Optimização de Localização de Reforços em Painéis Planos e Cilíndricos

Hugo Filipe Macedo de Pina

Dissertação para obtenção do Grau de Mestre em

Engenharia Aeroespacial

Orientadores: Prof. José Arnaldo Pereira Leite Miranda Guedes

Prof. Helder Carriço Rodrigues

Júri

Presidente: Prof. Fernando José Parracho Lau

Orientador: Prof. José Arnaldo Pereira Leite Miranda Guedes

Vogais: Prof. João Orlando Marques Gameiro Folgado

Novembro de 2014

Resumo

O trabalho desenvolvido no âmbito desta tese considera a utilização de materiais compósitos. Estes materiais são utilizados em inúmeras indústrias, incluindo a aeroespacial, e compõem diferentes tipos de objectos e estruturas.

Em particular, esta tese trata do desenvolvimento de um programa computacional que permita a optimização de reforço de uma estrutura tipo OPENCELL, utilizando o método DMO (Discrete Material Optimization), com o objectivo de maximizar a rigidez da mesma. É definida a formulação do problema, nomeadamente a função objectivo, variáveis de projecto, funções de peso, constrangimentos e cargas a aplicar. O software de análise de elementos finitos utilizado é o ANSYS, sendo utilizados dois optimizadores diferentes a partir do matlab, o FAIPA, e o FMINCON (optimizador da toolbox do MATLAB). Recorre-se também ao MATLAB para criar a interface entre optimizador e software de elementos finitos.

O modelo computacional desenvolvido permite comprovar a eficácia desta inovadora forma de utilização do método DMO, onde são utilizadas configurações de reforço como candidatos. O programa permite ainda verificar qual a melhor forma de reforçar o painel OPENCELL utilizado nos casos de estudo.

Ao longo do trabalho foram testados vários parâmetros e condições de fronteira, de modo a analisar e verificar a eficácia desta forma de utilização do DMO em diferentes situações.

Palavras-chaves: *Materiais Compósitos, OpenCell, Ansys, elementos finitos, Matlab, Optimização estrutural, DMO.*

Abstract

The subject of this thesis considers the use of composite materials. These materials can be used in many industries, including aerospace, to construct different types of objects and structures.

This thesis deals with the development of a computational model which allows the optimization of the reinforcements of an OPENCELL structure, using the DMO's method (Discrete Material Optimization) to maximize the structure stiffness. The formulation is accomplished by defining the objective function, design variables, weight functions, constraints and applying loads. The software ANSYS is used to perform finite elements analyses and two optimizers are used, FAIPA and FMINCON (optimizer from MATLAB's toolbox). MATLAB is also used to create an interface between the optimizer and ANSYS.

The developed computational model allows to prove the effectiveness of this innovate way to use DMO's method, where pre defined configurations are used as candidates. Further, the computational model allows to check which is the best way to make de reinforcement of the OPENCELL panel.

During this thesis several parameters and boundary conditions were used, in order to analyze the effectiveness of this new way to use DMO's method in different situations.

Key-words: *Composite Materials, OpenCell, Ansys, Finite elements, Matlab, structural optimization, DMO's method.*

Agradecimentos

Quero expressar o meu profundo agradecimento a todos os que tornaram possível a realização desta tese, em especial ao meu orientador, Professor José Guedes, e co-orientador, Professor Helder Rodrigues, pelo apoio, orientação e disponibilidade ao longo do trabalho.

Agradeço também ao Instituto Superior Técnico, pela excelente formação que me proporcionou.

Por fim, e não menos importante, quero agradecer à minha família e amigos, por todo o apoio e carinho que me deram durante esta fase da minha vida. Um agradecimento especial aos meus pais e irmão pela sua dedicação, motivação e amor, essenciais no meu sucesso académico e na minha vida.

