elásticos lineares. Define a relação entre a tensão (ou carga aplicada) (força por unidade de área) e a distorção (deformação proporcional) num material [36]. É um número que mede a resistência de um objeto que é deformado elasticamente (isto é, não permanentemente), quando uma força lhe é aplicada [37].

**Nó** – É uma localização de coordenadas no espaço onde os graus de liberdade são definidos. Os resultados de uma análise de elementos finitos (deformações e tensões) são normalmente dados nos nós [16].

**Pattern search** – Padrão de pesquisa de otimização. É uma família de métodos numéricos de otimização que não requer que o gradiente do problema seja otimizado. Daí, pode ser utilizado em

funções que não sejam contínuas ou diferenciáveis. Tais métodos de otimização são também conhecidos como métodos de procura direta, livres de derivadas, ou métodos de caixa-preta [38].

**Placa** – Uma peça plana de material [39].

**Primeiro parâmetro de Lamé ( $\lambda$ )** – É um módulo elástico [40].

**Segundo parâmetro de Lamé ( $\mu$ )** – É equivalente ao módulo de cisalhamento (G) [40].

**Tensão** – A ação sobre um corpo de qualquer sistema de forças equilibradas através da qual resulta tensão ou deformação. Uma carga, força, ou sistema de forças que produzem uma tensão. A resistência interna ou de reação de um corpo elástico às forças externas aplicadas sobre ele. A força por unidade de área [41].

**Tensor deformação ( $\varepsilon$ )** – É um tensor simétrico usado em mecânica de meios contínuos e mecânica de sólidos deformáveis para caracterizar a alteração de forma e volume de um corpo. Em três dimensões é tensor de ordem dois [42]. Cada uma das componentes do tensor é uma função cujo domínio é o conjunto de pontos do corpo cuja deformação se pretende caracterizar. O tensor de deformações está relacionado com o tensor de tensões ( $\sigma$ ) mediante as equações de Hooke generalizadas [43].

**Tensor tensão ( $\sigma$ )** – Se se tomar um cubo de material e submetê-lo a uma carga arbitrária, pode-se medir a tensão (ou stress) sobre ele em várias direções. Estas medidas irão formar um tensor de segunda ordem, que é o tensor tensão [11]. É um tensor de segunda ordem, com nove componentes,  $\sigma_{ij}$ , que define completamente o estado de tensão, ou stress, num ponto no domínio de um material na sua configuração deformada. O tensor relaciona um vetor diretor de unidade de comprimento,  $n$ , com o vetor tensão,  $T^n$ , sobre uma superfície imaginária perpendicular a  $n$ :  $T^n = n * \sigma$  ou  $T_j^n = \sigma_{ij} * n_i$  [44].

**Trabalho** – Produção de um efeito através da aplicação de uma força. É necessária energia para realizar trabalho, pelo que o trabalho é medido em joules [45].

**Traço** – Em álgebra linear, o traço de uma matriz quadrada é definido como sendo a soma dos elementos da diagonal principal (a diagonal a partir do canto superior esquerdo em direção ao canto inferior direito) [46].

**Treliça** – Armação formada pelo cruzamento de elementos retos (de madeira, aço, alumínio, etc.) cujas extremidades são ligadas em pontos conhecidos como nós. As treliças são dimensionadas de modo que as únicas forças atuantes nos elementos sejam de tração ou compressão. As treliças podem ser planas ou tridimensionais [47].



# Capítulo 1

## Introdução

A finalidade deste trabalho consiste no desenvolvimento de um modelo computacional que otimize o reforço de painéis planos com vigas *standard*, utilizando, para isso, métodos de otimização discreta.

Pretende-se reforçar um painel plano com vigas de dimensões standardizadas, sujeito a diferentes condições de fronteira e carregamentos, de modo a aumentar a sua rigidez. Para tal, é necessário desenvolver um modelo computacional que permita determinar a localização ótima desses reforços, bem como o tipo de vigas a utilizar, satisfazendo constrangimentos de volume/peso do reforço.

### 1.1 Motivação

O processo de conceção e fabricação de sistemas de reforços em estruturas tem sido desenvolvido ao longo dos séculos. No fabrico complexo de grande parte de estruturas metálicas de veículos, tanto nas indústrias aeroespacial e naval quanto na automóvel, o reforço de placas é uma constante necessidade. A utilização destes reforços em placas tem como finalidade aumentar a sua rigidez e, com isto, obter uma melhoria das suas características mecânicas. Com esta técnica, consegue-se uma redução significativa do material utilizado bem como uma diminuição na relação peso/potência de cada veículo em causa.

Atualmente, os computadores podem executar cálculos bastante complexos e processar rapidamente grandes quantidades de dados. Os projetos de engenharia e otimização beneficiam muito com esta revolução, pois eles exigem um grande número de cálculos e podem fazê-los num curto espaço de tempo. Isto é altamente desejável porque melhores sistemas projetados têm maior capacidade, menor custo e são mais fáceis de manter e executar. As técnicas de otimização são bastante gerais, tendo uma vasta gama de aplicabilidade em diversos campos, sendo apenas limitadas pela imaginação ou criatividade dos engenheiros (Arora [48]).

O processo de criação do *design* desejado, que depende da experiência e habilidade dos engenheiros, pode por vezes conduzir a resultados incorrectos na síntese de sistemas complexos. Com o intuito de aperfeiçoar esta fase crítica do projeto, vão-se desenvolvendo sistemas de cálculos compu-

tacionais que otimizam a topologia de reforços em placas, através de métodos numéricos e analíticos para análises estruturais.

Criar padrões de reforços para estruturas constituídas por placas e cascas é um problema particularmente interessante em desenho estrutural, porque a rigidez, resistência e vibração dessas estruturas reforçadas são dependentes do padrão de disposição do reforço. No entanto, é também um problema difícil de criar, especialmente quando o número de reforços é grande e o padrão de disposição é complicado, aumentando a complexidade quando as estruturas são submetidas a cargas de flexão e torção (Arora [48], Ding e Yamakaki [49]).

## 1.2 *State-of-the-Art*

Nas últimas décadas, o problema de otimização de reforços em placas tem sido investigado intensivamente por meio de várias abordagens.

Em 1981 e 1982, Cheng e Olhoff [50, 51] consideraram o problema de maximizar a rigidez global de placas elásticas descritas pela teoria de placas finas, assumindo a espessura da placa alterável para um volume fixo de material. Usaram a espessura da placa como a variável do projeto tendo em conta valores máximos e mínimos admissíveis para o modelo estudado. Os resultados foram comparados com resultados anteriormente obtidos a partir de formulações de *design* e comprovaram a robustez e melhor desempenho da nova formulação.

No ano de 1988, Yamazaki e Kobayashi [52] desenvolveram o modelo de estrutura com base em variáveis contínuas e obtiveram uma configuração de reforço ideal para duas direções através da análise de elementos finitos e programação matemática. No entanto, ainda em 1988, surgiu um método mais divulgado, desenvolvido por Bendsoe e Kikuchi [53]. Este método está relacionado com técnicas de produção que consistem em calcular a distribuição otimizada no espaço de um material anisotrópico que é construído a partir da introdução de pequenos orifícios distribuídos periodicamente num dado material homogêneo, isotrópico, com o requisito de que a estrutura resultante satisfaça os requisitos de projeto.

Em 1997, com o objetivo de aumentar a rigidez da estrutura do tipo placa, Chung e Lee [54] utilizaram a técnica de otimização de topologia para encontrar a forma e a localização ótimas dos reforços. Para obter esta otimização, usaram a densidade do material como variável do projeto através do método das direções admissíveis.

Com base no método de homogeneização, em 1998, Luo e Gea [55] introduziram um método de criação de um padrão de disposição para o reforço ideal, considerando quer a localização do reforço, quer a orientação, para melhor lidar com a estática e com problemas de valores próprios.



Em 1999, Krog e Olhoff [56] estudaram a otimização de topológica em estruturas do tipo placa circular, carregadas estaticamente, ou em vibração livre, tendo como objetivo a otimização da rigidez e dos seus modos de vibração para diferentes tipos de carregamento.

Lam e Santhikumar [57] propuseram, em 2003, um critério para a determinação automática do local ótimo a serem colocados os reforços, em que simultaneamente são sujeitos a um determinado conjunto de restrições impostas pelo projeto.

Em 2004, Ansola et al. [58] apresentaram um método combinado de otimização de reforço que consiste em otimizar simultaneamente a disposição dos reforços e a geometria dos mesmos.

Ding e Yamazaki [49] introduziram, em 2005, um método eficaz para a disposição e criação de reforços em placas e cascas. Este método é chamado de técnica de crescimento e adaptação, e consiste no crescimento de reforços em forma de ramificação automática que melhoram o seu desempenho mecânico.

### **1.3 Objetivos**

Como já foi referido, o principal objetivo deste trabalho é desenvolver um modelo computacional que otimize o reforço de um painel plano sujeito a diferentes carregamentos, com diferentes condições de fronteira, de modo a maximizar a rigidez dessa estrutura, ou seja, o conjunto painel-vigas.

Para este objetivo ser alcançado com sucesso, é necessário encontrar a formulação mais conveniente para este estudo. Utilizando o *software* de elementos finitos (ANSYS, Inc.), é construído um modelo computacional para simular a análise estrutural. Através de um *software* iterativo (MATLAB, The MathWorks, Inc.), cria-se um modelo computacional para efetuar a otimização. Por fim, é necessário criar uma interface entre a parte estrutural e o cálculo numérico de forma a obter o controlo e a otimização desejados.

## Capítulo 2

# Problema de Otimização

Este capítulo tem a finalidade de descrever o conceito de otimização e os processos estruturais especificamente focados neste estudo. São também apresentados os tipos de otimização estrutural e como é criada a formulação de um problema de otimização, salientando todos os procedimentos para a sua implementação.

### 2.1 Otimização

Otimização é um conceito peculiar à humanidade, que, por instinto e a qualquer instante, toma decisões estratégicas de forma a tirar o maior proveito de recursos disponíveis, sem comprometer a eficácia da função exercida. Nos seus termos mais básicos, otimização é uma disciplina matemática que diz respeito à descoberta de extremos (mínimos e máximos) de números, funções ou sistemas (Kiranyaz et al. [59]).

Os métodos de otimização podem classificar-se de acordo com o procedimento usado para solucionar o problema: (i) programação estocástica vs. determinística, (ii) programação de multiobjetivos, (iii) *network optimization*, (iv) programação linear vs. não linear, (v) métodos heurísticos e metaheurísticos [60, 61].

Enquanto que nos métodos determinísticos a introdução do *input* leva sempre aos mesmos resultados de *output*, a otimização estocástica nem sempre assegura que o ótimo global seja alcançado. Mas, a aplicação repetitiva de uma simulação estocástica, especialmente em problemas complexos, é mais provável finalmente alcançar o ótimo global do que a simples aplicação de um método determinístico [60].

Na otimização determinística, é assumido que os dados para um problema são conhecidos com exatidão. No entanto, para muitos problemas reais, os dados podem não ser conhecidos. Em otimização sob incerteza, ou otimização estocástica, a incerteza é incorporada no modelo. Modelos de programação estocásticos tiram vantagem do fato das distribuições de probabilidade que regem os dados serem conhecidas ou poderem ser estimadas; o objetivo é encontrar alguma diretiva que seja viável para todos os casos de dados possíveis e otimizar o desempenho esperado do modelo [62].

Os métodos heurísticos examinam soluções potenciais procurando pela solução ótima global, testando e avaliando soluções ao longo do percurso. A forma mais simples de tal procedimento é testar e avaliar todas as combinações possíveis [60].

A resolução de problemas de otimização global é uma tarefa desafiadora, com dificuldades adicionais quando derivadas não estão disponíveis para uso. No entanto, há um grande número de aplicações práticas no mundo real onde uma otimização global livre de derivadas é necessária [63].

O trabalho de Custódio e Madeira [63], foi a primeira tentativa de otimização generalizada por *pattern search* (ou *direct search*): GLODS ou *global and local optimization using direct search* (otimização global e local utilizando pesquisa direta). A *direct search* é uma família de métodos numéricos de otimização que não necessita do cálculo de derivadas. Daí, pode ser utilizada em funções que não sejam contínuas ou diferenciáveis [64].

O GLODS foi projetado para calcular todos os mínimos locais de um problema, a partir do qual o mínimo global é facilmente identificado. Com esta finalidade, os autores propuseram uma nova classe de algoritmos, com base na *direct search* direcional acoplada a estratégias *multistart* (ou multiarranque), de modo a resolver problemas de otimização global livre de derivadas. Em otimização livre de derivadas, a convergência global de um algoritmo significa garantir a convergência para um ponto estacionário, independentemente da aproximação inicial prevista para o otimizador. A ideia-chave é explorar toda a região admissível iniciando as pesquisas locais de diferentes pontos possíveis, mas evitando cálculos desnecessários, fundindo as pesquisas locais que estão suficientemente próximas. Esta implementação numérica apresenta a característica adicional de permitir calcular não apenas o mínimo global, mas os diferentes mínimos locais. Existem várias maneiras pelas quais a eficiência numérica da implementação considerada poderia ser melhorada. Em particular, a estrutura geral de *direct search* direcional e estratégias *multistart* sugerem fortemente a paralelização do algoritmo [63].

Como num método de *direct search* clássico do tipo direcional, a estrutura algorítmica do GLODS é organizada em torno de uma pesquisa e uma escolha por passos. O objetivo principal da fase de pesquisa é explorar toda a região admissível, numa tentativa de localizar subdomínios promissores, que passarão então a ser localmente explorados pela etapa de escolha do algoritmo. O *search step* é responsável por assegurar a convergência do método, mas a qualidade dos mínimos calculados, como correspondendo a valores mínimos locais ou globais, dependerá do *poll setp* (exploração e regiões previamente localizadas) [65]. Este método vai ser utilizado como algoritmo de otimização neste projeto, e é novamente tratado mais à frente na secção 3.4.2.

## 2.2 Otimização Estrutural

Segundo James Edward Gordon, em mecânica, uma estrutura é definida como "qualquer conjunto de materiais que se destina a sustentar cargas", e otimização significa "tornar as coisas melhores". Assim, otimização estrutural é um processo aprimorado de cálculo para alcançar a melhor maneira de sustentação de cargas por um conjunto de materiais. A figura seguinte (Fonte: [62]) exemplifica a situação em que uma carga deve ser transmitida a um suporte, partindo de uma região no espaço.

[66]

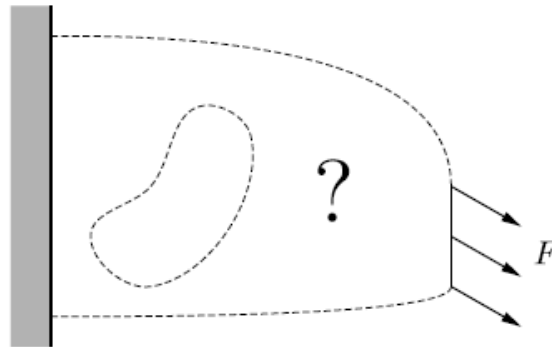


Figura 2.1: Problema de otimização estrutural. Encontrar a melhor estrutura que transmita a carga  $F$  ao apoio.

A otimização de estruturas está totalmente relacionada com a melhoria das suas características estruturais e mecânicas, minimizando o consumo de material e os custos finais de projeto. Normalmente, leva a uma apreciável economia relativamente ao projeto inicialmente considerado.

Obter uma otimização com sucesso requer uma correta formulação do problema, analisando as várias possibilidades de combinação dos parâmetros de projeto, a melhor eficácia no tempo de execução e uma correta projeção do modelo matemático representativo desse problema.

## 2.2.1 Tipos de Problemas de Otimização Estrutural

Os problemas de otimização estrutural são classificados dependendo da sua particularidade geométrica. Podem ser classificados em três classes: otimização de dimensão, otimização de geometria e otimização topológica. Seguidamente, é apresentado de forma sucinta as diferentes classes de otimização para melhor compreensão dos tipos de problemas habituais.

### 2.2.1.1 Otimização de Dimensão

A otimização de dimensão é utilizada quando a variável de projeto é algum tipo de espessura estrutural, isto é, pode ser, por exemplo, áreas de corte transversal dos membros de uma treliça ou a distribuição de espessura de uma chapa. Esta espessura ideal de um certo material é determinada com base numa média de desempenho da estrutura e dependerá dos carregamentos considerados. Em otimização, este método é geralmente utilizado após a otimização geométrica, uma vez que a geometria inicial do componente é previamente definida e interpretada. A figura 2.2 ilustra uma otimização de dimensão de uma treliça (Fonte: [62]).

### 2.2.1.2 Otimização de Geometria

Na otimização de geometria, as variáveis de projeto representam a forma ou o contorno de uma parte do domínio estrutural. As ligações da estrutura não são alteradas pelo sistema, isto é, a topologia da estrutura mantém-se constante, enquanto que o que se altera neste processo são as fronteiras do

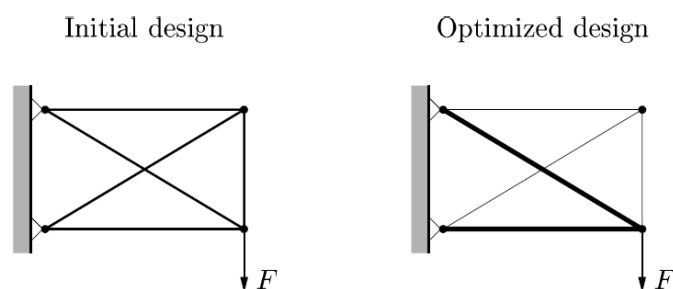


Figura 2.2: Otimização de dimensão.

domínio (forma e contornos da estrutura), que vão variando ao longo do decurso da otimização, como se pode verificar na figura seguinte (Fonte: [62]):

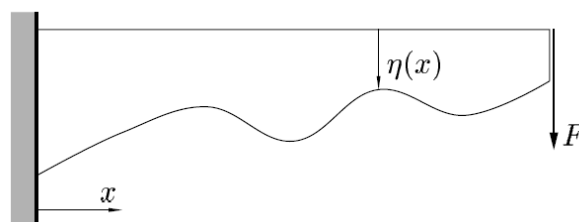


Figura 2.3: Otimização de geométrica.

Na figura, encontrar a função  $\eta(x)$ , descreve de forma semelhante a estrutura de viga a ser construída. Usualmente, neste tipo de problema, as variáveis de projeto são contínuas. Neste tipo de otimização, as dimensões são geradas em simultâneo com os contornos da estrutura. A otimização de geometria é geralmente um problema com maior dificuldade de implementação relativamente à otimização de dimensão.

### 2.2.1.3 Otimização de Topologia

A otimização de topologia é a otimização mais abrangente em projetos estruturais.

Neste método, as variáveis de projeto são parâmetros numéricos que podem alterar a distribuição de material ao longo da estrutura com o objectivo de economizar material em regiões com reduzida solicitação do mesmo. Existem dois tipos de variáveis de projeto, contínuas ou discretas.

Num caso de uma treliça, em que esta utilize as áreas de secção transversal das barras como variáveis de projeto discretas, pode-se permitir que estas possam ir a valor zero, o que faz com que seja possível remover da treliça as barras que não efetuam qualquer esforço (exemplo na figura seguinte (Fonte: [62])).

Se se utilizar otimização contínua, como por exemplo numa chapa bidimensional, as mudanças ideais de topologia podem ser feitas fazendo com que a espessura mínima da chapa possa ter valor zero e a espessura máxima seja um valor de interesse para o projeto. No caso tridimensional, o mesmo efeito pode ser alcançado ao utilizar a densidade como variável de projeto, onde só pode assumir valores entre 0 e 1. Um exemplo de otimização contínua bidimensional é apresentado na figura seguinte. A caixa

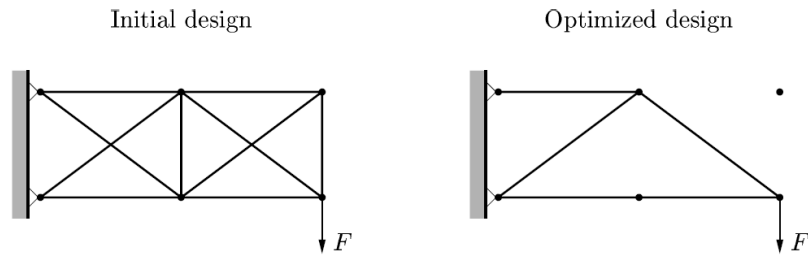


Figura 2.4: Otimização de topologia.

deverá ser preenchida com 50% do material e com as condições de fronteira exibidas na parte superior da figura. A distribuição ideal de material para o melhor desempenho das restrições dadas é, por fim, apresentada na parte inferior figura (Fonte: [62]).