Índice

Lista de Figuras xi
Lista de Tabelas xiii
Lista de Acrónimos xvii
Lista de Símbolos xix
Capítulo 1 1
Introdução 1
1.1 Painéis Sanduíche 1
1.2 Painéis OPENCELL 2
1.3 Optimização de Estruturas 3
1.4 State of Art 4
1.5 Motivação e Objectivos 5
1.6 Organização da Tese 6
Capítulo 2 7
Problema de Optimização 7
2.1 Formulação do problema de optimização 7
2.1.1 Objectivos 8
2.1.2 Variáveis de Projecto 8
2.1.2.1 Método DMO 8
2.1.2.2 Parametrização de uma estrutura 9
2.1.2.3 Método de penalização explícito 13
2.1.3 Função Objectivo 15
2.1.3.1 Métodos analíticos de análise de sensibilidade 16
2.1.3.2 Gradiente da matriz de rigidez global 19
2.1.4 Constrangimentos 22
2.2 Formulação genérica utilizada 22
2.3 Algoritmos de optimização 22
Capítulo 3 26
Programa Computacional 26
3.1 Análise ANSYS 26
3.1.1 Pré-processamento 26
3.1.2 Pós-processamento 29
3.1.2.1 Matrizes de Rigidez 29

3.1.2.2 Energia elástica de deformação 30
3.1.2.3 Nós por elemento 30
3.1.2.4 Graus de liberdade 30
3.2 Algoritmo de Optimização 31
3.3 Interface ANSYS – MATLAB 32
3.4 Como correr o programa 34
Capítulo 4 35
Resultados e Discussão 35
4.1 Caso 1 - Utilização do método DMO 36
4.1.1 Modelo com 3 variáveis de projecto 36
4.1.1.1 Escolha do esquema de DMO 37
4.1.1.2 Escolha do factor de penalização *p* 39
4.1.2 Modelo com 6 variáveis de projecto 39
4.1.2.1 Escolha do esquema de DMO 41
4.1.2.2 Escolha do factor de penalização *p* 42
4.1.3 Modelo com 12 variáveis de projecto 43
4.1.3.1 Escolha do esquema de DMO e factor p 45
4.2 Caso 2 - Método DMO + Método penalização explícito 46
4.2.1 Modelo com 3 variáveis de projecto 46
4.2.2 Modelo com 6 variáveis de projecto 47
4.2.3 Modelo com 12 variáveis de projecto 48
4.3 Caso 3 - Utilização de duas funções penalizadores 49
4.4 Caso 4 - FMINCON 50
4.4.1 Esquema 5 do DMO + Método penalização explícito 50
4.4.2 Esquema 4 do DMO + Método Penalização Explícito 58
4.4.3 Esquema 1 do DMO + duas funções penalizadores 59
4.4.4 Utilizando diferenças finitas 61
4.4.5 FMINCON vs FAIPA 66
4.5 Caso 5 - 192 variáveis de projecto 68
4.6 Caso 6 - Casca cilíncrica 71
4.6.1 Casca cilíncrica com 1 furo 71
4.6.2 Casca cilíncrica com 4 furos 74
4.6.3 Estudo de convergência 75
Capítulo 5 77

Conclusões 77
Capítulo 6 79
Trabalho Futuro 79
Bibliografia 80
Anexo A 84
A.1 Exemplo de um ficheiro APDL que permite obter a energia, e os graus de liberdade de cada nó, a partir do software ANSYS 84
A.2 Exemplo de um ficheiro APDL que permite obter as matrizes de rigidez de cada elemento, a partir do software ANSYS 91
A.3 Exemplo da derivada da função de peso 91

Lista de Figuras

Figura 1: Algumas configurações e tipos de núcleo de compósitos (retirado de [2]) 1
Figura 2: Compósito OpenCell cilíndrico (retirado de [7]) 2
Figura 3: Exemplos de cortes em compósitos OpenCell (retirado de [7]) 3
Figura 4: Modelo de evolução de um processo de design (retirado de [8]) 4
Figura 5: Estrutura OpenCell com corte em I 9
Figura 6: Exemplos de dobragens possíveis 9
Figura 7: Penalização A com 0<q<1 Figura 8: Penalização A com q>1 14
Figura 9: Derivada da função de penalização A com q=0.2 e q=2 14
Figura 10: Configurações resultantes dos diferentes tipos de dobragem 20
Figura 11: Fluxograma de um processo de optimização 23
Figura 12: Elemento SHELL181 27
Figura 13: a) Algoritmo de optimização b) Método adjunto + DMO 31
Figura 14: Interface ANSYS - MATLAB 32
Figura 15: Candidato 1 Figura 16: Candidato 2 36
Figura 17: Candidato 3 36
Figura 18: Estrutura com a placa superior encastrada em y=0 e y=4 36
Figura 19: Estrutura com as placas superior e inferior encastradas em y=0 e y=4 37
Figura 20: Numeração dos 4 lados de reforço 43
Figura 21: Estrutura com as placas encastradas em x=0 e y=4 57
Figura 22: Numeração dos recortes de reforço 69
Figura 23: Estrutura cilíndrica 72
Figura 24: Estrutura cilíndrica com carga pontual 73
Figura 25: Configurações -- w4 e w1 -- w1 e w12 -- w1 e w7 respectivamente 75
Figura 26: Estudo convergência da malha 76