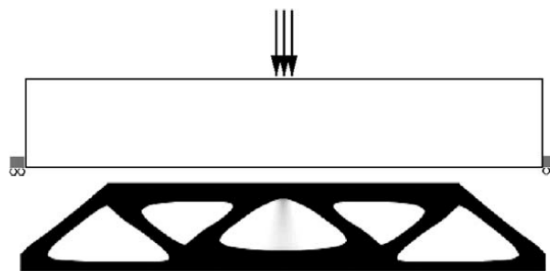


Figura 2.5: Otimização de topologia bidimensional.

Nesta tese, numa primeira abordagem ao problema, é feita uma breve análise da estrutura com variáveis de projeto do tipo contínuo para otimização topológica. Em seguida, com maior ênfase e com uma análise mais detalhada do problema, é utilizada otimização topológica, com variáveis de projeto do tipo discreto.

## 2.3 Etapas da Formulação do Problema de Otimização

A formulação de um projeto de otimização tem como objetivo a tradução da descrição do problema numa instrução matemática bem definida. Na generalidade dos problemas, é utilizado um procedimento de tarefas a serem executadas em cada uma das seguintes etapas [48]:

- 1ª Etapa- Objetivos do problema de otimização;
- 2ª Etapa- Dados e informações do problema;
- 3ª Etapa- Identificação e definição de variáveis de projeto;
- 4ª Etapa- Identificação da função objetivo;
- 5ª Etapa- Identificação dos constrangimentos.

### 2.3.1 Objetivos do Problema de Otimização

Na primeira etapa da formulação do problema, pretende-se desenvolver uma descrição do projeto em causa bem como uma avaliação teórica, que visa em anotar os objetivos gerais e as exigências a serem cumpridas [48].

Neste trabalho, como referido anteriormente, é pretendido reforçar um painel plano com vigas, sujeito a diferentes carregamentos e condições de fronteira, de modo a aumentar a sua rigidez. Para maximizar esta rigidez é utilizado um modelo computacional que permite determinar a localização ótima desses reforços, satisfazendo constrangimentos de volume/peso do reforço. Para avançar com a análise do problema, foi criado um modelo de base para a estrutura. Este modelo é composto por um painel de forma quadrada, apresentado na figura 2.6.

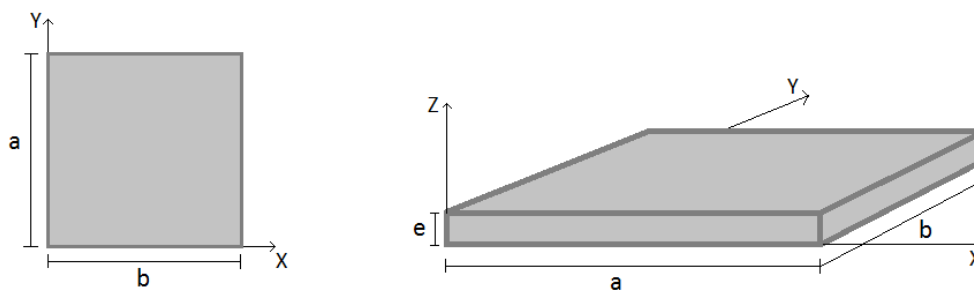


Figura 2.6: Painel/modelo base da estrutura.

Como indica a figura 2.6, o painel tem dimensões laterais de  $a$  e  $b$  e  $e$  de espessura. Se for aplicado  $a = b$ , é criada uma estrutura quadrada em que desta forma se pode obter um maior número de planos de simetria da estrutura, nomeadamente a simetria em torno das diagonais do quadrado. Posto isto, é possível também tirar partido desta simetria na criação dos reforços quando forem aplicadas cargas e constrangimentos igualmente simétricos relativamente a este plano.

### 2.3.2 Dados e Informações do Problema

Na segunda etapa da formulação é necessário desenvolver matematicamente o problema, e isso requer o conhecimento das propriedades dos materiais, desempenho dos requisitos, limites dos recursos, custo de matérias-primas, e outras informações relevantes. Além disso, a maioria dos problemas requer a capacidade de analisar projetos experimentais. Portanto, os procedimentos e ferramentas de análise são identificados nesta fase [48].

A maioria das ligas metálicas são consideradas materiais isotrópicos, onde, por definição, as propriedades do material são independentes da direção. Estes materiais têm apenas duas variáveis independentes na sua rigidez, chamadas constantes elásticas, em oposição às 21 constantes que existem no caso geral de materiais anisotrópicos. As duas constantes elásticas são geralmente expressas como o módulo de Young ( $E$ ) e o coeficiente de Poisson ( $\nu$ ), no entanto, em alternativa podem ser usadas as constantes  $K$  (módulo de compressibilidade) e/ou  $G$  (módulo de corte). Para materiais isotrópicos,  $G$  e

$K$  podem ser encontrados a partir de  $E$  e  $\nu$  por um conjunto de equações, e vice-versa.

Nesta condição de material isotrópico o módulo de cisalhamento ( $G$ ) relaciona-se com o módulo de Young ( $E$ ) e o coeficiente de Poisson ( $\nu$ ) pela seguinte equação:

$$\nu = \left( \frac{E}{2G} \right) - 1 \quad (2.1)$$

sendo o coeficiente de Poisson ( $\nu$ ) adimensional e o módulo de Young ( $E$ ) dado em  $Pa$ .

A Lei de Hooke para estes materiais pode ser expressa da seguinte forma:

$$\varepsilon_{ij} = \frac{1}{E}(\sigma_{ij} - \nu[\sigma_{kk}\delta_{ij} - \sigma_{ij}]) \quad (2.2)$$

ou;

$$\varepsilon = \frac{1}{E}(\sigma - \nu[tr(\sigma)I - \sigma]) \quad (2.3)$$

onde  $\varepsilon$  é o tensor extensão,  $\sigma$  o tensor tensão,  $\delta_{ij}$  o delta Kronecker e o  $tr(\sigma)$  é o traço do tensor tensão.

A forma de engenharia sob a qual a deformação ( $\varepsilon$ ) é expressa em termos do tensor das tensões ( $\sigma$ ) é dada por:

$$\varepsilon_{ij} = \frac{1}{2G}\sigma_{ij} \quad (2.4a)$$

$$\varepsilon_{ik} = \frac{1}{2G}\sigma_{ik} \quad (2.4b)$$

$$\varepsilon_{jk} = \frac{1}{2G}\sigma_{jk} \quad (2.4c)$$

Em forma de matriz, a lei de Hooke para estes materiais isotrópicos pode ser escrita como:

$$\begin{bmatrix} \varepsilon_{ii} \\ \varepsilon_{jj} \\ \varepsilon_{kk} \\ 2\varepsilon_{jk} \\ 2\varepsilon_{ik} \\ 2\varepsilon_{ij} \end{bmatrix} = \begin{bmatrix} \varepsilon_{ii} \\ \varepsilon_{jj} \\ \varepsilon_{kk} \\ \gamma_{jk} \\ \gamma_{ik} \\ \gamma_{ij} \end{bmatrix} = \frac{1}{E} \begin{bmatrix} 1 & -\nu & -\nu & 0 & 0 & 0 \\ -\nu & 1 & -\nu & 0 & 0 & 0 \\ -\nu & -\nu & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2(1+\nu) & 0 & 0 \\ 0 & 0 & 0 & 0 & 2(1+\nu) & 0 \\ 0 & 0 & 0 & 0 & 0 & 2(1+\nu) \end{bmatrix} \begin{bmatrix} \sigma_{ii} \\ \sigma_{jj} \\ \sigma_{kk} \\ \sigma_{jk} \\ \sigma_{ik} \\ \sigma_{ij} \end{bmatrix}$$

onde  $\gamma_{ij} = 2\varepsilon_{ij}$  é a deformação de corte, ou, pela relação inversa:

$$\begin{bmatrix} \sigma_{ii} \\ \sigma_{jj} \\ \sigma_{kk} \\ \sigma_{jk} \\ \sigma_{ik} \\ \sigma_{ij} \end{bmatrix} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{(1-2\nu)}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{(1-2\nu)}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{(1-2\nu)}{2} \end{bmatrix} \begin{bmatrix} \varepsilon_{ii} \\ \varepsilon_{jj} \\ \varepsilon_{kk} \\ 2\varepsilon_{jk} \\ 2\varepsilon_{ik} \\ 2\varepsilon_{ij} \end{bmatrix}$$



Recorrendo aos parâmetros de Lamé na elasticidade linear:

- $\lambda$ , também denominado primeiro parâmetros de Lamé;
- $\mu > 0$ , o módulo de cisalhamento ou segundo parâmetro de Lamé;

em que, nestes materiais homogêneos e isotrópicos, satisfazem a lei de Hooke tridimensional, podemos simplificar a equação para a forma:

$$\sigma = 2\mu\varepsilon + \lambda \operatorname{tr}(\varepsilon)I \quad (2.5)$$

$$\begin{bmatrix} \sigma_{ii} \\ \sigma_{jj} \\ \sigma_{kk} \\ \sigma_{jk} \\ \sigma_{ik} \\ \sigma_{ij} \end{bmatrix} = \begin{bmatrix} 2\mu + \lambda & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & 2\mu + \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & 2\mu + \lambda & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{bmatrix} \begin{bmatrix} \varepsilon_{ii} \\ \varepsilon_{jj} \\ \varepsilon_{kk} \\ 2\varepsilon_{jk} \\ 2\varepsilon_{ik} \\ 2\varepsilon_{ij} \end{bmatrix}$$

sendo  $\sigma$  o tensor tensão,  $\varepsilon$  o tensor deformação,  $I$  a matriz identidade e  $\operatorname{tr}(\cdot)$  a função traço.

### 2.3.2.1 Teoria de Placas

Muitas teorias de placas foram desenvolvidas desde o final do século XIX, mas, em engenharia, duas foram aceites e são amplamente utilizadas, são as teorias de:

- **Kirchhoff-Love**, utilizada para determinar tensões e deformações em placas finas sujeitas a forças e momentos aplicados.
- **Mindlin-Reissner**, uma extensão da teoria de placas de Kirchhoff-Love e é utilizada para calcular as deformações e tensões em placas cuja espessura é da ordem de um décimo das dimensões planares.

Como o projeto desenvolvido no presente trabalho abrange a modelação de um reforço numa placa fina, seguidamente, é apresentada a teoria mais apropriada a esta análise e que está de acordo com o elemento de placa utilizado, isto é, a teoria de placas de Kirchhoff-Love.

### 2.3.2.2 Teoria de Placas de Kirchhoff-Love

Uma placa é uma estrutura plana em que a espessura tem uma dimensão muito menor do que as outras dimensões, como indicado na figura 2.7 (Fonte: [67]). A placa pode ser referida à superfície média a qual bissecta a espessura em cada ponto. [67]

A teoria clássica de placas segue um conjunto de hipóteses que são as seguintes:

1. A placa é constituída por material elástico, homogéneo e isotrópico.
2. Inicialmente, a placa é plana.

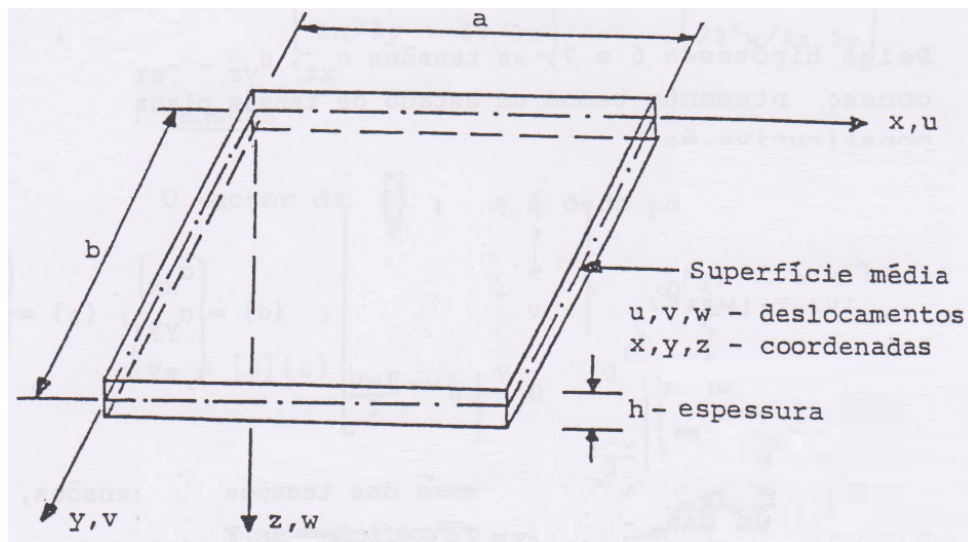


Figura 2.7: Sistema de coordenadas e superfície média de uma placa.

3. A espessura ( $h$ ) da placa é pequena em comparação com as restantes dimensões ( $a, b$ ), que são pelo menos 10 vezes maiores do que a espessura ( $a > 10 \cdot h, b > 10 \cdot h$ ).
4. O deslocamento transversal ( $w$ ) é pequeno em comparação com a espessura ( $h$ ), sendo o deslocamento máximo menor do que  $1/5$  da espessura ( $w_{max} < h/5$ ).
5. As tangentes da superfície média depois de uma deformação são pequenas em comparação com a unidade.
6. As deformações são tais que os planos normais à superfície média continuam perpendiculares à superfície média depois da deformada ( $\varepsilon_{xz} = \varepsilon_{yz} = 0$ ).
7. As tensões normais à superfície média são desprezáveis ( $\sigma_{zz} = 0$ ).
8. As extensões da superfície média são desprezáveis em comparação com as extensões de flexão ( $u = v = 0$  na superfície média).

O modelo matemático bidimensional desenvolvido por Kirchhoff–Love assume também um plano médio da superfície (figura 2.7). Este pode ser utilizado para representar uma placa tridimensional em forma bidimensional [68].

Esta teoria é uma extensão da teoria de vigas defendida por Euler-Bernoulli, e que foi desenvolvida por Love no ano de 1888, utilizando suposições propostas por Kirchhoff. A técnica da teoria clássica em flexão de placas finas ficou então tradicionalmente associada aos nomes de Kirchhoff e Love [69].

São consideradas as seguintes hipóteses neste modelo: (i) linhas retas normais à superfície média permanecem retas após a deformação; (ii) linhas retas normais à superfície média permanecem normais à superfície média após a deformação; (iii) a espessura da placa permanece a mesma durante a deformação. Deste modo, tanto os efeitos transversais de cisalhamento quanto os efeitos transversais normais são negligenciados [70].

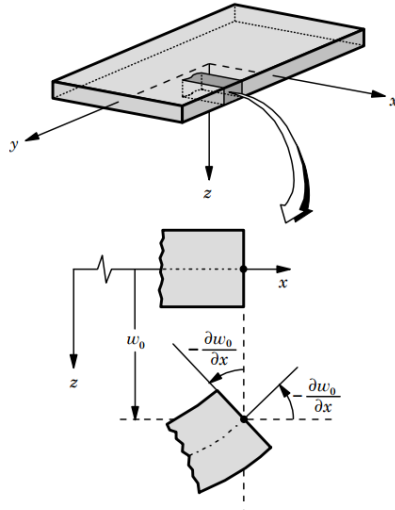


Figura 2.8: Placa plana e com deformação sob as hipóteses de Kirchhoff-Love.

### Campo de Deslocamentos

O campo deslocamento é definido pelas hipóteses 6 ( $\varepsilon_{xz} = \varepsilon_{yz} = 0$ ) e 8 ( $u = v = 0$ ). Com referência à figura 2.8 (Fonte: [69]), o campo deslocamento aproximado é dado por:

$$\begin{cases} u(x, y, z) = -z \frac{\partial w}{\partial x} \\ v(x, y, z) = -z \frac{\partial w}{\partial y} \\ w(x, y, z) = w_0(x, y) \end{cases} \quad (2.6)$$

onde  $\frac{\partial w}{\partial x}$  e  $\frac{\partial w}{\partial y}$  são as rotações dos planos perpendiculares à superfície média segundo o eixo  $x$  e  $y$ , depois da deformação [69, 67].

### Lei Constitutiva

Através das hipóteses 6 ( $\varepsilon_{xz} = \varepsilon_{yz} = 0$ ) e 7 ( $\sigma_{zz} = 0$ ), as tensões  $\sigma_{xz}$ ,  $\sigma_{yz}$  e  $\sigma_{zz}$  são nulas, e consequentemente temos um estado de tensão plana cuja lei constitutiva é:

$$\{\sigma\} = [D]\{\varepsilon\} \quad (2.7)$$

com;

$$\{\sigma\} = \begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{Bmatrix}; \quad [D] = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix}; \quad \{\varepsilon\} = \begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ 2\varepsilon_{xy} \end{Bmatrix}$$

onde  $\{\sigma\}$  e  $\{\varepsilon\}$  são os vetores das tensões e extensões,  $E$  é o módulo de elasticidade,  $\nu$  é o coeficiente de Poisson e  $[D]$  é a matriz constitutiva de flexão.

### Extensões

A partir das relações da teoria da elasticidade clássica para pequenos deslocamentos, e da equação (2.6), facilmente se pode calcular as extensões para uma placa fina. O vetor das extensões já simplifi-

cado, é então dado por:

$$\{\varepsilon\} = \begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ 2\varepsilon_{xy} \end{Bmatrix} = \begin{Bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \end{Bmatrix} = -z \begin{Bmatrix} \frac{\partial^2 w}{\partial x^2} \\ \frac{\partial^2 w}{\partial y^2} \\ 2 \frac{\partial^2 w}{\partial x \partial y} \end{Bmatrix} \quad (2.8)$$

Em função dessas deformações, as tensões são normalmente calculadas a partir das equação (2.7) da Lei Constitutiva.

### Tensões

Com a equação (2.7), a matriz constitutiva  $[D]$  e as extensões dadas pela equação (2.8), o vetor das tensões é dado por:

$$\{\sigma\} = [D]\{\varepsilon\} = \frac{-zE}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \begin{Bmatrix} \frac{\partial^2 w}{\partial x^2} \\ \frac{\partial^2 w}{\partial y^2} \\ 2 \frac{\partial^2 w}{\partial x \partial y} \end{Bmatrix} \quad (2.9)$$

Expandindo, as expressões das tensões vêm:

$$\sigma_{xx} = \frac{-Ez}{1-\nu^2} \left( \frac{\partial^2 w}{\partial x^2} + \nu \frac{\partial^2 w}{\partial y^2} \right) \quad (2.10)$$

$$\sigma_{yy} = \frac{-Ez}{1-\nu^2} \left( \frac{\partial^2 w}{\partial y^2} + \nu \frac{\partial^2 w}{\partial x^2} \right) \quad (2.11)$$

$$\sigma_{xy} = \frac{-Ez}{1+\nu} \left( \frac{\partial^2 w}{\partial x \partial y} \right) \quad (2.12)$$

### Momentos

As expressões dos momentos são obtidas com a integração das tensões ao longo da espessura ( $h$ ) da placa:

$$\begin{Bmatrix} M_{xx} \\ M_{yy} \\ M_{xy} \end{Bmatrix} = \int_{-\frac{h}{2}}^{\frac{h}{2}} \begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{Bmatrix} z dz = \int_{-\frac{h}{2}}^{\frac{h}{2}} D \begin{Bmatrix} -\frac{\partial^2 w}{\partial x^2} \\ -\frac{\partial^2 w}{\partial y^2} \\ -2 \frac{\partial^2 w}{\partial x \partial y} \end{Bmatrix} z^2 dz = \frac{h^3}{12} D \begin{Bmatrix} -\frac{\partial^2 w}{\partial x^2} \\ -\frac{\partial^2 w}{\partial y^2} \\ -2 \frac{\partial^2 w}{\partial x \partial y} \end{Bmatrix} \quad (2.13)$$

Expandindo, as expressões dos momentos vêm dadas por:

$$M_{xx} = -D \left( \frac{\partial^2 w}{\partial x^2} + \nu \frac{\partial^2 w}{\partial y^2} \right); \quad M_{yy} = -D \left( \frac{\partial^2 w}{\partial y^2} + \nu \frac{\partial^2 w}{\partial x^2} \right); \quad M_{xy} = -(1-\nu)D \left( \frac{\partial^2 w}{\partial x \partial y} \right) \quad (2.14)$$

onde  $D = \frac{Eh^3}{12(1-\nu^2)}$  é o coeficiente de rigidez de flexão da placa.

É de notar as relações existentes entre as tensões e momentos:

$$\sigma_{xx} = \frac{12zM_{xx}}{h^3}; \quad \sigma_{yy} = \frac{12zM_{yy}}{h^3}; \quad \sigma_{xy} = \frac{12zM_{xy}}{h^3} \quad (2.15)$$

### Equações de Equilíbrio

Para se determinar as equações de equilíbrio, é considerado um elemento infinitesimal de placa (dimensões,  $dx - dy - h$ ), sujeito a uma carga aplicada  $P_z$  e em estado de equilíbrio. Para este equilíbrio, é necessário que:

- a soma dos momentos em torno do eixo dos  $x$  seja nula;
- a soma dos momentos em torno do eixo dos  $y$  seja nula;
- a soma das forças na direção  $z$  seja nula.

Considerando o equilíbrio de momentos em torno do eixo dos  $y$ , e desprezando termos de segunda ordem, vem:

$$\frac{\partial M_{xx}}{\partial x} dx dy + \frac{\partial M_{yx}}{\partial y} dx dy - Q_x dx dy = 0 \quad (2.16)$$

ou

$$Q_x = \frac{\partial M_{xx}}{\partial x} + \frac{\partial M_{yx}}{\partial y}$$

Da mesma forma, considerando o equilíbrio de momentos em torno do eixo dos  $x$ :

$$Q_y = \frac{\partial M_{xy}}{\partial x} + \frac{\partial M_{yy}}{\partial y} \quad (2.17)$$

Considerando o equilíbrio de forças na direção  $z$ :

$$\frac{\partial Q_x}{\partial x} dx dy + \frac{\partial Q_y}{\partial y} dx dy + P_z dx dy = 0 \quad (2.18)$$

ou

$$\frac{\partial Q_x}{\partial x} + \frac{\partial Q_y}{\partial y} + P_z = 0$$

Substituindo as equações dos esforços transversos (2.16) e (2.17) na equação (2.18) obtemos a equação de placas em equilíbrio estático:

$$\frac{\partial^2 M_{xx}}{\partial x^2} + 2 \frac{\partial^2 M_{xy}}{\partial x \partial y} + \frac{\partial^2 M_{yy}}{\partial y^2} = -P_z(x, y) \quad (2.19)$$

[69] [67]

### 2.3.3 Identificação e Definição de Variáveis de Projeto

O passo seguinte no processo de formulação é identificar um conjunto de variáveis de projeto que descrevem o sistema. Em geral, elas são chamadas de variáveis de otimização, são consideradas

como livres porque são independentes umas das outras o quanto possível, e podemos atribuir-lhes qualquer valor dentro das restrições que são impostas no projeto [48].

A noção de melhorar ou otimizar uma estrutura implica implicitamente alguma liberdade para alterar essa estrutura. O potencial da transformação é normalmente expresso em termos da mudança admissível de um grupo de parâmetros, parâmetros esses que são chamados de variáveis de projeto em otimização estrutural [71].

Este conjunto de  $n$  variáveis de projeto é normalmente denominado como:

$$\mathbf{X} = (x_1, x_2, \dots, x_n) \quad (2.20)$$

Como já referido anteriormente, as variáveis de projeto podem ser de características geométricas, como por exemplo, a espessura de uma placa ou secção transversal de barras, e podem também ser propriedades do material, como o módulo de elasticidade e a densidade, etc. Para diferentes valores de variáveis de projeto, surgem esquemas/*design* diferentes a produzir, e o número de variáveis independentes especifica o número de graus de liberdade do problema.

Neste trabalho, as variáveis de projeto são as secções transversais das vigas utilizadas para o reforço do painel. Foram criadas secções de vigas quadradas, como se pode ver na figura 2.9, e as variáveis de projeto são mais precisamente a espessura da secção,  $t$ .

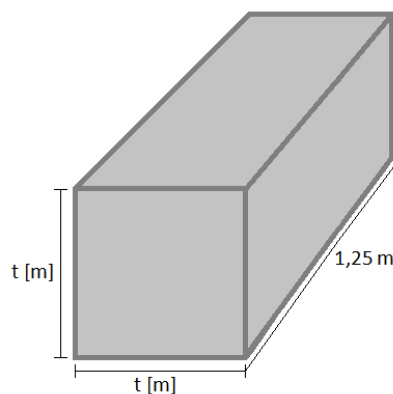


Figura 2.9: Forma das vigas/variáveis de projeto.

Estas vigas, numa primeira análise, foram utilizadas com variáveis do tipo contínuo, o que corresponde a ter espessuras  $t$  a variar em qualquer valor entre "zero" e "dez" centímetros ( $0 \text{ m} \leq t \leq 0,1 \text{ m}$ ), isto é, podem variar entre não ter viga e ter o máximo de viga estabelecido. Numa segunda análise, são utilizadas variáveis do tipo discreto (ou inteiro), onde o programa funciona de uma maneira mais simples na criação deste tipo de reforços, ou seja, as vigas a colocar como reforço no painel só podem ter uma medida de secção transversal, o que leva o programa otimizador a decidir se coloca ou não a viga no sítio previamente estabelecido.

### 2.3.4 Identificação da Função Objetivo

Certamente que não poderá haver muitos/vários projetos viáveis para um sistema, e alguns serão melhores que outros. Então para comparar estes diferentes modelos, deve-se dispor de um critério. O critério deve ser uma função cujo valor numérico pode ser obtido cada vez que um *design* é determinado, isto é, o critério deve ser uma função geralmente não linear determinada em função das variáveis de projeto (vetor  $\mathbf{X}$ ). O critério é normalmente chamado de função objetivo,  $f(\mathbf{X})$ , e precisa ser maximizado ou minimizado, dependendo dos requisitos de problema [48].

A noção de otimização implica que exista algum valor na função  $f(\mathbf{X})$ , ou nas funções  $f(\mathbf{X}) = [f_1(\mathbf{X}), f_2(\mathbf{X}), \dots, f_p(\mathbf{X})]$ , que pode ser melhorado, e pode ser utilizado como medida de eficácia de *design*. Para problemas estruturais de otimização de peso, deslocamentos, tensões, frequências de vibração e deformação, ou qualquer combinação destes, podem ser usadas funções objetivo [71].

Para maximizar a rigidez de uma estrutura que se encontra em domínio elástico, a rigidez pode ser determinada através da minimização do trabalho realizado pelas forças aplicadas na estrutura, quando esta se encontra no seu estado de equilíbrio. Este trabalho é designado de **compliance**,  $C$ , e, de forma indireta, a minimização desta grandeza permite minimizar os deslocamentos provocados por essas cargas/forças. Portanto, pode-se considerar a *compliance*,  $C$ , como uma função objetivo,  $l(\mathbf{X})$ :

$$C = l(\mathbf{X}) \quad (2.21)$$

Para tornar possível a formulação com o Método de Elementos Finitos (MEF) na resolução do problema estático, o campo deslocamentos de um elemento de volume,  $\{v\}$ , é dado por:

$$\{v\} = [N]\{d\} \quad (2.22)$$

onde  $[N]$  é a matriz das funções de forma do elemento, e  $\{d\}$  é o vetor de deslocamentos nodais do elemento (vetor dos graus de liberdade). Tendo isto, e novamente para um elemento de volume, pode-se definir o vetor do campo de extensões como:

$$\{\varepsilon\} = [\Delta]\{v\} = [\Delta][N]\{d\} = [B]\{d\} \quad (2.23)$$

onde  $[\Delta]$  é a matriz do operador diferencial, e  $[B]$  é a matriz extensões do elemento.

Constata-se também que, para um elemento de volume, a equação de equilíbrio estático toma a seguinte forma:

$$[K_e]\{d\} = \{P\} \quad (2.24)$$

com

$$[K_e] = \int_{V_e} [B]^T [D] [B] dV_e \quad (2.25)$$

$$P = \int_{S_e} [N]^T \{ps\} dS_e \quad (2.26)$$

onde o  $[K_e]$  é a matriz de rigidez do elemento,  $\{P\}$  o vetor de forças de superfície do elemento,  $[D]$  é a matriz constitutiva de flexão, e  $\{ps\}$  é o vetor das forças por unidade de área atuando sobre a superfície.

A matriz de rigidez global,  $[K]$ , é obtida através da assemblagem de todas as matrizes de rigidez elementares,  $[K_e]$ , que constituem a estrutura.

O campo deslocamentos,  $u$ , associado à posição de equilíbrio da estrutura pode ser obtido através da equação:

$$F = [K]u \quad (2.27)$$

onde  $F$  representa o vetor de forças aplicado na estrutura.

Com o vetor de forças,  $F$ , e o campo deslocamentos,  $u$ , pode-se confirmar a compliance,  $C$  dada por:

$$C = F^T u \quad (2.28)$$

Por outro lado a energia elástica de deformação total da estrutura,  $U$ , é definida como:

$$U = \frac{1}{2} u^T [K] u \quad (2.29)$$

Utilizando a equação (2.28), e combinando com a equação (2.27), obtemos a relação da *compliance*,  $C$ , com a energia elástica de deformação total da estrutura,  $U$ , ou seja:

$$C = F^T u = u^T [K] u = 2U \quad (2.30)$$

onde a *compliance*,  $C$ , está linearmente relacionada com a energia elástica de deformação total da estrutura,  $U$ . [62]

Num trabalho com estas características, tanto a *compliance* como a energia elástica de deformação podem ser tratadas como função objetivo. No entanto na realização deste projeto, para otimizar o reforço do painel, a **energia elástica de deformação total da estrutura,  $U$** , foi a escolhida como função objetivo. Tendo isto, pode-se dizer que a equação para a função objetivo,  $f(X)$ , a ser tratada é dada como:

$$f(X) = U \quad (2.31)$$

### 2.3.5 Identificação dos Constrangimentos

A última etapa do processo de formulação do problema de otimização diz respeito à identificação e desenvolvimento da formulação das restrições. Estas restrições são constrangimentos, que num



problema de otimização, podem ser definidos em dois tipos, isto é, constrangimentos de igualdade ou de desigualdade.

Nos constrangimentos de desigualdade,  $g_k$ , como é o caso dos limites superiores e inferiores impostos às variáveis de projeto, as restrições são expressas por:

$$g_k(X) = g_k(x_1, x_2, \dots, x_n) \leq 0; \quad k = 1, \dots, m \quad (2.32)$$

com  $m$  restrições de desigualdade.

Se os constrangimentos forem de igualdade,  $h_j$ , como por exemplo, as equações de equilíbrio de uma estrutura, as restrições vêm na forma:

$$h_j(X) = h_j(x_1, x_2, \dots, x_n) = 0; \quad j = 1, \dots, p \quad (2.33)$$

sujeito às  $p$  restrições de igualdade [48].

## 2.4 Formulação Sintetizada do Problema

Neste problema de otimização, é necessário encontrar o ótimo vetor das variáveis de projeto  $X = (x_1, x_2, \dots, x_n)$ , de modo a minimizar a função objetivo,  $f(\mathbf{X})$ , que está sujeita às restrições de projeto, como indica em (2.34). A maximização da rigidez da estrutura sujeita a um determinado carregamento e condições de fronteira, pode ser formulada de tal forma que se considere um volume fixo de material existente na estrutura. O problema é elaborado da forma seguinte:

$$\begin{array}{ll} \text{Minimizar} & f(X, u) \\ \text{Sujeito} & \longrightarrow V(X) = V_{\text{imposto}} \\ & \longrightarrow [K(X)]u = F \end{array} \quad (2.34)$$

onde  $V$  é o volume da estrutura,  $F$  é o vetor de forças aplicado na estrutura,  $u$  é o vetor deslocamentos e  $[K(X)]$  é a matriz rigidez global que se obtém com recurso ao programa de elementos finitos.

Para as variáveis de projeto,  $X = (x_1, x_2, \dots, x_n)$ , foram utilizados os dois tipos de constrangimentos, de igualdade e de desigualdade. Como já referido anteriormente, houve uma breve análise com variáveis do tipo contínuo, onde as variáveis de projeto podem assumir qualquer valor entre "zero" e "um", o que implica incluir constrangimentos de desigualdade, ou seja:

$$0 \leq x_i \leq 1 \implies \begin{cases} -x_i \leq 0, & i = 1, 2, \dots, n \\ x_i - 1 \leq 0, & i = 1, 2, \dots, n \end{cases} \quad (2.35)$$

Nas restantes análises, as variáveis de projeto são do tipo discreto, onde estas assumem somente dois valores, "zero" (viga ausente na estrutura) e "um" (viga presente na estrutura), o que implica utilização de constrangimentos de igualdade:

$$\begin{cases} x_i = 0, & i = 1, 2, \dots, n \\ \text{ou} \\ x_i = 1, & i = 1, 2, \dots, n \end{cases} \quad (2.36)$$

Concluída a formulação do problema de otimização, é agora necessário estabelecer o algoritmo de otimização a ser utilizado.

## 2.5 Algoritmo de Otimização

Um processo de otimização pode ser classificado e dividido em duas classes: global e local.

- A otimização global visa encontrar a melhor solução do conjunto de "todas" as soluções possíveis independentemente das condições iniciais do processo de procura.
- A otimização local encontra a melhor solução dentro de um conjunto de soluções que estão próximas de outra. A solução encontrada depende do ponto inicial do processo de procura.

Para um processo de otimização global são necessários algoritmos computacionais, com elevada complexidade, no entanto, como a maioria dos algoritmos têm algumas incertezas nos seus passos computacionais é prudente controlar o processo iterativamente de forma a otimizar da melhor forma a função objetivo, atendendo sempre aos critérios de paragem estabelecidos, aos valores iniciais das variáveis de projeto ( $X$ ), bem como às restrições impostas ao problema.

Se considerarmos um problema de minimização, e se a função objetivo  $f(X)$  tiver vários mínimos locais, isto leva a que a escolha inicial das variáveis de projeto ( $X$ ) determine qual deles será calculado/localizado [48, 72].

Em grande parte dos casos, a otimização não é eficiente o suficiente para gerar soluções em tempo disponível. Assim sendo, para o algoritmo de otimização ter um tempo de execução aceitável com soluções muito provavelmente corretas, é necessário a utilização de algoritmos heurísticos.

Os algoritmos heurísticos são bastante utilizados em problemas de otimização, contudo, apesar de não efetuarem diretamente uma procura em otimização pura, geram soluções aceitáveis, são de fácil implementação e computacionalmente mais eficientes. Outra característica deste algoritmo é o de que sempre gera uma solução para o problema, mesmo que em algumas dessas vezes as soluções não sejam de boa qualidade. Se um problema é resolvido repetitivamente e os parâmetros se alteram constantemente, as chances de falha de uma heurística são consideravelmente maiores. No entanto, o algoritmo heurístico pode ser melhorado através da incorporação de algoritmos localizados. Melhores soluções são geradas para um subconjunto de condições utilizando alguns passos de otimização local. Contudo, não é garantido que a melhor solução seja alcançada num problema de otimização.

Na figura 2.10, é apresentado um fluxograma ilustrativo de um algoritmo de otimização.

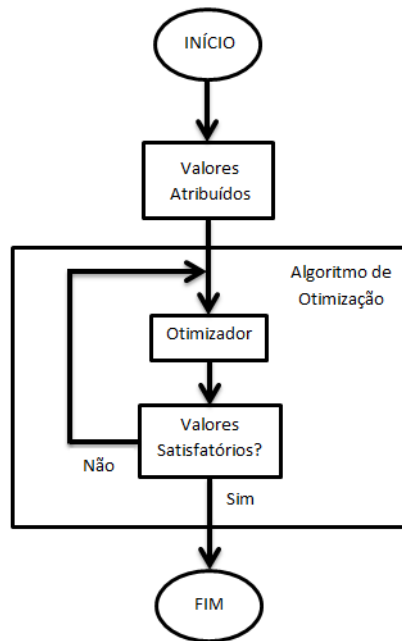


Figura 2.10: Fluxograma representativo de um algoritmo de otimização.

Tendo em conta os critérios de paragem e os constrangimentos do problema, os algoritmos de otimização consistem num processo iterativo no qual são efetuados sucessivas avaliações a uma função objetivo de forma a encontrar o seu valor ótimo [48].

## Capítulo 3

# Modelo Computacional

Neste capítulo, são apresentados de forma mais detalhada as implementações e procedimentos utilizados no modelo computacional deste projeto recorrendo à utilização dos *softwares* MATLAB e ANSYS. A análise estrutural é efetuada pelo programa de elementos finitos (ANSYS) e o algoritmo de otimização é implementado pela *Global Optimization Toolbox* do MATLAB, ou pelo novo algoritmo de otimização GLODS [63].

A interface entre a análise de elementos finitos e o algoritmo de otimização é efetuada também pelo programa MATLAB.

A estrutura geral do programa computacional desenvolvido, onde se verifica a interação entre o MATLAB e o ANSYS está representada no esquema da figura 3.1.

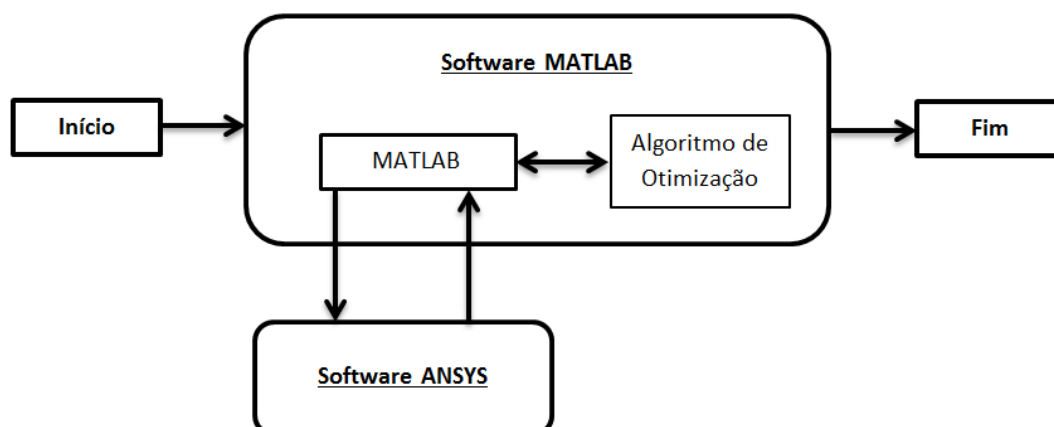


Figura 3.1: Esquema representativo da interação MATLAB-ANSYS

### 3.1 Conexão MATLAB-ANSYS

Este subcapítulo pretende descrever, de forma geral, a conexão entre os dois *softwares* utilizados para a realização deste estudo.

A plataforma de interface é praticamente toda desenvolvida em MATLAB, onde se constrói uma ligação entre os programas. O MATLAB fornece os resultados dos cálculos efetuados inicialmente e, subsequentemente, feita a análise em elementos finitos, o ANSYS devolve os resultados e soluções correspondentes ao MATLAB. Gera-se um ciclo computacional de cálculos e avaliações tal como ilustrado na figura 3.1 [73].

Para o utilizador ter acesso a esta conexão entre o MATLAB e o ANSYS, terá de abrir o ficheiro com a função MATLAB designada de **funcFF.m**. O programa desenvolvido pode ser adaptado para correr em qualquer computador. Para isso, basta o utilizador aceder ao ficheiro e efetuar as alterações dos parâmetros que definem as diretorias, substituindo estas pelas diretorias correspondentes ao ambiente informático do utilizador.

O ficheiro **funcFF.m** está dividido em duas partes: a primeira diz respeito à diretoria de localização do ficheiro de texto (**Constante.txt**) que é enviado pelo MATLAB ao ANSYS, e a segunda parte diz respeito às diretorias de localização do programa de elementos finitos (ANSYS) e à diretoria da pasta onde será escrito o ficheiro de resultados (**Energia.txt**) de cada avaliação.

O ANSYS funciona em modo automático, isto é, não tem qualquer intervenção do utilizador para efetuar a ligação, pois a conexão MATLAB-ANSYS está definida para iniciar o ANSYS em *batch mode* com o recurso a um comando em *DOS, IBM Corp.* É importante que os ficheiros **Constante.txt** e **Energia.txt** sejam criados na mesma pasta de diretoria, pois não só facilita a sua localização como favorece a organização das inúmeras pastas que podem ser criadas para as diversas análises e configurações em volta de um simples reforço num painel plano.

### 3.2 Cálculos e Dados Iniciais

Ao iniciar uma análise computacional de uma estrutura, é necessário atribuir ao programa alguns parâmetros e restrições iniciais para que o processo de otimização arranque de forma eficiente. Para iniciar o programa, o utilizador tem de aceder ao ficheiro **driver-glods-progressivo.m**, depois de verificar/alterar a configuração do reforço da estrutura através do vetor inicial das variáveis de projeto **x-ini** e iniciar os limites superiores (**ubound**) e inferiores (**lbound**) das mesmas. Deste modo pode dar arranque ao programa de otimização computacional.