Lista de Tabelas

Tabela 1: Solução obtida directamente do ANSYS 38
Tabela 2: Solução da função objectivo usando o esquema 4 e 5 do DMO 38
Tabela 3: Solução das funções de peso usando o esquema 4 e 5 do DMO 38
Tabela 4: Função objectivo e funções de peso com p=2, p=4 e p=5 39
Tabela 5: Candidatos possíveis com 6 variáveis de projecto 40
Tabela 6: Solução obtida directamente do ANSYS 40
Tabela 7: Solução da função objectivo usando o esquema 4 e 5 do DMO 41
Tabela 8: Solução das funções de peso usando o esquema 4 e 5 do DMO 41
Tabela 9: Solução da função objectivo usando o esquema 4 e 5 do DMO 41
Tabela 10: Solução das funções de peso usando o esquema 4 e 5 do DMO 42
Tabela 11: Solução da função objectivo com p=2, p=4, p=6 e p=8 42
Tabela 12: Solução das funções de peso com p=2, p=4, p=6 e p=8 43
Tabela 13: Candidatos possíveis com 12 variáveis de projecto, para os lados 1 e 2 44
Tabela 14: Candidatos possíveis com 12 variáveis de projecto, para os lados 3 e 4 44
Tabela 15: Solução obtida directamente do ANSYS 45
Tabela 16: Solução da função objectivo usando o esquema 4 e 5 do DMO 45
Tabela 17: Solução da função objectivo usando o esquema 4 e 5 do DMO 47
Tabela 18: Solução das funções de peso usando o esquema 4 e 5 do DMO 47
Tabela 19: Solução da função objectivo usando o esquema 5 do DMO 48
Tabela 20: Solução das funções de peso usando o esquema 5 do DMO 48
Tabela 21: Solução da função objectivo usando o esquema 5 do DMO 48
Tabela 22: Solução da função objectivo usando o FAIPA e o FMINCON 50
Tabela 23: Solução das funções de peso usando o FAIPA e o FMINCON 51
Tabela 24: Solução obtida usando o FAIPA e o FMINCON 51
Tabela 25: Solução da função objectivo usando o FAIPA e o FMINCON 51
Tabela 26: Solução das funções de peso usando o FAIPA e o FMINCON 52
Tabela 27: Solução obtida usando o FAIPA e o FMINCON 52
Tabela 28: Solução da função objectivo usando p=2, p=4, p=6, p=8 e p=12 52
Tabela 29: Solução das funções de peso usando p=2, p=4, p=6, p=8 e p=12 53
Tabela 30: Solução das funções de peso usando p=2, p=4, p=6, p=8 e p=12 53
Tabela 31: Solução obtida usando p=2, p=4, p=6, p=8 e p=12 53
Tabela 32: Solução da função objectivo usando $\boldsymbol{k = 0.01}$, $\boldsymbol{k = 0.1}$, $\boldsymbol{k = 1}$, $\boldsymbol{k = 10}$ e $\boldsymbol{k = 100}$ 53
Tabela 33: Solução das funções de peso usando $\boldsymbol{k = 0.01}$, $\boldsymbol{k = 0.1}$, $\boldsymbol{k = 1}$, $\boldsymbol{k = 10}$ e $\boldsymbol{k = 100}$ 54