O vetor **x-ini** pode ser criado em modo aleatório ( $x-ini=[];$ ) ou em modo previamente estabelecido (como por exemplo:  $x-ini=[0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ 1\ 1\ 0\ 0\ 0];$ ). A diferença entre ambos é que existem análises em que é importante começar com uma certa configuração já otimizada anteriormente ou com uma forma pressuposta pelo engenheiro. Por vezes, estas decisões podem poupar muito tempo de cálculo e memória computacional.

Outro dado importante, é a fixação do número de reforços a colocar no painel. Este parâmetro

é designado pela variável **NN**, e, para o alterar, é necessário aceder ao ficheiro MATLAB **fround.m**. O **NN** estabelece o numero exato de vigas a aplicar no painel, mas, no entanto, existe um número máximo de posições para estas mesmas vigas, dependendo este de algumas limitações impostas que são atribuídas relativamente a zonas onde se pode ou não colocar reforços. Estes diferentes casos estão apresentados de forma mais detalhada na secção 3.3.1.4.

### 3.3 Análise de Elementos Finitos

O método utilizado para implementar a análise de elementos finitos de um painel quadrangular (como demonstrado na figura 2.6 representada na secção 2.3.1) vai ser descrito neste subcapítulo.

É através do *software* ANSYS que é executada a análise em elementos finitos, onde os códigos são escritos em forma de comandos APDL (Ansys Parametric Design Language) [74].

Estes códigos criados são então gravados num ficheiro **.txt** (com a designação de **codigo.txt**) e, juntamente com o ficheiro **Constante.txt**, contêm toda a informação necessária para o programa correr e efetuar análises sem qualquer dificuldade. Seguidamente a cada avaliação, o ANSYS calcula a energia elástica da estrutura e concede o resultado através do ficheiro **Energia.txt** ao MATLAB, correspondendo este resultado à avaliação feita conforme as escolhas inicialmente definidas pelo utilizador. Na figura 3.2 pode-se verificar e compreender o processo de funcionamento do programa.

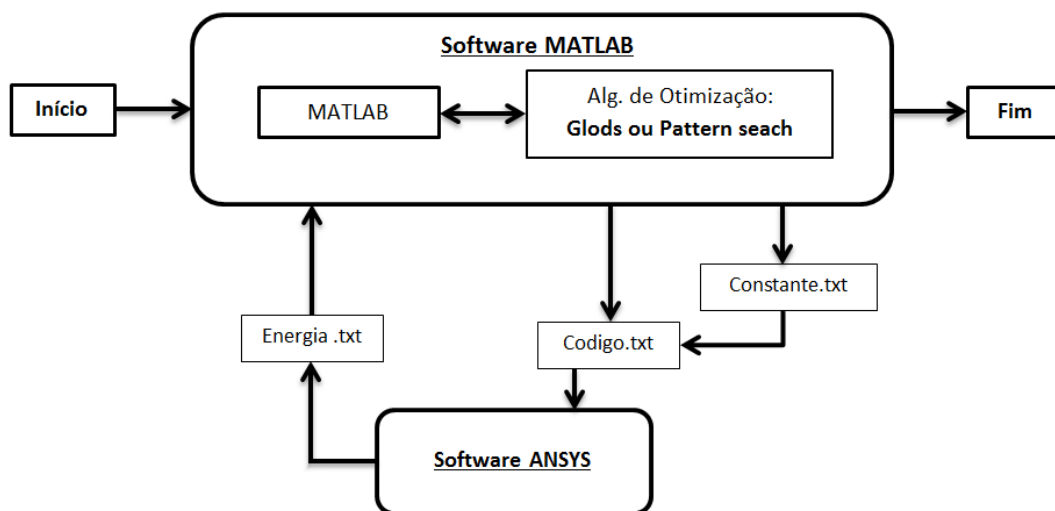


Figura 3.2: Esquema do programa computacional

#### 3.3.1 Pré-Processamento

Neste tópico, vai-se explicar a construção do modelo de painel implementado no programa de elementos finitos (ANSYS), levando em conta todas as formulações convenientes para a sua execução, ou seja, mais concretamente, vai-se explicar a formulação utilizada em código APDL.

A construção de um modelo estrutural a ser implementado em programação de elementos finitos necessita de alguns procedimentos recorrentes. Nos tópicos seguintes, está descrito como foram tratados

estes procedimentos na criação do modelo de painel explorado neste trabalho.

### 3.3.1.1 Construção do Modelo do Painel

De modo a modelar o painel de acordo com os objetivos do problema de otimização descritos anteriormente na secção 2.3.1, o painel foi construído em código APDL (designado de *codigo.txt*), de maneira que este consiga aceder aos dados enviados pelo MATLAB (*Constante.txt*) e que o utilizador possa também alterar as suas dimensões se assim o entender. As dimensões estão então definidas no ficheiro *codigo.txt*, como largura (**a**), comprimento (**b**) e espessura (**espessura**).

As propriedades do material utilizado no elemento estão apresentadas na tabela 3.1.

Propriedades do Material	
Módulo de Elasticidade	290 GPa
Coefficiente de Poisson	0.3

Tabela 3.1: Propriedades do material utilizado.

### 3.3.1.2 Malha

A malha representativa da estrutura, é criada de maneira a gerar elementos quadrangulares ou retangulares. No entanto, para as análises efetuadas neste trabalho, apenas foram utilizados elementos quadrangulares. Para controlar a configuração e a refinação da malha, o utilizador tem de recorrer ao parâmetro *num* que também se encontra no início do ficheiro *codigo.txt*. O *num* controla o número de divisões existente nas linhas criadas entre cada *keypoint*. Assim, de forma indireta, consegue-se controlar o número de elementos que constituem a malha do painel.

Para este trabalho, a refinação da malha foi estabelecida de maneira a conciliar dois pontos importantes, ou seja, ter uma boa solução na análise de elementos finitos (convergência da solução), e por outro lado, ser um processo de cálculo não muito pesado computacionalmente, para não demorar muito demasiado tempo. Levando estes dois pontos em prática, chegou-se a um valor de parâmetro, *num* = 5, ou seja, correspondente a uma malha constituída por 10000 elementos.

### 3.3.1.3 Seleção dos Tipos de Elemento

Na implementação de um modelo de estrutura constituído por um painel plano com um reforço em vigas, é necessário seleccionar dois tipos de elementos para este estudo, um elemento representante do painel e outro para viga.

Para representar um painel plano pode-se recorrer a vários tipos de elementos *SHELL*, como *SHELL63*, *SHELL93*, *SHELL181* e *SHELL281*. Para a representação da viga, existem elementos do tipo *BEAM*, como por exemplo, *BEAM188* e *BEAM189*. Analisando a aplicação de cada tipo de elemento e as respetivas soluções previstas, os tipos de elementos escolhidos para serem utilizados na modelação computacional do painel e viga, foram respetivamente o *SHELL93* e o *BEAM189* [74].

O elemento de placa *SHELL93* é definido por 8 nós, com seis graus de liberdade em cada nó (translações e rotações nas direções x, y, z). Este elemento não leva em conta as deformações de

cisalhamento ao longo da espessura do painel, o que vai de acordo com a teoria de placas de Kirchhoff-Love tratada na secção 2.3.2.2. O elemento de viga *BEAM189* é constituído por 3 nós e, tal como o elemento de placa, tem seis graus de liberdade em cada nó (translações e rotações nas direcções x, y, z). Na figura 3.3, está representado um esquema ilustrativo dos elementos utilizados.

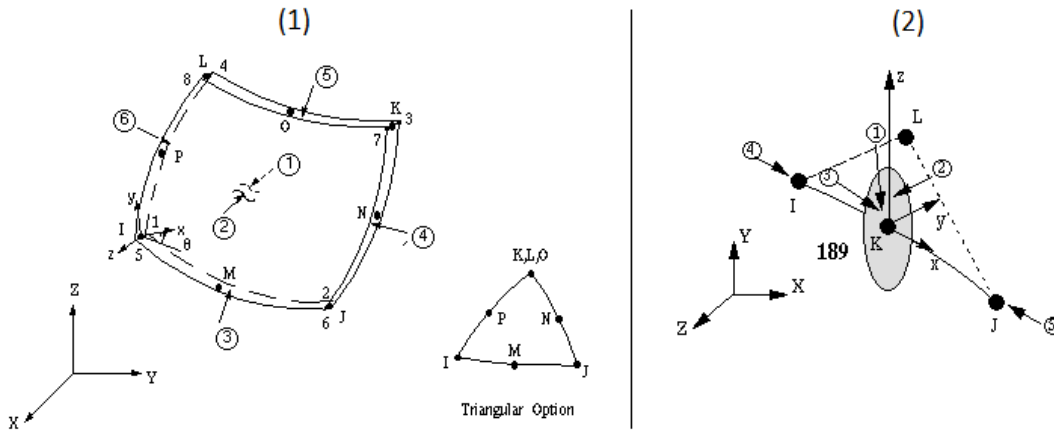


Figura 3.3: Esquema representativo dos elementos: (1) *SHELL93* e (2) *BEAM189*

### 3.3.1.4 Definir Reforços para o Painel

Para o programa computacional de elementos finitos ter capacidade de gerar e otimizar o reforço, é necessário criar posições previamente estabelecidas onde as vigas possam ser introduzidas. Existe sempre um número superior de posições relativamente ao número de vigas a colocar, ou seja, com este processo, gera-se sempre muitas hipóteses de reforço para as mesmas condições impostas à estrutura. No entanto, todas estas diferentes estruturas têm uma rigidez própria e somente poucas destas tendem a convergir para um valor máximo, isto é, existe uma configuração ideal para cada ensaio, mas não quer dizer que não hajam configurações parecidas tão aperfeiçoadas como a estrutura ideal. Isto é muito importante no âmbito da engenharia, pois estruturas topologicamente diferentes podem ser idênticas em termos de características mecânicas, o que leva a um grande benefício na escolha e no processo de fabricação.

Para este estudo foram projetados dois perfis de possíveis posições para colocar vigas, um perfil de 24 e outro de 32, que foram criados de maneira a que as posições fiquem igualmente espaçadas entre si e cobrindo todo o domínio do painel. Em ambos os casos, o número de reforços a colocar em cada estrutura é predefinido inicialmente pelo utilizador (explicado anteriormente na secção 3.2).

Na figura 3.4, estão os esquemas representativos das estruturas de 24 e 32 posições, que computacionalmente são tratadas como variáveis de projeto.

No esquema do lado esquerdo, está representado o perfil com a configuração de 24 variáveis, onde o programa só permite implementar reforços no interior do painel. Este perfil é muito eficaz em cargas aplicadas na zona do canto superior esquerdo do painel, isto porque essas cargas encontram-se afastadas das extremidades livres e mais perto dos apoios do painel, ou seja, na zona mais rígida da estrutura.



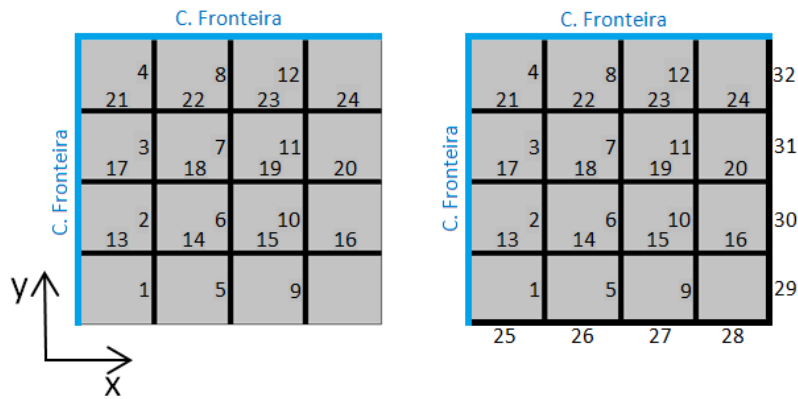


Figura 3.4: Esquemas de 24 (à esquerda) e 32 variáveis (à direita)

No esquema da direita, foram adicionadas mais 8 variáveis ao programa, ficando com um total de 32 variáveis. Este aumento de número de variáveis deveu-se basicamente a consentir à estrutura a instalação de vigas ao longo das extremidades livres do painel. É uma opção mais vantajosa em termos de resultados, pois não só consegue obter as configurações do perfil de 24 variáveis como produzir resultados ainda mais interessantes, nomeadamente em cargas aplicadas perto das extremidades livres do painel.

Outra análise interessante para este estudo é tirar partido da simetria da estrutura, de maneira a simplificar o programa de elementos finitos. A ideia foi manter o mesmo número de posições disponíveis mas reduzir para metade o número de variáveis de projeto, tal como mostra a figura 3.5. Para tal, foi criado um novo programa correspondente a cada um dos casos descritos acima, mas com algumas limitações.

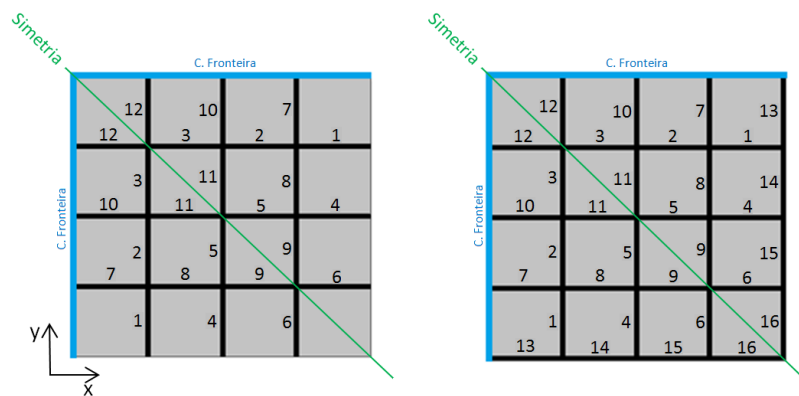


Figura 3.5: Esquemas de 12 (à esquerda) e 16 variáveis (à direita)

As variáveis de projeto que preenchem todas as posições na parte de inferior ao plano de simetria são novamente repetidas na parte superior, correspondendo assim duas vigas para uma só variável de projeto. Estes esquemas simétricos têm características que os diferenciam dos esquemas anteriores, ou seja:

- as configurações geradas neste programa são sempre simétricas em relação ao plano de simetria;
- o programa computacional é bastante mais leve e rápido a gerar soluções;

- são limitados a colocar número par de reforços, mesmo escolhendo número ímpar de variáveis de projeto.

### 3.3.2 Processamento

Depois da construção do modelo de painel a implementar no programa de elementos finitos tratado anteriormente (tópico 3.3.1), há necessidade de criar condições essenciais para analisar estaticamente o modelo e originar casos de estudo. Para isso, recorre-se ao processamento de dados. O processamento é uma série de atividades executadas ordenadamente, que resultará em uma espécie de arranjo de informações, onde inicialmente são coletadas informações, ou dados, que passam por uma organização onde, no final, será transmitido ao utilizador o dado pertinente à sua busca, ou objetivo. A obtenção inicial de dados (informações) é denominada de processamento. Nos tópicos seguintes é descrito o processamento implementado.

#### 3.3.2.1 Implementação das Variáveis de Projeto

As variáveis de projeto iniciais são convertidas do formato de "zeros" e "uns" (tratado na secção 3.2) para o formato das dimensões "0.0001" e "0.1", aplicadas no ANSYS, e são gravadas no ficheiro **Constante.txt**, isto tudo controlado pela função de Matlab **funcFF.m**, ficando assim guardadas em pasta, aguardando a "chamada" do código APDL (**codigo.txt**). É de notar que, no ANSYS é impossível definir posições para colocar vigas sem atribuir qualquer espessura, o que levou a recorrer à dimensão 0.0001 *m* para simular a ausência de viga nessa posição.

O ANSYS é inicializado e corre o ficheiro **codigo.txt** que também funciona com a leitura em paralelo do ficheiro **Constante.txt**. Esta leitura é possível graças à utilização do comando **\*Use**, que funciona como se as variáveis de projeto já estivessem inseridas no código. Depois do programa correr o ficheiro **codigo.txt**, este dá origem a uma estrutura constituída por uma geometria obtida pela análise topológica do algoritmo de otimização. Em cada ciclo de avaliação estática da estrutura, é gerado um novo ficheiro **Constante.txt**, correspondendo a uma nova configuração de vigas implementadas no painel.

A implementação de vigas no programa de elementos finitos, ANSYS, é efetuado através da definição de linhas onde estas vão ser colocadas e a atribuição de espessuras (que correspondem às variáveis de projeto). Para a construção de um modelo da estrutura bem definido, é importante definir as localizações corretas e ao pormenor das vigas soldadas no painel. Para ajudar a compreender este processo, verifiquemos as figuras 3.6, 3.7 e 3.8, obtidas pelo ANSYS.

A figura 3.6, representa um pormenor da estrutura construída sem a adição de qualquer *offset* nas vigas. Como se pode observar, com este processo, as vigas ficam inseridas dentro do painel, o que fica longe de corresponder à estrutura que se quer obter. Tendo isto, houve então necessidade de reconstruir todo o processo de implementação de vigas com a adição de *offsets*, para assim garantir a estrutura desejada.

As figuras seguintes, apresentam no seu lado esquerdo uma imagem obtida no ANSYS, que demonstra localmente as vigas com a adição de *offsets* no programa, para que estas não fiquem contidas

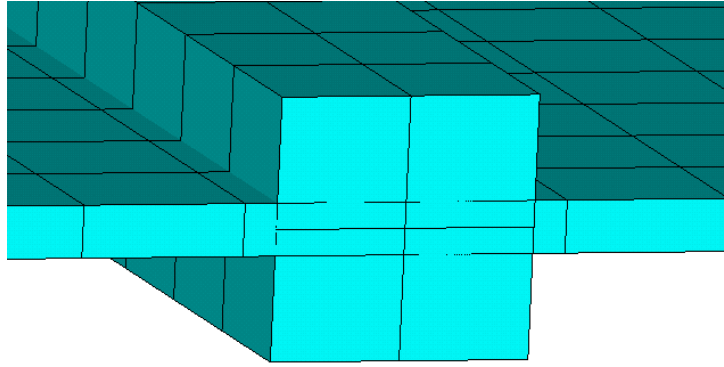


Figura 3.6: Imagem de um pormenor da estrutura sem adição de *offsets*.

dentro do painel. Isto porque no ANSYS, as vigas são definidas por uma linha que tem de ser coincidente com outra linha do painel, então, é necessário realizar um deslocamento de forma a ajustar as vigas à posição desejada para a estrutura. Do lado direito das figuras, é representado um esquema ilustrativo de como foi consumado o processo de deslocação.

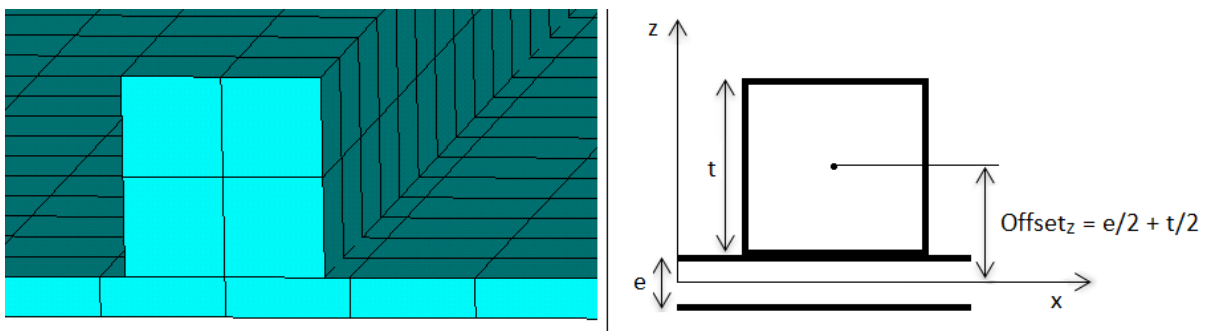


Figura 3.7: Esquema representativo do deslocamento vertical atribuído nas vigas (*offset-z*).

A figura 3.7, demonstra os deslocamentos verticais (*offset-z*) efetuados nas vigas que são introduzidas em zonas "interiores" do painel, ou seja, em locais somente descritos nos esquemas de 24 posições, tratados na secção 3.3.1.4. Também está representado a forma como se obteve este deslocamento, onde o (**e**) corresponde à espessura do painel, e o (**t**) à largura da viga.

Em seguida, na figura 3.8, está retratado o processo de implementação dos restantes 8 reforços que podem ser inseridos nos bordos do painel dos esquemas de 32 posições.

Neste caso da figura 3.8, podemos observar a representação dos *offsets* atribuídos aos reforços localizados no bordo do painel. Neste local, houve necessidade de se deslocar as vigas em desvios verticais e horizontais da sua posição inicial, para que deste modo, a base da viga fique totalmente em contato com o painel, e assim poder ser futuramente soldada nesta posição. Do lado direito da figura, pode-se observar também a forma como se implementou estes deslocamentos no código desenvolvido. As variáveis (**e**) corresponde igualmente à espessura do painel, e o (**t**) à largura da viga.

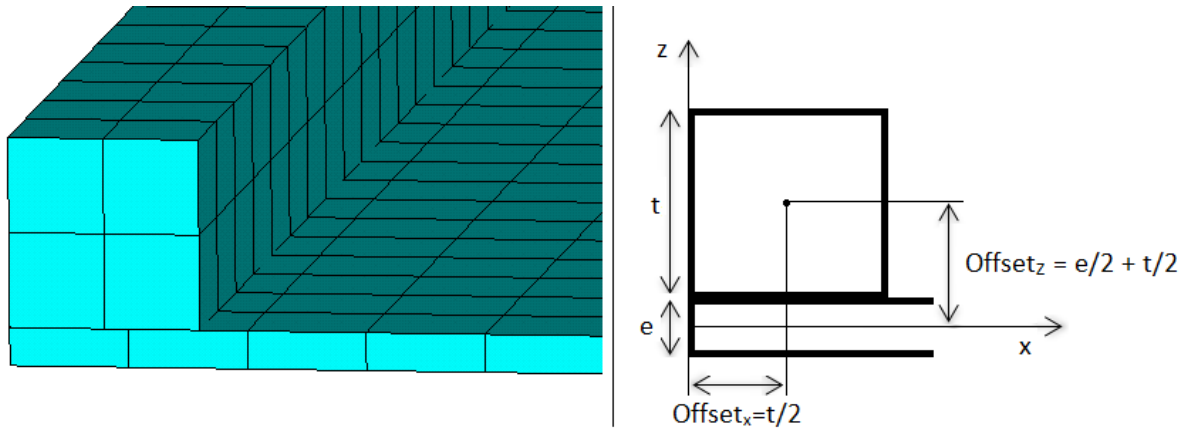


Figura 3.8: Esquema representativo do deslocamento vertical e horizontal atribuído às vigas do bordo do painel (offset-x e offset-z).

### 3.3.2.2 Condições de Fronteira

Ao longo da evolução deste trabalho, foram sendo criadas inúmeras condições de fronteira para melhor testar a capacidade computacional em gerar o reforço do painel pretendido, no entanto, prevaleceu somente uma condição de fronteira, para assim ser possível comparar as diferentes configurações que são geradas pela alteração de outros parâmetros e restrições. No programa computacional construído no ficheiro *codigo.txt*, está criada esta condição de fronteira, que se baseia em ter o painel simplesmente apoiado em dois lados consecutivos, como indicado na figura 3.9.

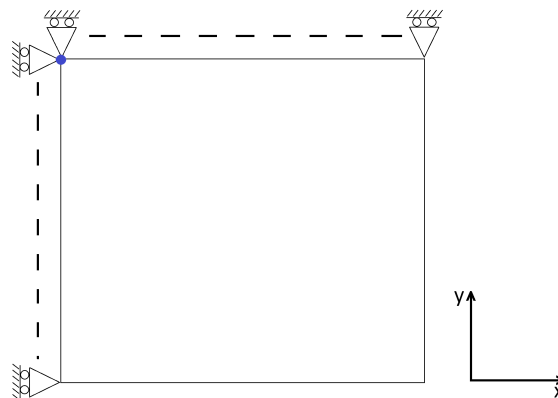


Figura 3.9: Condições de fronteira do painel.

### 3.3.2.3 Cargas Aplicadas

Tirando partido da simetria formada no painel (como indicado na secção 2.3.1), através da geometria e das condições de fronteira aplicadas, houve também necessidade de colocar os carregamentos em certas zonas, para continuar a ser uma análise de estrutura simétrica. Foram efetuados 3 carregamentos distintos, o primeiro (**F1**) no ponto central do painel, o segundo (**F2**) desviado para baixo e para a direita, e o terceiro (**F3**) na ponta do painel, como se podem ver representados na figura 3.10.

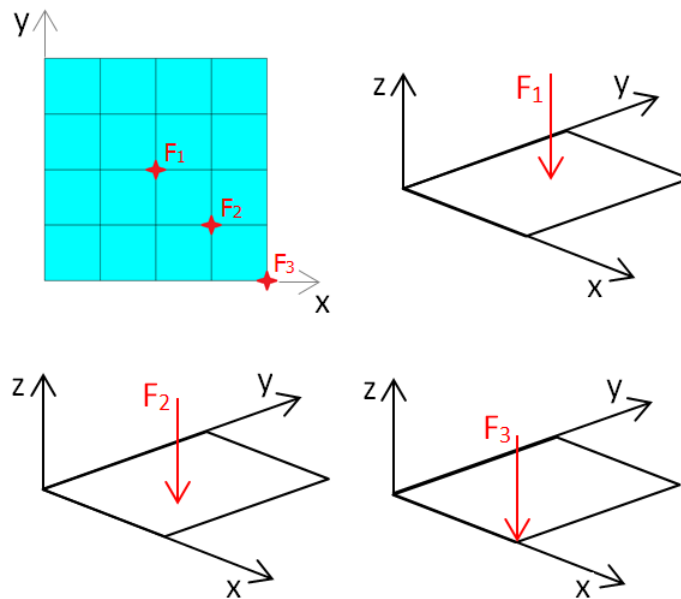


Figura 3.10: Cargamentos aplicados no painel.

Todos estes carregamentos efetuados têm a mesma força aplicada ( $F=5000\text{ N}$ ), pois ao serem aplicados em localizações diferentes, o reforço obtido também vai ser diferente.

### 3.3.2.4 Análise Estrutural

A análise estrutural tem como finalidade testar a estrutura composta pelo painel–vigas e, em seguida, enviar o resultado da análise estática para o ficheiro **Energia.txt**, que corresponde à energia elástica de deformação,  $U$ , calculada pela análise estática efetuada no programa de elementos finitos (ANSYS).

## 3.3.3 Pós-Processamento

### 3.3.3.1 Energia Elástica de Deformação

Para se obter a função objetivo,  $f(\mathbf{X})$ , deste trabalho, pode-se utilizar, para sua determinação, duas opções, com base no programa de elementos finitos (ANSYS).

A primeira opção tem como base a utilização da energia elástica de deformação total da estrutura. Para se obter esta energia, basta correr o ficheiro **codigo.txt** no programa de elementos finitos, que está programado de maneira a utilizar o comando **ETABLE, SENE** para avaliar a energia elástica de deformação em cada elemento e, em seguida, aplicar o comando **SSUM** para efetuar o somatório de toda a energia obtida em todos os elementos. Assim, de forma direta como demonstra a equação 2.31, fica bastante simples determinar a função objetivo.

A segunda opção de procedimento para esta determinação, é através do recurso à utilização do vetor campo deslocamentos,  $\mathbf{u}$ , e do vetor de forças aplicado na estrutura,  $\mathbf{F}$ . Com estes, é então possível aplicar a equação 2.28 e obter também o desejado resultado.

A primeira opção foi o procedimento utilizado para a determinação da função objetivo deste projeto. Mais adiante, serão apresentados os respectivos resultados e avaliações do mesmo.

### 3.4 Algoritmos de Otimização Aplicados

Os algoritmos de otimização que foram implementados no programa computacional deste trabalho vão ser apresentados neste subcapítulo. Recordando o que já foi referido no início deste trabalho, foram utilizados dois tipos de variáveis de projeto para o estudo do melhor reforço a aplicar no painel. Para tal, houve uma primeira análise utilizando variáveis do tipo contínuo, e outra utilizando variáveis do tipo discreto, em que se aplicou a cada tipo um diferente algoritmo de otimização. Para otimizar o primeiro caso de estudo em otimização contínua, foi aplicado um algoritmo meta-heurístico designado como **pattern search**, que está disponível na *toolbox* de otimização do programa MATLAB (*Global Optimization Toolbox*). No caso de estudo em otimização com variáveis discretas, foi utilizado um método recente para a otimização global, denominado de **GLODS** (*global and local optimization using direct search*) [63].

Seguidamente, são apresentados os algoritmos aplicados em cada tipo de otimização de forma mais detalhada.

#### 3.4.1 *Pattern Search*

*Pattern search* é um método de otimização que foi proposto por Hooke e Jeeves em 1961, e funciona de tal maneira que, a partir de uma solução, a solução corrente, procura soluções melhores na sua vizinhança. O algoritmo pode ser descrito nos seguintes passos:

1. Uma solução é fornecida ou gerada e esta será a solução corrente;
2. Na vizinhança da solução corrente, novas soluções são simultaneamente geradas seguindo um dado padrão (daí o nome do método);
3. São calculados os valores das novas funções objetivo e seguidamente são comparados ao valor da função objetivo corrente;
4. Quando é encontrado um melhor valor de função objetivo, a sua correspondente solução passa a ser a solução corrente, e é indicado como a captação ocorreu com sucesso;
5. Ao ocorrer a última captação sem alterar a distância ( $\alpha$ ) entre as novas soluções e a solução corrente, esta distância diminui de forma a achar novas soluções (caso contrário, é aumentada).

O algoritmo repete os passos 2 a 5 até que algum critério de paragem seja atingido. Normalmente, estes critérios aplicados atendem às seguintes regras:

1. Número máximo de iterações atingidas sem melhoria da função objetivo;
2. A área para geração de novas soluções vizinhas é menor que um determinado valor previamente definido;

### 3. Tempo máximo de execução do algoritmo.

Através do MATLAB, é possível especificar o seu fator de contração da função que vai gerar as próximas soluções, no caso de haver captação, bem como o seu fator de expansão, no caso de não haver captação [75].

#### 3.4.2 GLODS

O GLODS é um novo algoritmo desenvolvido para *single optimization*, adequado para constrangimentos limitados, otimização global e livres de derivadas (como argumentado na secção 2.1). Usando a procura direta do tipo direcional, o método alterna entre *search step*, onde potencialmente boas regiões são localizadas, e *poll step*, onde são exploradas as regiões anteriormente localizadas. Esta exploração é feita através do lançamento de diversos métodos de pesquisa padrão (*pattern search*), um em cada uma das regiões de interesse. De forma diferente de uma estratégia *multistart*, os vários métodos de *pattern search* irão fundir-se quando estiverem suficientemente próximos uns dos outros. O objetivo do GLODS é acabar com o maior número *pattern search* ativos, como o número de mínimos locais, o que facilmente permitiria a localização do possível valor do extremo global [63].

## Capítulo 4

# Programas Implementados

Neste capítulo, serão apresentados e analisados os programas desenvolvidos para os dois tipos de otimização topológica implementados neste projeto. Ou seja, o primeiro programa tem como base variáveis do tipo contínuo e foi desenvolvido com recurso ao algoritmo de otimização Pattern Search. O segundo programa já foi elaborado com variáveis de tipo discreto e teve o GLODS como algoritmo de otimização. Nos subcapítulos seguintes, estão demonstrados e explicados todos os procedimentos utilizados para a criação dos programas.

### 4.1 Programa de Otimização Contínua

Neste subcapítulo, é apresentado de forma sucinta, a implementação do programa computacional de otimização contínua que ocorreu no início deste projeto. Também vai ser referido, de como surgiu a necessidade de alterar do programa de otimização contínuo para discreto.

No início, as variáveis de projeto implementadas no programa foram do tipo contínuas, onde as avaliações serviram de certo modo, para obter e avaliar as primeiras configurações de reforço, bem como a conexão formada entre os dois *softwares* utilizados.

Neste programa foi utilizado o *Pattern search* como algoritmo de otimização (já referido na secção 3.4), onde, antes do arranque da avaliação, o utilizador terá de acessar ao ficheiro MATLAB designado de **programa.m** e, definir os extremos superiores e inferiores das variáveis de projeto, através das variáveis, **LB**, para o desvio inferior e **UB**, para o desvio superior. Foram efetuadas várias avaliações de modo a compreender se o modelo estrutural estava bem formulado no código construído no ficheiro **codigo.txt**. Um exemplo de verificação do código, foi utilizar uma solução obtida na otimização de um esquema normal de 24 posições e, inverter todas as variáveis de projeto relativamente ao eixo de simetria. Em seguida verificar se as soluções obtidas são iguais, se forem o código está bem formulado. Foi o que aconteceu, como se verifica na figura 4.1, que retrata a otimização de 25% do maior reforço a ser aplicado em esquemas de 24 posições (corresponde a um volume de 6 vigas na sua máxima espessura), com carga  $F_1$  aplicada.

As imagens representadas na figura 4.1 foram obtidas no programa de elementos finitos, ANSYS.



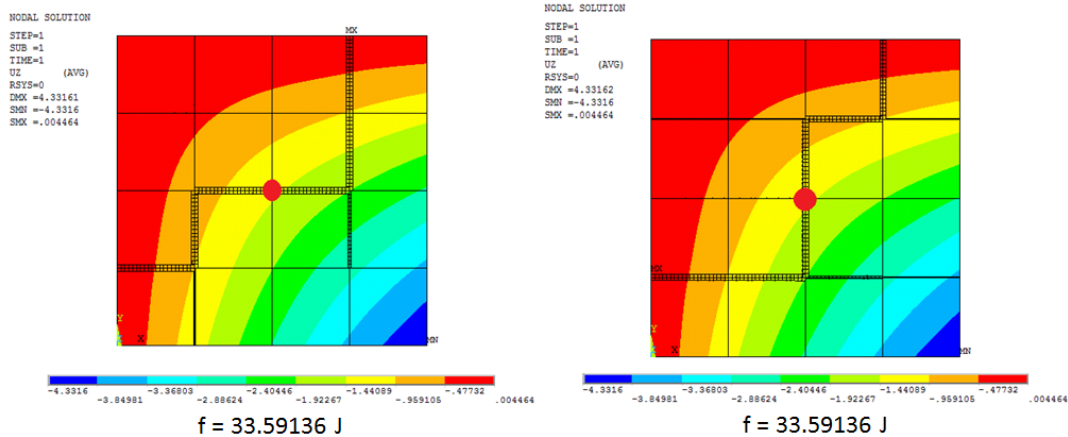


Figura 4.1: Duas soluções iguais, mas com variáveis invertidas relativamente ao eixo de simetria.

No decorrer deste processo de avaliações, 3 quantidades de material a aplicar como reforço foram admitidas, ou seja, foram aplicados, 25%, 50%, e 75% do volume máximo admitido em cada esquema. No entanto, surgiram algumas limitações neste tipo de otimização que iam em oposição à verdadeira necessidade da maioria dos casos recorrentes, que é reforçar um painel somente com um tipo de viga a aplicar. Por exemplo, se aplicar 25% do volume máximo num esquema de 24 posições, este corresponde a aplicar um reforço de 6 vigas com espessura máxima, o que não acontece neste caso, como se pode verificar na figura 4.1. Esta solução obteve uma configuração de reforço com 6 vigas perto da espessura máxima, uma viga a metade da espessura, outra viga delgada e, as restantes vigas muito perto do valor mínimo admissível, como demonstram os dados retirados do ficheiro **Constante.txt**, representados na figura 4.2.

t1=0.01679	t13=0.09687
t2=0.09492	t14=0.00117
t3=0.00898	t15=0.00117
t4=0.00117	t16=0.00117
t5=0.00117	t17=0.00117
t6=0.00117	t18=0.09882
t7=0.00117	t19=0.09882
t8=0.00117	t20=0.00312
t9=0.00507	t21=0.00117
t10=0.04609	t22=0.00117
t11=0.09882	t23=0.00117
t12=0.09882	t24=0.00117

Figura 4.2: Dados do ficheiro **Constante.txt** correspondentes a 25% de reforço máximo aplicado no esquema de 24 posições com carga  $F_1$ .

Para ter somente um tipo de viga a aplicar no painel, optou-se então por se modificar o tipo de otimização deste trabalho para variáveis de projeto do tipo discreto e alterou-se também o algoritmo de otimização para o GLODS, que tem como base, o *Pattern search* até aqui utilizado.

## 4.2 Programa de Otimização Discreta

A programação do modelo computacional de otimização dos reforços em painéis planos utilizando variáveis discretas foi o estudo mais relevante para o trabalho, pois, como já foi frisado anteriormente, existiu necessidade de utilizar este método para satisfazer a ideia inicial do projeto, que visa em utilizar somente uma dimensão de viga para reforçar o painel. Este problema em estudo teve então como objetivo otimizar reforços em painéis com recurso a vigas de secção quadrada, com espessuras laterais de 10 centímetros e 1,25 metros de comprimento (referido no tópico 2.3.3). Através do método de elementos finitos, em combinação com o algoritmo de otimização GLODS, foram analisadas várias combinações das configurações inicialmente projetadas, que se basearam na alteração das condições impostas às estruturas e nas forças aplicadas. Recordando as secções 3.3.1.4 e 3.3.2.3, pode-se visualizar as configurações implementadas (esquemas de 24 e 32 posições). Introduzindo as forças a atuar ( $F_1$ ,  $F_2$  e  $F_3$ ) e o número de vigas a colocar no reforço, pode-se dar início ao cálculo de otimização destes casos de estudo.

No decorrer deste trabalho, houve necessidade de dividir o projeto em distintos casos de estudo; no entanto, todos eles convergem para o mesmo método de otimização embora sigam uma geração de resultados que, individualmente, se vão adequando às suas distintas limitações. Para melhor compreensão dos vários casos em estudo e da maneira como estes foram organizados no decorrer do trabalho, recorreu-se à construção de diagramas que estão representados nas figuras 4.3 e 4.4.

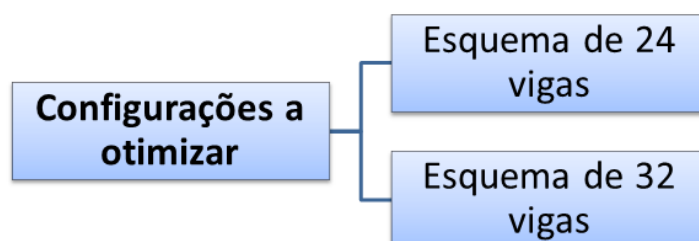


Figura 4.3: Diagrama dos esquemas a analisar

No diagrama da figura 4.3 está representado o principal conceito deste trabalho. Está dividido em dois subgrupos de avaliações; o primeiro é constituído pelo grupo de configurações que levou a cabo otimizar o reforço sem colocação de vigas na extremidade do painel (Esquema de 24 vigas), e o segundo é o grupo de configurações que permite vigas na extremidade do painel (Esquema de 32 vigas).

Com isto, podemos avançar para a explicação da forma como estão modelados estes esquemas em causa. Na figura 4.4 está representado o diagrama que retrata a organização das diversas análises que foram realizadas neste trabalho. Pode-se observar que este diagrama é representativo de ambos os esquemas (24 e 32 vigas), apresentados na figura 4.3, isto porque se procedeu ao mesmo método utilizado nos dois esquemas, podendo-se então afirmar que o diagrama corresponde a um método de organização geral para cada esquema.