Tabela 34: Solução das funções de peso usando $\boldsymbol{k = 0.01}$, $\boldsymbol{k = 0.1}$, $\boldsymbol{k = 1}$, $\boldsymbol{k = 10}$ e $\boldsymbol{k = 100}$ 54
Tabela 35: Solução obtida usando $\boldsymbol{k = 0.01}$, $\boldsymbol{k = 0.1}$, $\boldsymbol{k = 1}$, $\boldsymbol{k = 10}$ e $\boldsymbol{k = 100}$ 54
Tabela 36: Solução obtida directamente do ANSYS com uma malha de 4 elementos por área 55
Tabela 37: Solução da função objectivo usando uma malha de 4 elementos 56
Tabela 38: Solução das funções de peso usando uma malha de 4 elementos 56
Tabela 39: Solução da função objectivo usando estrutura com 9 furos idênticos 56
Tabela 40: Solução das funções de peso usando estrutura com 9 furos idênticos 56
Tabela 41: Solução obtida usando estrutura com 9 furos idênticos 57
Tabela 42: Solução da função objectivo usando estrutura encastrada em x=0 e y=4 57
Tabela 43: Solução das funções de peso usando estrutura encastrada em x=0 e y=4 57
Tabela 44: Solução da função objectivo usando esquema 4 do DMO com penalizações 58
Tabela 45: Solução das funções de peso usando esquema 4 do DMO com penalizações ..59
Tabela 46: Solução obtida usando esquema 4 do DMO com penalizações 59
Tabela 47: Solução da função objectivo usando esquema 1 do DMO com penalizações 59
Tabela 48: Solução das funções de peso usando esquema 1 do DMO com penalizações ..60
Tabela 49: Solução das funções de peso usando esquema 1 do DMO com penalizações ..60
Tabela 50: Solução obtida usando esquema 1 do DMO com penalizações 60
Tabela 51: Número de iterações e tempos de análises usando os esquemas 1, 4 e 5 do DMO 61
Tabela 52: Solução da função objectivo usando esquema 1 do DMO com penalizações 62
Tabela 53: Solução das funções de peso usando esquema 1 do DMO com penalizações ..62
Tabela 54: Solução obtidas usando esquema 1 do DMO com penalizações 62
Tabela 55: Solução da função objectivo usando esquema 1 do DMO com penalizações 63
Tabela 56: Solução das funções de peso usando esquema 1 do DMO com penalizações ..64
Tabela 57: Solução das funções de peso usando esquema 1 do DMO com penalizações ..64
Tabela 58: Solução obtidas usando esquema 1 do DMO com penalizações 65
Tabela 59: Solução da função objectivo usando o FAIPA e o FMINCON 66
Tabela 60: Solução das funções de peso usando o FAIPA e o FMINCON 66
Tabela 61: Solução das funções de peso usando o FAIPA e o FMINCON 67
Tabela 62: Solução obtidas usando o FAIPA e o FMINCON 67
Tabela 63: Candidatos possíveis com 12 variáveis de projecto, lados 1 e 2 68
Tabela 64: Candidatos possíveis com 12 variáveis de projecto, lados 3 e 4 69
Tabela 65: Solução da função objectivo – placa com 192 variáveis 70
Tabela 66: Solução das funções de peso – placa com 192 variáveis 70
Tabela 67: Configuração final 70
Tabela 68: Solução da função objectivo – placa cilíndrica com pressão aplicada 72
Tabela 69: Solução das funções de peso – placa cilíndrica com pressão aplicada 72
Tabela 70: Solução das funções de peso – placa cilíndrica com pressão aplicada 72

Tabela 71: Configuração final – placa cilíndrica com pressão aplicada 72
Tabela 72: Solução da função objectivo – placa cilíndrica com carga pontual aplicada 73
Tabela 73: Solução das funções de peso – placa cilíndrica com carga pontual aplicada 73
Tabela 74: Solução das funções de peso – placa cilíndrica com carga pontual aplicada 73
Tabela 75: Configuração final - placa cilíndrica com carga pontual aplicada 74
Tabela 76: Solução da função objectivo - placa cilíndrica com 4 furos 74
Tabela 77: Solução das funções de peso - placa cilíndrica com 4 furos 74
Tabela 78: Solução das funções de peso - placa cilíndrica com 4 furos 75
Tabela 79: Configuração final - placa cilíndrica com 4 furos 75