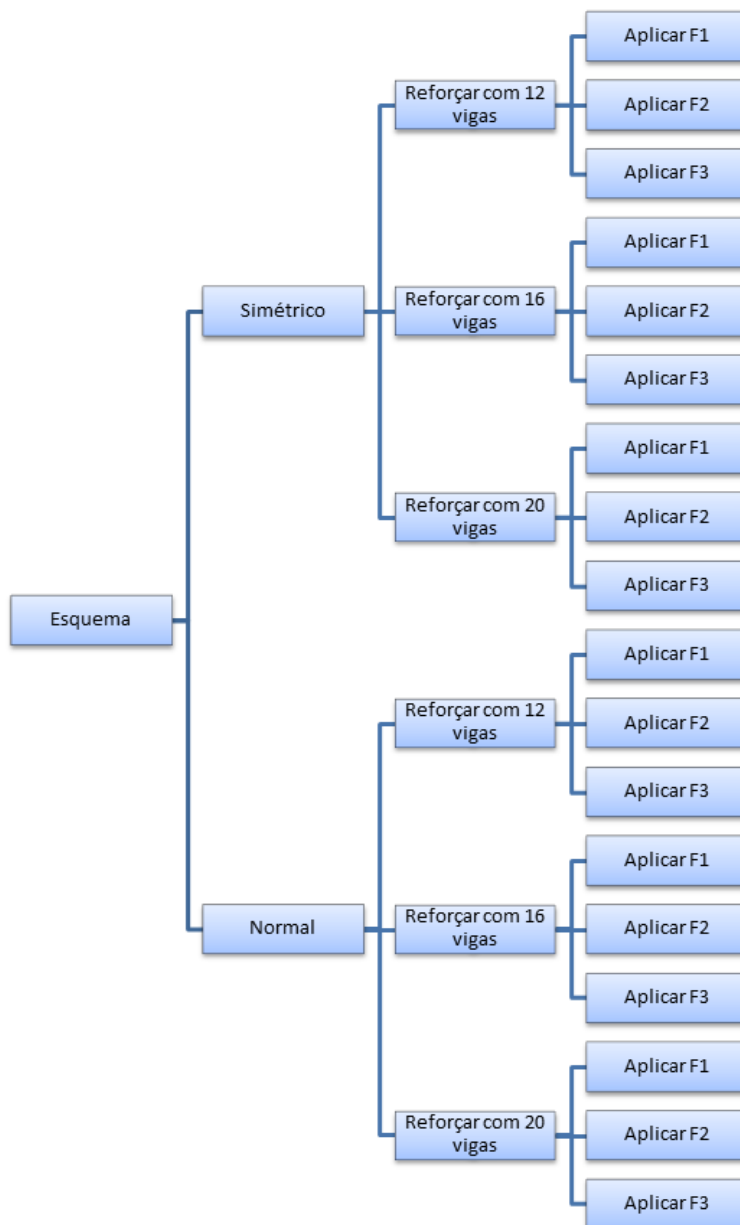


Figura 4.4: Diagrama representativo das análises efetuadas em cada esquema

Os esquemas podem levar ou não em conta a simetria do painel, sendo o utilizador a decidir se quer otimizar o reforço utilizando a via de esquema simétrico ou a via de esquema não simétrico (normal). Se escolher, por exemplo, o esquema simétrico de 24 vigas, o utilizador sabe que o programa vai funcionar com 12 variáveis e está restrito a ter configurações somente simétricas como resultado final. Ao escolher esquemas não simétricos, o programa funciona com 24 variáveis, o que leva a resultados que podem ou não obter configurações simétricas, dependendo também das restrições, das cargas aplicadas e do processo de paragem do algoritmo.

Depois do utilizador escolher se recorre ou não ao perfil simétrico, vai ter de decidir o número de vigas a colocar como reforço, bem como a posição e tipo de cargas a aplicar.

Neste estudo, para cada caso apresentado, foram selecionadas 3 quantidades de vigas a aplicar como reforço do painel, ou seja, podemos:

- aplicar 12 vigas;
- aplicar 16 vigas;
- ou aplicar 20 vigas.

Foram escolhidas estas 3 numerações com vista a ser possível realizá-las em qualquer um dos esquemas apresentados, isto para posteriormente se poder comparar os resultados obtidos nos esquemas implementados, verificar a adaptação destes às diferentes limitações introduzidas, bem como tirar conclusões da forma como se comportam os vários tipos de estrutura.

Em cada um dos 12 casos criados e descritos acima, foram aplicados 3 carregamentos em diferentes localizações. Estes carregamentos são compostos por cargas pontualmente aplicadas, em localizações conforme foram indicadas na secção 3.3.2.3. Com estas forças ( $F_1$ ,  $F_2$  e  $F_3$ ), introduzidas no plano de simetria de cada estrutura, ficamos com um projeto dividido em 36 casos de estudo, onde em cada um foi aplicado o mesmo modelo de otimização computacional.

O modelo de otimização computacional está preparado para efetuar a convergência da função objetivo através do ajuste das vigas ao longo do painel, de maneira a que, no final, seja obtida uma estrutura com *design* ótimo para o reforço. Este modelo foi obtido devido às inúmeras análises que são feitas em processos iterativos que atuam através do ciclo computacional constituído pela interação entre o MATLAB e o ANSYS.

Concluído o processo de otimização, o algoritmo GLODS deteta um conjunto de soluções que correspondem aos mínimos locais e, possivelmente, ao mínimo global da avaliação. Todo este processo de como o programa do modelo computacional está construído, vai ser tratado no tópico seguinte.

#### 4.2.1 Implementação do Programa de Otimização Discreta

Nesta secção, serão apresentados todos os procedimentos que foram desenvolvidos e implementados no programa computacional para obter e organizar as soluções deste projeto.

Para iniciar o programa, o utilizador tem de acessar ao ficheiro **Run\_mod.m**, que serve de controlador do programa otimizador. Este tem como base um ciclo *for*, e é através deste que o utilizador controla o número de vezes que o programa vai analisar o mesmo problema.

Foram corridos várias vezes os programas de cada caso em estudo de maneira a se obter várias soluções para a mesma experiência. Isto é possível devido à maneira como o GLODS funciona, ou seja, este explora o domínio através do lançamento de várias pesquisas diretas direcionais, uma em cada região de interesse. De forma diferente de uma estratégia *multistart* simples, as pesquisas diretas irão se fundir quando estiverem suficientemente perto umas das outras. O objetivo do GLODS é terminar com o maior número de pesquisas diretas, como o número de mínimos locais; quantas mais corridas forem efetuadas, mais facilmente é possível localizar o valor do mínimo global. [63]

Os parâmetros importantes do GLODS para este projeto estão representados no ficheiro **parameters\_gloids\_progressivo.m**, que são todos os parâmetros de inicialização do algoritmo otimizador. O utilizador tem possibilidade de alterar 3 valores. São eles:

a variável **list**, variando **0-4**:

- **0**, se o algoritmo inicializa a lista de pontos com um único ponto;
- **1**, se uma estratégia de amostragem hipercúbica é considerada para a inicialização;
- **2**, se é utilizada amostragem aleatória;
- **3**, se os pontos são considerados igualmente espaçados num segmento de reta, juntando-se as variáveis de limites superiores e inferiores, em conjunto com o ponto central;
- **4**, se o algoritmo é inicializado com uma lista fornecida pelo utilizador,

a variável ***user\_list\_size***, variando **0-1**:

- **0**, se o tamanho inicial é igual à dimensão do problema;
- **1**, se os conjuntos do utilizador são abaixo do tamanho da lista inicial de pontos,

e a variável ***nPini***, que corresponde ao número de pontos a serem considerados na inicialização.

No decorrer da análise de otimização do programa, vão sendo geradas aleatoriamente a variáveis de projeto, e muitas das vezes vão-se repetindo estas mesmas variáveis. Para que o programa otimizador não repita a análise em elementos finitos do mesmo vetor de variáveis de projeto, foi construído um ficheiro de memória. Este ficheiro tem a designação de ***CACHE.mat*** e está contido no ficheiro ***func-f***. A ***CACHE.mat*** guarda todos os vetores das variáveis de projeto (***CACHE.X***) e os correspondentes valores da função objetivo (***CACHE.OF***). Esta memória é muito importante para o programa computacional, pois este escusa de executar o programa de elementos finitos (ANSYS) várias vezes com o mesmo vetor, pois basta executar este vetor uma vez e jamais o volta a executar, ficando guardado em memória. Poupa-se muito tempo de cálculo e serve de informação para posteriores análises semelhantes, tal como se vai verificar mais adiante. Se for necessário, por algum motivo, excluir a informação contida na CACHE, basta o utilizador correr o ficheiro ***limpaCACHE.m***.

Quando o programa otimizador chega ao fim, os resultados são guardados de forma sequencial no ficheiro ***glods\_partial\_results.txt***, contendo todo o historial de mínimos locais e variáveis de projeto. Para se compreender a forma como são obtidos e tratados os resultados, foi criado o subcapítulo seguinte.

## 4.2.2 Obtenção e Tratamento de Resultados

Depois do programa avaliar todas as análises que o utilizador pretendeu "correr", estas são guardadas na diretoria de trabalho. Vários ficheiros ***glods\_partial\_results.txt*** representantes de cada análise são inseridos da seguinte forma:

***glods\_partial\_results.1.txt***

***glods\_partial\_results.2.txt***

***glods\_partial\_results.3.txt***

⋮

sendo 1, 2, 3, ..., n, os números representativos de cada avaliação.

Como estes ficheiros estão guardados de forma desordenada e, em alguns casos, com repetição de valores, foi criado um ficheiro conversor de posições que também tem a função de eliminar as soluções repetidas, ao qual foi chamado de **Conversor.m**. Os ficheiros **glods\_partial\_results.txt** são agora representados por ficheiros **Solução.txt** da mesma forma:

**Solução\_1.txt**

**Solução\_2.txt**

**Solução\_3.txt**

⋮

Este ficheiro, **Conversor.m**, não só tem as tarefas descritas, como também desenha e guarda a melhor solução obtida de cada avaliação na pasta **Resultados**. Para a criação destas figuras representativas da estrutura criada no programa de elementos finitos (ANSYS), foram criados uns códigos em ficheiros MATLAB que constroem também estas mesmas figuras. Estes códigos construtores são utilizados ao longo do tratamento de resultados e são denominados como **Figura1.m** e **Figura2.m**.

Analisando o trabalho em si, fez todo o sentido começar pelas avaliações das configurações simétricas, pois estas têm metade da variáveis de projeto das restantes, o que leva a um tempo de execução bastante mais rápido. Visto isto, houve também necessidade de tratar os resultados dos esquemas simétricos de maneira ligeiramente diferente do modo de tratamento dos esquemas normais (não simétricos).

#### 4.2.2.1 Esquemas Simétricos

Para o tratamento de resultados dos esquemas simétricos, foi construído um programa secundário que tem como finalidade selecionar o melhor e o pior grupo de soluções em cada esquema avaliado. Este programa, designado de **BestWorst.m**, tem capacidade de colocar as soluções correspondentes à melhor análise no ficheiro **Solução\_melhor\_Final.txt**, e as soluções da pior análise no ficheiro **Solução\_pior\_Final.txt**. Em seguida, converte as soluções em figuras representativas das estruturas alcançadas no ANSYS e são colocadas nas pastas **Confi\_boa** e **Confi\_ma**, respetivamente. Como exemplo, em seguida são apresentadas as soluções obtidas na primeira análise efetuada pelo programa (pasta designada de **12\_6\_F1**), que foi a otimização de um reforço constituído por 12 vigas (6 variáveis) em esquema simétrico de 24 posições (12 posições a metade do painel) e com a carga aplicada no centro do painel ( $F_1$ ).

Neste exemplo, as soluções da melhor avaliação foram obtidas e colocadas automaticamente na pasta **Confi\_boa** com as seguintes configurações (figura 4.5):

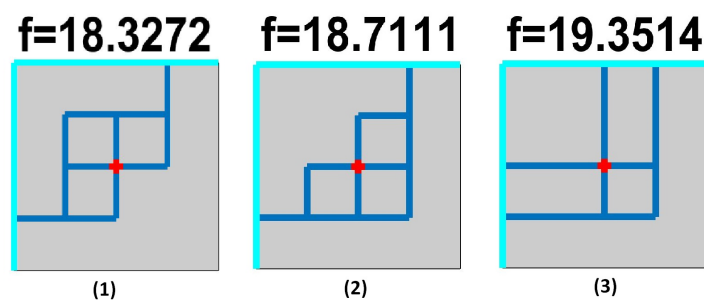


Figura 4.5: Esquemas representativos da melhor avaliação.

e na pasta **Confí.ma** as respectivas melhores soluções da pior avaliação (figura 4.6):

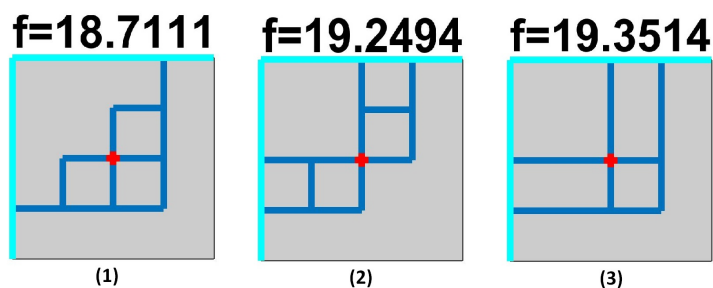


Figura 4.6: Esquemas representativos da pior avaliação.

Nas figuras, está representado, por cima de cada esquema, o valor da **energia elástica de deformação da estrutura,  $U$** , aqui representado com a variável  $f$  [N.m], valor que foi utilizado como função objetivo e que vai acompanhar os restantes resultados daqui em diante e, servindo assim como termo de comparação.

É de notar que nas soluções encontradas das figuras 4.5 e 4.6 existem esquemas iguais e com o mesmo valor de energia elástica de deformação,  $f$ . Isto acontece devido ao fato das avaliações encontrarem mínimos locais comuns entre elas, mas não é seguro que todas encontrem o mínimo global, como aconteceu neste mesmo exemplo. Portanto podemos desde já concluir que é importante efetuar várias análises ao mesmo modelo em estudo para ter a certeza que se vai obter a melhor configuração de reforço possível.

Depois de se concluir todas as avaliações aos esquemas simétricos, houve necessidade de transferir toda a informação contida em cada CACHE dos esquemas simétricos para a CACHE dos respectivos esquemas não simétricos. Com esta informação, as avaliações a fazer no programa de elementos finitos (ANSYS) são muito menores, porque apenas as estruturas não simétricas são avaliadas. Para esta transação de informação, foi construído um conversor de CACHE denominado de **Criar.nova.CACHE.m**, que funciona de forma a duplicar as variáveis de projeto de maneira a que as estruturas criadas em esquemas simétricos correspondam agora às estruturas em esquemas não simétricos.

#### 4.2.2.2 Esquemas Normais ( ou Não Simétricos)

Como já foi referido, a otimização de reforços utilizando como base esquemas não simétricos levou em conta a utilização da memória em CACHE utilizada nos esquemas simétricos. No entanto, também é importante começar a otimização destes problemas com as correspondentes soluções encontradas nos esquemas simétricos anteriormente avaliados. Para tal, foi necessário aceder ao ficheiro do GLODS *parameters\_glods\_progressivo.m*, e alterar os dois primeiros parâmetros para valor zero, como indicado na explicação do início da secção 4.2.1. Depois, basta aceder ao ficheiro *driver\_glods\_progressivo.m* e introduzir no vetor *x\_ini*, os valores correspondentes ao vetor da melhor solução do caso simétrico. Tendo isto, o programa de otimização com recurso a esquemas não simétricos está pronto a correr; para tal, basta arrancar com o ficheiro *Run\_mod.m*.

Neste programa, não se obtém pastas com os melhores e os piores grupos de soluções do esquema em avaliação, porque o programa está construído de maneira a verificar se a partir da melhor configuração alcançada nos esquemas simétricos, consegue melhorar para uma solução não simétrica. É um processo um pouco incongruente, porque se a estrutura é simétrica e está sujeita a restrições simétricas, só se deveria obter como melhores soluções os reforços simétricos, mas isso não acontece em alguns casos como podemos verificar no capítulo 5.

Quando a análise chega ao fim, o programa organiza e elimina as repetidas soluções e grava as principais soluções no ficheiro *Solução\_1.txt* também de forma organizada. Em seguida corre-se o ficheiro *Criar\_configuracoes.m*. Este pega nas soluções obtidas no *Solução\_1.txt*, cria e guarda as configurações correspondentes na pasta *Configurações Finais*. De notar que esta pasta é criada na mesma diretoria que os ficheiros de cada análise em causa, ou seja, todos os casos aqui estudados têm a sua própria pasta com todos os ficheiros com ele relacionados.

Como exemplo, são aqui apresentadas na fig 4.7 as soluções obtidas num reforço constituído por 12 vigas em esquema normal (não simétrico) de 24 posições e com carga aplicada no centro do painel ( $F_1$ ). Estas soluções encontram-se na pasta *Configurações Finais* que está inserida no problema *24.12.F1*.

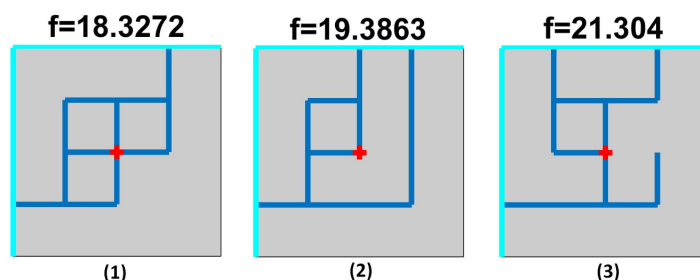


Figura 4.7: Soluções representativas de esquemas normais (não simétricos).

Como se pode visualizar, o programa não encontrou melhor solução que a constatada no programa apresentado anteriormente, no entanto, foram encontradas duas soluções não simétricas que mesmo sendo piores que as simétricas, dariam possivelmente um aceitável reforço.



## Capítulo 5

# Análise e Discussão de Resultados

No decorrer deste capítulo vão ser apresentados e avaliados todos os resultados relevantes para esta projeto. O capítulo vai estar repartido em três partes, que de certa forma retratam as fases importantes deste trabalho. A primeira parte é dedicada às configurações obtidas com esquemas de 24 posições, onde o processo de otimização ocorreu com recurso a 12 e 24 variáveis, a segunda parte, retrata as configurações referentes aos esquemas de 32 posições, em que o processo de otimização aconteceu com 16 e 32 variáveis e a ultima parte tem como finalidade a comparação de todos os esquemas avaliados, principalmente os que têm semelhantes restrições entre si, independentemente dos processos utilizados.

### 5.1 Resultados dos Esquemas com 24 Posições

Este subcapítulo é dedicado à primeira parte frisada anteriormente, onde serão apresentadas as melhores configurações em esquemas de 24 posições obtidas em diversos casos de estudo impostos ao painel durante este projeto.

A figura 5.1 representa de forma sucinta os vários casos implementados em esquemas de 24 posições onde estão apresentadas as cargas aplicadas no painel, no entanto, estas cargas foram utilizadas de forma isolada umas das outras, e estão aqui representadas em conjunto de forma a recordar as posições onde estas foram colocadas, tal como foi realçado na secção 3.3.2.3.

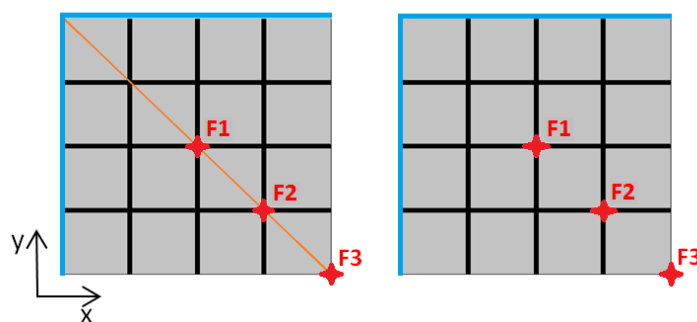


Figura 5.1: Esquema simétrico e esquema normal de 24 posições com cargas representadas.

Para melhor visualização e comparação dos resultados obtidos nos esquemas referidos acima, foi construída a tabela 5.1.



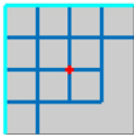


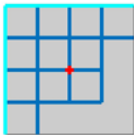
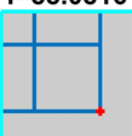

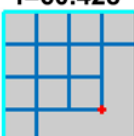


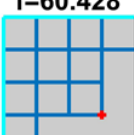
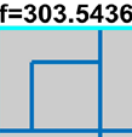
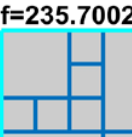

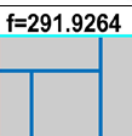
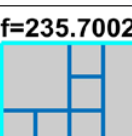

Carga	Esquema	N=12/24	N=16/24	N=20/24
<b>F<sub>1</sub></b>	Simétrico	f=18.3272 	f=14.5169 	f=13.0449 
	Normal	f=18.3272 	f=14.5169 	f=13.0449 
Carga	Esquema	N=12/24	N=16/24	N=20/24
<b>F<sub>2</sub></b>	Simétrico	f=88.0816 	f=67.4218 	f=60.428 
	Normal	f=88.0816 	f=67.4218 	f=60.428 
Carga	Esquema	N=12/24	N=16/24	N=20/24
<b>F<sub>3</sub></b>	Simétrico	f=303.5436 	f=235.7002 	f=198.5702 
	Normal	f=291.9264 	f=235.7002 	f=198.5702 

Tabela 5.1: Configurações das soluções dos esquemas de 24 vigas.

A tabela está dividida de forma a que cada carga aplicada ( $F_1$ ,  $F_2$  e  $F_3$ ), lhe é correspondido os diferentes tipos de configurações a que foi introduzida (esquema simétrico ou normal). Em seguida estes são-lhes estabelecidos os resultados correspondentes em cada tipo de estrutura aqui tratada, isto é, as melhores soluções obtidas nos três tipos de reforços estudados (**N**). O **N** representa a variável que corresponde ao número de vigas a introduzir em cada reforço, ou seja, se **N** = 12/24, corresponde a dizer que se quer reforçar o painel introduzindo 12 vigas num total de 24 posições disponíveis.

Avaliando os resultados visualizados na tabela 5.1, podemos concluir que;

- Para a carga  $F_1$ ;

- As melhores soluções encontradas são todas simétricas, mesmo utilizando esquemas normais,
  - Conforme se vai aumentando o número de vigas como reforço, este vai acrescentando vigas à estrutura mais fraca (**N = 12/24**) e por aí em diante, sem alterar o seu formato de reforço,
  - Como era de esperar, conforme se vai aumentando o número de vigas definidas como reforço, vai diminuindo a função objetivo, **f** [N.m].
  - Ao se passar de 16 para 20 vigas como reforço, o aumento da rigidez é pouco significativo, pois a estrutura já está bem reforçada e as novas vigas introduzidas são aplicadas em zonas com pouca deformação.
- Para a carga  $F_2$ ;
    - Também não se conseguiu obter melhorias das soluções dos esquemas simétricos com a implementação destes na otimização em esquemas normais,
    - O aumento de vigas no reforço não altera o tipo de estrutura que vem sendo obtido em reforços mais fracos, simplesmente acrescenta vigas às soluções anteriores,
    - Os valores da função objetivo aumentaram significativamente em comparação com os valores obtidos com carga  $F_1$ , isto acontece devido ao fato da carga  $F_2$  estar mais afastada da zona onde os apoios estão em contacto com o painel,
- Para a carga  $F_3$ ;
    - Existiu uma melhoria da função objetivo quando foram analisadas as configurações normais, nomeadamente na utilização do reforço com 12 vigas. Esta melhoria é bastante significativa, contudo, surge uma questão: Porque é que se obteve uma solução não simétrica? Isto acontece devido ao fato das posições das vigas estarem limitadas aos esquemas apresentados e ser otimização discreta. Se não houvesse restrição no local onde se pode colocar vigas, a melhor solução para este volume de reforço era sempre uma estrutura simétrica, no entanto, como o local onde estas são aplicadas está limitado a certas posições, o reforço converge para uma solução não simétrica.
    - É de notar que olhando para as soluções de 12 vigas podemos prever que, se não houvesse limitações quanto à posição das vigas, a solução ideal para este número de vigas deveria ser uma configuração parecida à combinação destas duas soluções obtidas neste caso.
    - Com o aumento de reforço de 16 para 20 vigas, a configuração da estrutura sofreu poucas alterações, aumentando só o reforço. Isto também está relacionado com a limitação das posições onde se coloca as vigas. Ou seja, as vigas estão impedidas de serem colocadas no ponto onde é aplicado o carregamento (esquema de 24 posições).

## 5.2 Resultados dos Esquemas com 32 Posições

Este subcapítulo vai-se destinar à segunda parte da análise de resultados e, retrata as soluções encontradas pelo programa otimizador utilizando esquemas de 32 posições onde é possível a colocação de vigas.

Como na apresentação de resultados anteriores, aqui também foi apresentado um esquema demonstrativo dos casos estudados. O esquema está representado na figura 5.2 e de forma resumida mostra os esquemas de 32 posições com os três tipos de carregamentos aplicados no painel.

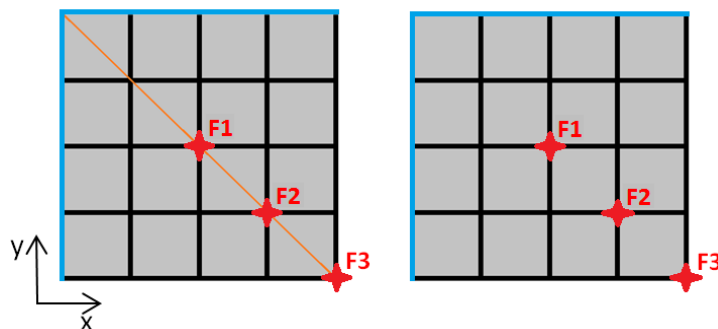


Figura 5.2: Esquema simétrico e esquema normal de 32 posições com cargas representadas.

Para cada um dos casos representados na figura 5.2, foi obtido com sucesso a otimização da função objetivo e a criação da correspondente estrutura de reforço. Para melhor apresentação das mesmas, são igualmente inseridas na tabela 5.2 como aconteceu nos casos anteriores.

A tabela 5.2 está organizada do mesmo modo que a exposta no subcapítulo anterior, para assim ser possível constatar de forma clara todas as semelhanças e divergências existentes entre ambas.

Examinando as soluções destacadas na tabela 5.2 pode-se concluir que:

- Para a carga  $F_1$ ;
  - Todas as soluções encontradas em ambos os esquemas (simétricos e normais) são simétricas,
  - Pode-se observar que, com recurso a poucas vigas como reforço (12 e 16), as soluções têm configurações mais centralizadas e juntas à carga aplicada, no entanto com o aumento do reforço para 20 vigas, este já utiliza vigas no bordo do painel como reforço a implementar,
  - À semelhança dos casos avaliados com esquemas de 24 posições, ao se aumentar 4 vigas no reforço ( $N=16/2$  para  $N=20/32$ ), a redução da função objetivo não é significativa. Isto deve-se à localização da carga  $F_1$ , das 3 cargas aplicadas esta é a que se situa numa posição menos afetiva ao painel porque está relativamente mais perto dos apoios, logo não é necessário aplicar grande quantidade de reforço para se obter bons resultados.
- Para a carga  $F_2$ ;
  - Obteve-se melhoria da função objetivo quando se passou à análise em esquemas normais. Aconteceu em  $N=12/32$  e  $N=16/32$ , alterando um pouco as estruturas. A melhoria é muito

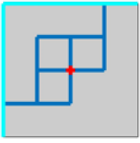
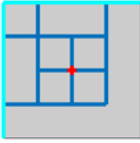

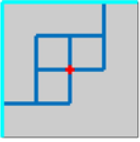
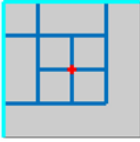



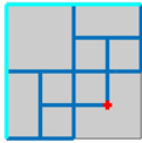
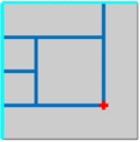
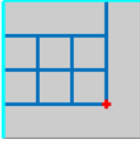
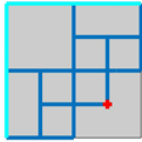





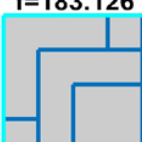
Carga	Esquema	N=12/32	N=16/32	N=20/32
<b>F<sub>1</sub></b>	Simétrico	f=18.3272 	f=14.5169 	f=12.8207 
	Normal	f=18.3272 	f=14.5169 	f=12.8207 
Carga	Esquema	N=12/32	N=16/32	N=20/32
<b>F<sub>2</sub></b>	Simétrico	f=88.0815 	f=67.4218 	f=58.338 
	Normal	f=87.9636 	f=67.2931 	f=58.338 
Carga	Esquema	N=12/32	N=16/32	N=20/32
<b>F<sub>3</sub></b>	Simétrico	f=299.7511 	f=222.9399 	f=183.126 
	Normal	f=299.7511 	f=222.9399 	f=183.126 

Tabela 5.2: Configurações das soluções dos esquemas de 32 vigas.

reduzida e possivelmente não faria muito sentido a aplicação de um reforço não simétrico nestas condições de projeto, mas é um fato interessante que acontece,

- Para a aplicação de N=20/32, a estrutura já não sofreu alterações,
- A colocação de vigas no bordo do painel só foram utilizadas no caso do reforço mais robusto (N=20/32), isto acontece devido ao fato desta carga estar colocada numa posição em que se pode considerar como "instável" relativamente às posições estabelecidas para colocar os reforços. Portanto quando se tem poucas vigas como reforço, são criadas configurações sem recorrer à utilização das posições laterais, e quando se aumenta o número de vigas como reforço, este diverge para outro formato de estrutura utilizando então as posições disponíveis

no bordo do painel.

- Para a carga  $F_3$ ;
  - Não houve melhorias da função objetivo com a reavaliação em esquemas normais, logo as melhores soluções continuam a ser simétricas,
  - À medida que se vão aumentando o número de vigas no reforço, este vai alterando o seu formato de estrutura, isto deve-se ao fato de o reforço principal estar no bordo do painel. Uma vez este preenchido, as restantes vigas são colocadas em outras posições de modo a minimizar ao máximo a função objetivo,
  - A estrutura otimizada com recurso a 20 vigas ( $N=20/32$ ), é uma combinação das estruturas de 12 e 16 vigas, concluindo-se que esta configuração de posições está bem adaptada para otimizar o reforço do painel sujeito a este tipo de carregamento.

### 5.2.1 Relação Energia/Peso da Estrutura

De modo a ter mais um termo de comparação e validação dos resultados foi calculada a relação **Energia/Peso** de todas as melhores soluções encontradas em cada caso avaliado. Esta relação é muito importante se a estrutura constituída pelo painel e reforço levar em conta o seu peso. Isto é, a estrutura pode ser colocada em funções que o seu baixo peso seja um requisito de projeto, como é o caso da industria de veículos.

Para efetuar o calculo da relação peso/energia foram necessários os seguintes dados;

Massa volúmica (aço) =  $7860 \text{ Kg/m}^3$ ,

Volume da viga:

$$V_{viga} = lado * lado * comprimento = 0.1 * 0.1 * 1.25 = 0.0125 \text{ cm}^3 \quad (5.1)$$

Peso da viga:

$$Peso_{viga} = Massa \text{ volumica} * V_{viga} = 7860 * 0.0125 = 98.25 \text{ Kg} \quad (5.2)$$

Por fim é calculada a relação **Energia/Peso** para cada caso de estrutura otimizada, ou seja, para implementação de 12, 16 ou 20 vigas como reforço do painel. Este calculo é todo efetuado do mesmo modo que o próximo exemplo, que se refere ao esquema de 24 posições com força aplicada no centro do painel,  $F_1$ :

$$\left( \frac{f}{P} \right)_{12 \text{ vigas}} = \frac{f}{12 * Peso_{viga}} = \frac{18.3272}{12 * 98.25} = 0.0155 \text{ J/kg} \quad (5.3)$$

$$\left( \frac{f}{P} \right)_{16 \text{ vigas}} = \frac{f}{16 * Peso_{viga}} = \frac{14.5169}{16 * 98.25} = 0.0092 \text{ J/kg} \quad (5.4)$$

$$\left(\frac{f}{P}\right)_{20 \text{ vigas}} = \frac{f}{20 * \text{Peso}_{\text{viga}}} = \frac{13.0449}{20 * 98.25} = 0.0067 \text{ J/kg} \quad (5.5)$$

Todos estes cálculos são apresentados de forma organizada na tabela 5.3 que está apresentada em seguida:

Configurações de 24 posições				
Carga	Esquema	$(f/P)_{12 \text{ vigas}}$	$(f/P)_{16 \text{ vigas}}$	$(f/P)_{20 \text{ vigas}}$
F1	Simétrico	0.0155 J/kg	0.0092 J/kg	0.0067 J/kg
	Normal	0.0155 J/kg	0.0092 J/kg	0.0067 J/kg
F2	Simétrico	0.0751 J/kg	0.0429 J/kg	0.0307 J/kg
	Normal	0.0751 J/kg	0.0429 J/kg	0.0307 J/kg
F3	Simétrico	0.2574 J/kg	0.1499 J/kg	0.1011 J/kg
	Normal	0.2476 J/kg	0.1499 J/kg	0.1011 J/kg
Configurações de 32 posições				
Carga	Esquema	$(f/P)_{12 \text{ vigas}}$	$(f/P)_{16 \text{ vigas}}$	$(f/P)_{20 \text{ vigas}}$
F1	Simétrico	0.0155 J/kg	0.0092 J/kg	0.0065 J/kg
	Normal	0.0155 J/kg	0.0092 J/kg	0.0065 J/kg
F2	Simétrico	0.0747 J/kg	0.0429 J/kg	0.0297 J/kg
	Normal	0.0746 J/kg	0.0428 J/kg	0.0297 J/kg
F3	Simétrico	0.2542 J/kg	0.1418 J/kg	0.0932 J/kg
	Normal	0.2542 J/kg	0.1418 J/kg	0.0932 J/kg

Tabela 5.3: Tabela de resultados da relação Energia/Peso.

A relação Energia/Peso é um parâmetro com alguma importância na classificação deste tipo de reforços estudados, nomeadamente para verificar de forma mais explícita a eficiência do reforço aplicado. Nos vários exemplos de reforço do painel, sujeitos às mesmas forças e condições de fronteira aplicadas, consegue-se por exemplo, verificar através deste parâmetro, qual reforço tem as melhores características referentes aos requisitos que foram mencionados anteriormente, ou seja, baixo peso estrutural para máxima rigidez. Pegando no primeiro caso apresentado,  $(f/P)_{12 \text{ vigas}} = 0.0155 \text{ J/Kg}$ , isto corresponde a dizer que por cada quilo do reforço aplicado neste painel, corresponde a 0.0155 Joules de energia elástica de deformação. Assim sendo, é desejável que a relação Energia/Peso tenha o valor mais baixo possível para que a estrutura painel-vigas apresente as melhores características.

Avaliando os resultados obtidos pela relação Energia/Peso apresentados na tabela 5.3, pode-se concluir que;

- A melhor estrutura conseguida pelo programa computacional otimizador, aconteceu em ambos os esquemas de 32 posições (simétrico e normal), com carga  $F_1$ , e com reforço constituído por 20 vigas. Faz todo o sentido ser este caso, porque as 32 posições das vigas neste caso em estudo, estão bem distribuídas em torno do ponto onde a carga foi aplicada, e juntando a isto, o reforço aqui aplicado corresponde ao máximo estabelecido para este projeto.
- A pior estrutura obtida pelo programa computacional otimizador, foi no esquema simétrico de 24

posições, com carga  $F_3$  aplicada e, reforço constituído por 12 vigas. Ao contrário do que foi dito no ponto anterior, aqui a carga é aplicada no vértice oposto às faces em contacto com os apoios. Nesta configuração é impossível aplicar reforços neste ponto (esquema de 24 posições) e, em conjunto com o mínimo de vigas aqui estudadas, faz com que este caso corresponda à pior estrutura aqui desenvolvida.

- O cálculo deste parâmetro, ainda fazia mais sentido, se fosse obtido com soluções de otimização em variáveis contínuas, no entanto, avaliando as soluções aqui demonstradas pela tabela 5.3, conseguimos verificar de forma clara, quais são os casos que deve ser introduzido, mantido ou retirado reforço. Por exemplo, olhando para os valores correspondentes à carga aplicada,  $F_1$ , pode-se concluir que o reforço de 16 vigas já é satisfatório, pois tem um valor bastante reduzido, e está na mesma ordem de grandeza que o reforço de 20 vigas, logo não é necessário a introdução de mais 4 vigas na estrutura, poupando um aumento de 393 kg ao seu peso. Um exemplo contrário, é o da carga aplicada  $F_3$ , em que somente com a aplicação de 20 vigas como reforço, o valor da relação Energia/Peso começa a dar valores mais satisfatórios, no entanto, é um caso em que se podia aumentar ainda mais o reforço. Isto porque, a força aplicada está colocada no pior sítio do painel (ponto mais afastado dos apoios) o que faz este caso ter os piores resultados alcançados no trabalho.

### 5.3 Comparação de Resultados Entre os Esquemas de 24 e 32 Posições

- Para a carga  $F_1$ ;
  - Conclui-se que entre os dois esquemas só existiram diferenças de resultados na otimização com recurso a 20 vigas de reforço. Isto porque a carga  $F_1$  está aplicada no centro do painel, o que leva o programa de 32 posições só a recorrer das posições do bordo do painel quando no centro já está em parte preenchido,
  - A melhoria encontrada no esquema de 32 posições não é significativa, no entanto é uma melhor solução e apresenta um formato de estrutura mais disperso, o que pode beneficiar em outras características da estrutura aqui não estudadas, como modos de vibração, etc.,
- Para a carga  $F_2$ ;
  - Notou-se uma diferença ocorrida na otimização em que foram utilizadas 20 vigas como reforço, onde se obteve uma melhoria com a implementação das posições no bordo do painel. Este criou uma estrutura também simétrica, mas com um ligeiro ganho na função objetivo.
  - Outra diferença aconteceu na otimização em esquemas normais de 12 e 16 vigas de reforço, ou seja, nos esquemas normais de 24 posições não houve alterações em comparação com



esquemas simétricos, mas em esquemas normais de 32 posições foi obtido uma ligeira melhoria. Isto tem a ver com a localização dos mínimos locais obtidos na avaliação da estrutura e também com o número de variáveis de projeto, pois ao avaliar-se a mesma análise nos dois esquemas (24 e 32 posições), numa única avaliação, o esquema de 32 variáveis melhorou, enquanto o de 24 se manteve igual. Para ser possível atingir os mesmos resultados nos esquemas de 24 variáveis, deveria repetir-se a avaliação computacional mais algumas vezes.

- Para a carga  $F_3$ ;
  - Foi o único caso em que todas as soluções obtidas nos esquemas de 32 posições deram diferentes soluções dos esquemas de 24 posições. É fácil de perceber o porquê deste acontecimento, como já foi dito, isto ocorreu devido ao fato da carga estar aplicada num vértice do painel, onde os esquemas de 24 vigas não têm acesso à colocação das mesmas.
  - Em quase todas as avaliações em esquemas de 32 posições ocorreu melhoria da função objetivo face às análises efetuadas nos esquemas de 24 variáveis, exceto no esquema normal de  $N=12/24$  em que a solução obtida não foi vencida. Tal como aconteceu no exemplo anterior (em carga  $F_2$ ), aqui também se deveria efetuar mais algumas avaliações, e possivelmente podia ser encontrada a mesma solução em esquema normal de 32 variáveis.

De notar que todos os resultados não simétricos aqui apresentados têm exatamente a mesma solução de função objetivo na configuração de reforço invertida relativamente ao eixo de simetria do painel, tal como aconteceu na programação em variáveis do tipo contínuo (ver exemplo representado na figura 4.1). Como exemplo é apresentado na figura 5.3 as duas soluções para  $N=12/24$  em esquemas normais com carga  $F_3$  aplicada.

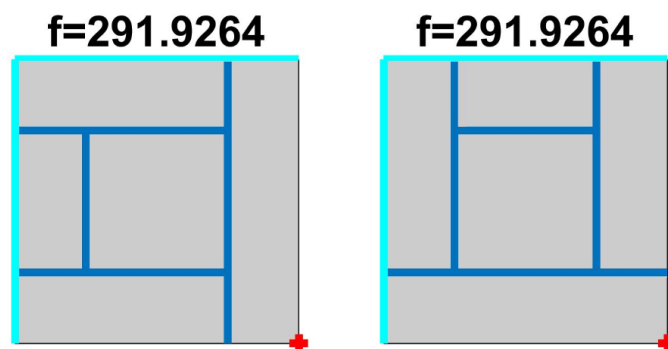


Figura 5.3: Soluções com mesma função objetivo mas com configurações invertidas relativamente ao eixo de simetria do painel.

## Capítulo 6

# Conclusões e Trabalho Futuro

### 6.1 Conclusões

Nesta tese pretendia-se desenvolver um modelo computacional de otimização de topologia discreta de reforços em painéis planos com o objetivo de maximizar a rigidez da estrutura, sujeita a diferentes condições de fronteira e carregamentos. O problema de otimização foi formulado pelo algoritmo, GLODS, que através da sua técnica, explora toda a região admissível do domínio, localizando subdomínios promissores, que são avaliados consoante a escolha dos parâmetros iniciais. A parte de análise estrutural foi efetuada pelo programa de elementos finitos, ANSYS, dedicado ao cálculo da função objetivo. O problema foi implementado e testado em painéis de forma quadrada, em que pelos resultados obtidos, se pode afirmar que o objetivo foi atingido para a maioria dos casos analisados.

Inicialmente testaram-se alguns exemplos onde a solução ótima era previsível, de maneira a testar o algoritmo implementado, bem como o modelo estrutural. Posteriormente, foram testados os exemplos mais complexos que seguiram todos os objetivos do trabalho.

Um ponto interessante, não foi frisado anteriormente, foi a diferença existente nas soluções obtidas antes e depois da adição dos *offsets* na estrutura (descritos na secção 3.3.2.1), onde se concluiu que houve um aumento de rigidez nas estruturas, com a implementação deste deslocamento no código APDL.

No decorrer do trabalho, houve necessidade de dividir o projeto em dois subgrupos distintos que convergiram para o mesmo método de otimização, embora sigam uma geração de resultados que, individualmente, se adequaram às suas diferentes limitações. O primeiro foi constituído pelo grupo de configurações que otimizou os reforços sem ter em conta a colocação de vigas na extremidade do painel (esquema de 24 vigas), e o segundo foi o grupo de configurações que permite vigas na sua extremidade (esquema de 32 vigas).

Dos resultados obtidos, verificou-se que os programas desenvolvidos para os esquemas de 32 posições foram em geral mais eficientes que os mesmos em esquemas de 24 posições. No entanto, o tempo de cálculo dispendido é em média superior ( $Tempo\_dispendido_{(24)} \simeq 50000segundos = 13.8horas$  e  $Tempo\_dispendido_{(32)} \simeq 90000segundos = 25horas$ ). Contudo, para as cargas aplicadas F1 e F2, o

programa correspondente ao esquema de 24 posições tem alguma vantagem quando comparado com o de 32 posições, pois a maioria das soluções são iguais e, nos casos em que as soluções deram resultados diferentes, não se observa uma perda significativa. Concluiu-se com isto, que levando em conta o tempo de processamento, os casos em que as cargas são aplicadas em zonas mais perto do centro do painel, é aconselhável que a otimização se efetue com o programa otimizador discreto de 24 posições.

Ao se avaliar o grupo de melhores soluções obtidas em cada avaliação (soluções que podem ser vistas na pasta **Confi.boa** de cada caso em estudo, referida na secção 4.2.2.1), podemos concluir que, as soluções seguintes à melhor solução têm resultados tão bons como esta, mas por vezes com configurações de reforços visivelmente bastante diferentes, o que pode conduzir a uma melhor adaptação por parte destas configurações a alguns projetos onde os painéis poderão ser aplicados. Outra vantagem em algumas destas soluções, é a simplificação da construção do reforço. Por exemplo, se se observar as configurações da figura 4.5, pode-se concluir que a construção do reforço da imagem nº 3 tem vantagem na sua construção comparativamente à construção do reforço da melhor solução (imagem nº 1), ou seja, enquanto na melhor solução se tem de cortar uma viga em 7 partes para a construção do reforço, nesta solução (imagem nº 3) somente se tem de cortar a viga em 5 partes para a conceção do mesmo tipo de reforço.

Olhando de uma forma geral para todas as soluções obtidas nas tabelas 5.1, 5.2 e 5.3, pode-se concluir que, tendo só como ponto de partida a maximização da rigidez e a minimização da relação Energia/Peso da estrutura, o melhor programa a ser utilizado é o programa otimizador discreto de 32 posições.

## 6.2 Trabalho Futuro

No decorrer do desenvolvimento deste projeto foram surgindo algumas ideias para trabalhos futuros que também têm como contexto, o desenvolvimento de modelos computacionais que otimizem reforços em painéis planos tendo como recurso vigas, entre os quais se podem salientar os seguintes;

- Propor novas dimensões e formatos para o painel,
- Aplicar novas quantidades de reforço, bem como a introdução de número ímpar de vigas a utilizar, no programa otimizador de esquemas normais (não simétricos de 24 e 32 variáveis),
- Propor novas cargas a aplicar no painel, em localizações fora do eixo de simetria da estrutura, e efetuar análises somente com o programa otimizador de esquemas normais,
- Propor novas condições de fronteira e efetuar também as análises com o programa de esquemas normais,
- Aperfeiçoar o programa de otimização contínua e aplicar os exemplos dados nos tópicos anteriores,

- Aplicar vários carregamentos em simultâneo, com diferentes direções e sentidos, e/ou cargas distribuídas.

# Bibliografia

- [1] Dictionary.com. (2016). The definition of anisotropic. [online] Available at: <http://www.dictionary.com/browse/anisotropic> [Accessed 26 Apr. 2016].
- [2] TheFreeDictionary.com. (2016). Batch mode. [online] Available at: <http://encyclopedia2.thefreedictionary.com/Batch+mode> [Accessed 29 Apr. 2016].
- [3] Priberam.pt. (2016). Significado/definição de cisalhamento no dicionário priberam da língua portuguesa. [online] Available at: <http://www.priberam.pt/dlpo/cisalhamento> [Accessed 27 Apr. 2016].
- [4] Dicionário E-Civil. (2016). O que é coeficiente de poisson? [online] Available at: <http://www.ecivilnet.com/dicionario/o-que-e-coeficiente-de-poisson.html> [Accessed 27 Apr. 2016].
- [5] Pt.wikipedia.org. (2016). Coeficiente de poisson. [online] Available at: [https://pt.wikipedia.org/wiki/Coeficiente\\_de\\_Poisson](https://pt.wikipedia.org/wiki/Coeficiente_de_Poisson) [Accessed 27 Apr. 2016].
- [6] Dictionary.com. (2016). The definition of compliance. [online] Available at: <http://www.dictionary.com/browse/compliance> [Accessed 28 Apr. 2016].
- [7] Wikipedia. (2015). Compliance. [online] Available at: <https://en.wikipedia.org/wiki/Compliance> [Accessed 28 Apr. 2016].
- [8] Codecogs.com. (2016). Elastic constants – materials – engineering reference with worked examples. [online] Available at: <http://www.codecogs.com/library/engineering/materials/elastic-constants.php>.
- [9] Wikipedia. (2016). Deflection (engineering). [online] Available at: [https://en.wikipedia.org/wiki/Deflection\\_\(engineering\)](https://en.wikipedia.org/wiki/Deflection_(engineering)) [Accessed 28 Apr. 2016].
- [10] Dictionary.com. (2002). The definition of strain. [online] Available at: <http://www.dictionary.com/browse/strain?s=t> [Accessed 27 Apr. 2016].
- [11] Mineral Physics. (2015). Tensors, stress, strain, elasticity. [online] Available at: [http://serc.carleton.edu/NAGTWorkshops/mineralogy/mineral\\_physics/tensors.html](http://serc.carleton.edu/NAGTWorkshops/mineralogy/mineral_physics/tensors.html) [Accessed 27 Apr. 2016].
- [12] Wikipedia. (2016). Deformation (mechanics). [online] Available at: [https://en.wikipedia.org/wiki/Deformation\\_\(mechanics\)](https://en.wikipedia.org/wiki/Deformation_(mechanics)) [Accessed 27 Apr. 2016].

- [13] Wikipedia. (2016). Kronecker delta. [online] Available at: [https://en.wikipedia.org/wiki/Kronecker\\_delta](https://en.wikipedia.org/wiki/Kronecker_delta) [Accessed 27 Apr. 2016].
- [14] Wikipedia. (2015). Linear elasticity. [online] Available at: [https://en.wikipedia.org/wiki/Linear\\_elasticity](https://en.wikipedia.org/wiki/Linear_elasticity) [Accessed 28 Apr. 2016].
- [15] (2011). Parreira. Material didático/stt 5860. [online] Stt.eesc.usp.br. Available at: <http://www.stt.eesc.usp.br/index.php/material-didatico/category/9-parreira> [Accessed 28 Apr. 2016].
- [16] Download.autodesk.com. (n.d.). Nodes and elements. [online] Available at: [http://download.autodesk.com/us/algor/userguides/mergedProjects/Getting\\_Started/introduction\\_to\\_algor/Nodes\\_and\\_Elements.htm](http://download.autodesk.com/us/algor/userguides/mergedProjects/Getting_Started/introduction_to_algor/Nodes_and_Elements.htm) [Accessed 28 Apr. 2016].
- [17] Engineering dictionary.org. (2008). Elastic energy definition — engineering dictionary. [online] Available at: [http://www.engineering-dictionary.org/elastic\\_energy](http://www.engineering-dictionary.org/elastic_energy) [Accessed 29 Apr. 2016].
- [18] Dictionary.com. (2016). The definition of statics. [online] Available at: <http://www.dictionary.com/browse/statics> [Accessed 26 Apr. 2016].
- [19] Wikipedia. (2016). Heuristic (computer science). [online] Available at: [https://en.wikipedia.org/wiki/Heuristic\\_\(computer\\_science\)](https://en.wikipedia.org/wiki/Heuristic_(computer_science)) [Accessed 29 Apr. 2016].
- [20] Dictionary.com. (2016). The definition of isotropic. [online] Available at: <http://www.dictionary.com/browse/isotropic> [Accessed 26 Apr. 2016].
- [21] Engineeringtoolbox.com. (2016). Hooke's law. [online] Available at: [http://www.engineeringtoolbox.com/hookes-law-force-spring-constant-d\\_1853.html](http://www.engineeringtoolbox.com/hookes-law-force-spring-constant-d_1853.html) [Accessed 27 Apr. 2016].
- [22] Wikipedia. (2016). Hooke's law. [online] Available at: <https://en.wikipedia.org/wiki/Hooke> [Accessed 27 Apr. 2016].
- [23] Wiki.csiamerica.com. (2016). Meshing - technical knowledge base - computers and structures, inc. - technical knowledge base. [online] Available at: <https://wiki.csiamerica.com/display/kb/Meshing> [Accessed 29 Apr. 2016].
- [24] Pt.wikipedia.org. (2015). Matriz diagonal. [online] Available at: [https://pt.wikipedia.org/wiki/Matriz\\_diagonal](https://pt.wikipedia.org/wiki/Matriz_diagonal) [Accessed 27 Apr. 2016].
- [25] Pt.wikipedia.org. (2015). Matriz identidade. [online] Available at: [https://pt.wikipedia.org/wiki/Matriz\\_identidade](https://pt.wikipedia.org/wiki/Matriz_identidade) [Accessed 27 Apr. 2016].
- [26] Wu W. Pardalos P. Du, D. and N. (2011). Sohaee. Feasible direction method. [abstract]. wiley encyclopedia of operations research and management science. [online] Available at: <http://doi.org/10.1002/9780470400531.eorms0320>.

- [27] P. Ch. 10 – Nonlinear Programming Methods – Algorithms for Constrained Optimization. (2003). In Jansen and J. (2003). Bard. Supplements - operations research models and methods. [online] Me.utexas.edu. Available at: [https://www.me.utexas.edu/jansen/ORMM/supplements/pdf\\_supplements.html](https://www.me.utexas.edu/jansen/ORMM/supplements/pdf_supplements.html) [Accessed 26 Apr. 2016].
- [28] X. Chen and M. (2000). Kostreva. 14 - methods of feasible directions: a review. In: X. Yang, A. Mees, M. Fisher and L. Jennings, ed., Progress in Optimization, 1st ed. Kluwer Academic Publishers, pp.205-219.
- [29] Wceng fea.com. (2008). Finite element analysis engineering ltd. [online] Available at: <http://www.wceng-fea.com/FEM.htm> [Accessed 28 Apr. 2016].
- [30] Belhachmi Z. Allaire, G. and F. (2004). Jouve. Eurf.pdf – the homogenization method for topology and shape optimization – single and multiple loads case. in: Index of / jouve/papers. [online] Cmap.polytechnique.fr. Available at: <http://www.cmap.polytechnique.fr/jouve/papers/> [Accessed 26 Apr. 2016].
- [31] K. Suzuki and N. (1991). Kikuchi. A homogenization method for shape and topology optimization. Computer Methods in Applied Mechanics and Engineering, 93(3), pp.291-318. Available at: [http://dx.doi.org/10.1016/0045-7825\(91\)90245-2](http://dx.doi.org/10.1016/0045-7825(91)90245-2).
- [32] Pt.wikipedia.org. (2016). Módulo de cisalhamento. [online] Available at: <https://pt.wikipedia.org/wiki/M2016>.
- [33] Engineeringtoolbox.com. (2016). Modulus of rigidity. [online] Available at: [http://www.engineeringtoolbox.com/modulus-rigidity-d\\_946.html](http://www.engineeringtoolbox.com/modulus-rigidity-d_946.html) [Accessed 27 Apr. 2016].
- [34] K. Yamazaki and A. (1988). Kobayashi. Optimum compliance design of stiffener layout of thin plate. 1st Report. A treatment as pseudo-continuous structure. [Abstract]. JSMET, 54(501), pp.1165-1171. Available at: <http://doi.org/10.1299/kikaia.54.1165>.
- [35] Pt.wikipedia.org. (2016). Módulo volumétrico. [online] Available at: [https://pt.wikipedia.org/wiki/M%C3%B3dulo\\_volum%C3%A9trico](https://pt.wikipedia.org/wiki/M%C3%B3dulo_volum%C3%A9trico) [Accessed 27 Apr. 2016].
- [36] Wikipedia. (2016). Young's modulus. [online] Available at: [https://en.wikipedia.org/wiki/Young%27s\\_modulus](https://en.wikipedia.org/wiki/Young%27s_modulus) [Accessed 27 Apr. 2016].
- [37] Wikipedia. (2016). Elastic modulus. [online] Available at: [https://en.wikipedia.org/wiki/Elastic\\_modulus](https://en.wikipedia.org/wiki/Elastic_modulus) [Accessed 27 Apr. 2016].
- [38] Wikipedia. (2015). Pattern search (optimization). [online] Available at: [https://en.wikipedia.org/wiki/Pattern\\_search\\_\(optimization\)](https://en.wikipedia.org/wiki/Pattern_search_(optimization)) [Accessed 29 Apr. 2016].
- [39] Engineering dictionary.org. (2016). Plate meaning — engineering dictionary. [online] Available at: <http://www.engineering-dictionary.org/plate> [Accessed 27 Apr. 2016].

- [40] Subsurfwiki.org. (2012). 1st lamé parameter - subsurfwiki. [online] Available at: [http://subsurfwiki.org/wiki/1st\\_Lam%C3%A9\\_parameter](http://subsurfwiki.org/wiki/1st_Lam%C3%A9_parameter) [Accessed 28 Apr. 2016].
- [41] Dictionary.com. (2010). The definition of stress. [online] Available at: <http://www.dictionary.com/browse/stress?s=t> [Accessed 27 Apr. 2016].
- [42] Pt.wikipedia.org. (2016). Tensor deformação. [online] Available at: [https://pt.wikipedia.org/wiki/Tensor\\_deforma%C3%A7%C3%A3o](https://pt.wikipedia.org/wiki/Tensor_deforma%C3%A7%C3%A3o) [Accessed 27 Apr. 2016].
- [43] Pt.encydia.com. (n.d.). Tensor deformação. [online] Available at: [http://pt.encydia.com/es/Tensor\\_deforma%C3%A7%C3%A3o](http://pt.encydia.com/es/Tensor_deforma%C3%A7%C3%A3o) [Accessed 27 Apr. 2016].
- [44] Wikipedia. (2016). Cauchy stress tensor. [online] Available at: [https://en.wikipedia.org/wiki/Cauchy\\_stress\\_tensor](https://en.wikipedia.org/wiki/Cauchy_stress_tensor) [Accessed 27 Apr. 2016].
- [45] Engineering dictionary.org. (2008). Work definition — engineering dictionary. [online] Available at: <http://www.engineering-dictionary.org/work> [Accessed 28 Apr. 2016].
- [46] Wikipedia. (2016). Trace (linear algebra). [online] Available at: [https://en.wikipedia.org/wiki/Trace\\_\(linear\\_algebra\)](https://en.wikipedia.org/wiki/Trace_(linear_algebra)) [Accessed 27 Apr. 2016].
- [47] Dicionário E-Civil. (2016). O que é treliça? [online] Available at: <http://www.ecivilnet.com/dicionario/o-que-e-trelica.html> [Accessed 27 Apr. 2016].
- [48] Jabir S. Arora. *Introduction to Optimum Design*. Elsevier, 2<sup>nd</sup> edition, 2004.
- [49] X. Ding and K. Yamakaki. Adaptive growth technique of stiffener layout pattern for plate and shell structures to achieve minimum compliance. volume 37(3), pages 259–276. *Engineering Optimization*, 2005.
- [50] Keng Tung Cheng and Niels Olhoff. An investigation concerning optimum design of solid elastic plates. volume 17, pages 305–323. *Int. J. Solids Struct.*, 1981.
- [51] Keng Tung Cheng and Niels Olhoff. Regularized formulation for optimum design of axisymmetric plates. volume 18, pages 153–169. *Int. J. Solids Struct.*, 1982.
- [52] Kouetsu Yamazaki and Atushi Kobayashi. Optimum compliance design of stiffener layout of thin plate. volume 54(501), pages 1165–1171. Japan Society of Mechanical Engineers, 1988.
- [53] Martin Philip Bendsoe and Noboru Kikuchi. Generating optimum topologies in structural design using homogenization method. volume 71(2), pages 197–224. *Comput. Methods Appl. Mech. Eng.*, 1988.
- [54] J. Chung and K. Lee. Optimal design of rib structures using the topology optimization technique. volume 211, pages 425–437. *Proc. Inst. Mech. Eng.*, 1997.
- [55] J. Luo and H.C. Gea. A systematic topology optimization approach for optimum stiffener design. In *Structural Optimization*, volume 16, pages 280–288. *Proc. Inst. Mech. Eng.*, 1998.



- [56] Lars A. Krog and Niels Olhoff. Optimum topology and reinforcement design of disk and plate structures with multiple stiffness and eigenfrequency objectives. volume 72, pages 535–563. *Comput. Struct.*, 1999.
- [57] Y.C. Lam and S. Santhikumar. Automated rib location and optimization for plate structures. In *Struct Multidisc Optim*, volume 25, pages 35–45. Springer-Verlag, 2003.
- [58] R. Ansola, J. Canales, J.A. Tarrago, and J. Rasmussen. Combined shape and reinforcement layout optimization of shell structures. In *Struct Multidisc Optim*, volume 27, pages 219–227. 2004.
- [59] Serkan Kiranyaz, Turker Ince, and Moncef Gabbouj. Optimization techniques: An overview. In *Multidimensional Particle Swarm Optimization for Machine Learning and Pattern Recognition*, volume 15, pages 13–44. Springer Berlin Heidelberg, 2014.
- [60] Nicos Theodossiou, Diamantis Karakatsanis, and Ioannis Kougiyas. The history of optimization. applications in water resources management. *IWA Regional Symposium on Water, Wastewater and Environment: Traditions and Culture. At Patras, Greece.*, 2014.
- [61] Neos-guide.org. *Optimization Taxonomy, NEOS.*, 2016. [online] Available at: <http://neos-guide.org/content/optimization-taxonomy> [Accessed 14 Feb 2016].
- [62] Peter W. Christensen and Anders Klarbring. *An Introduction to Structural Optimization*. Springer Science+ Business Media B.V., 1<sup>nd</sup> edition, 2009.
- [63] A. L. Custódio and J. F. A. Madeira. Glods: Global and local optimization using direct search. *Journal of Global Optimization*, 62:1–28, 2015.
- [64] Robert Hooke and T. A. Jeeves. Direct search, solution of numerical and statistical problems. *Journal of the Association for Computing Machinery*, 8(2):212–229, 1961. doi:10.1145/321062.321069.
- [65] R. Segurado, M. Costa, N. Duić, and M.G. Carvalho. Integrated analysis of energy and water supply in islands. case study of s. vicente, cape verde. *Energy.*, 92(3):639–648, 2015.
- [66] James Edward Gordon. *Structures or Why Things Dont Fall Down*. Penguin, 1<sup>nd</sup> edition, 1978.
- [67] Carlos A. Mota Soares. Teorias e análises de placas: Métodos analíticos e aproximados. pages 1–14. Technical report, 1982.
- [68] A. E. H. Love. In *On the small free vibrations and deformations of elastic shells*, volume 17, pages 491–546. Royal Society (London), 1988.
- [69] J. N. Reddy. *Theory and Analysis of Elastic Plates and Shells*. Taylor and Francis, 2<sup>nd</sup> edition, 2007.
- [70] J. Reddy and C. (2002). Wang. On shear deformation plate solutions: Relationship to the classical solutions. In: D. Durban, D. Givoli and J. Simmonds, ed., *Advances in the Mechanics of Plates and Shells*, 1st ed. New York: Kluwer Academic Publishers, pp.259-276.

- [71] Raphael T. Haftka and Zafer Gurdal. Elements of structural optimization. In *Solid Mechanics and Its Applications*, volume 11, page 3. Springer Netherlands, 1992.
- [72] Nielsen Castelo Damasceno. O que é um algoritmo de otimização? *Portal Educação*, 2013. <https://www.portaleducacao.com.br/educacao/artigos/33378/o-que-e-um-algoritmo-de-otimizacao>.
- [73] C. Vieira and V. Morais. *MATLAB 7 and 6*. FCA, Editora de Informática, Lisboa, 3<sup>rd</sup> edition, 2010.
- [74] ANSYS. *Structural Static Analysis (Online Documentação)*. ANSYS 11 Help System, 2007.
- [75] Mathworks. *Global Optimization Toolbox*. Disponível em: [www.mathworks.com/products/global-optimization/](http://www.mathworks.com/products/global-optimization/), 2016.