Lista de Acrónimos

APDL – *ANSYS Parametric Design Language*

DMO – *Discrete Material Optimization*

DOF – *Degree of freedom (grau de liberdade)*

FAIPA – *Feasible Arc Interior Point Algorithm*

MPE – *Método de Penalização Explícito*

SIMP – *Solid Isotropic Material with Penalization*

STO – *Structural Topology Optimization*

Lista de Símbolos

A, B - *Funções de penalização*

[B] - *Matriz extensão-deslocamento*

C - *Compliance*

[D] - *Matriz constitutiva*

E - *Módulo de elasticidade*

f - *Vector de cargas*

H - *Matriz hessiana*

[K] - *Matrizes de rigidez global*

$[K]^e$ - *Matrizes de rigidez elementares*

k,q,c - *Constantes*

N^c - *Número de candidatos*

p - *Fator de penalização das variáveis de projeto*

U - *Energia elástica de deformação total da estrutura*

u - *Vector deslocamentos*

v - *Deslocamento adjunto*

x_0 - *Valor inicial das variaveis de projecto*

x_i - *Variável de project*

w_i - *Funções de peso*

Capítulo 1

Introdução

Neste capítulo é feita uma breve introdução ao assunto em estudo no presente trabalho. São, também, referidos os principais objectivos e motivações que levaram à realização da tese. Por fim, é apresentada a organização dos conteúdos que compõem a tese.

1.1 Painéis Sanduíche

A necessidade actual de obtenção de materiais leves e com elevada resistência tem aumentado a procura e desenvolvimento de vários tipos de compósitos.

Obtidos pela combinação de duas lâminas exteriores, compostas por materiais mais resistêntes, e um material de enchimento interior, com uma menor resistência e rigidez mas com altura elevada [1], [2] e [3], os painéis sanduíche são um dos tipos de compósitos mais utilizados nas indústrias aeroespacial, automóvel, naval, de construção, entre outras.

Os painéis sanduíche apresentam grandes vantagens [4] ao nivel da resistência e rigidez específica que, devido à altura do núcleo do painel, são bastante elevadas. Estes compósitos são bons isoladores térmicos, resistem a ambientes hostis e o facto de o núcleo ser leve reduz significativamente o peso da estrutura [5].

As lâminas superior e inferior são geralmente feitas em metal ou fibras reforçadas de modo a que possuam a resistência pretendida [6]. Este aspecto, aliado ao menor peso e custo possível, é fulcral na escolha da lâmina.

Existem vários tipos de núcleo que se pode utilizar (espumas, madeira, favos de mel, cortiça, entre outros) e várias configurações possíveis para o mesmo. A sua escolha deve ter em conta os propósitos da sua aplicação [6].

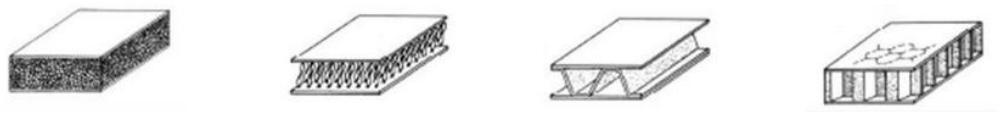


Figura 1: Algumas configurações e tipos de núcleo de compósitos (retirado de [2])

1.2 Painéis OPENCELL

Recentemente, foi criado pela PLY Engenharia uma inovadora classe de estruturas sanduíche que difere das configurações convencionais no facto de não utilizar material adicional no núcleo. O nome desta nova classe é OPENCELL.

Esta estrutura possui duas lâminas externas que asseguram uma boa resistência à flexão, tal como num painel sanduíche convencional. Neste caso, uma das lâminas é recortada, sendo esse recorte dobrado e soldado à outra lâmina. Assim, o material que irá compor o núcleo e conferir a espessura ao painel é o mesmo que faz parte da lâmina.

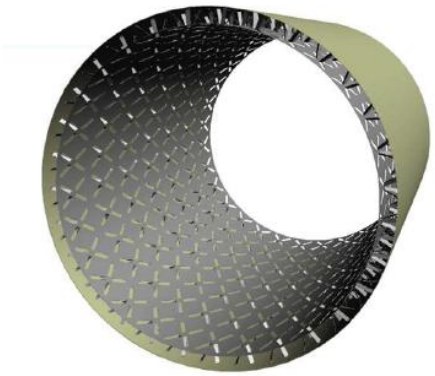


Figura 2: Compósito OpenCell cilíndrico (retirado de [7])

Com a utilização deste tipo de estruturas verificam-se as seguintes vantagens:

- A rigidez e resistência pretendidas são obtidas sem se recorrer a outro tipo de material.
- Diminui os custos de produção.
- Diminui o tempo e custo do processo de reciclagem.
- Menor peso da estrutura/painel.

É ainda possível aplicar vários tipo de corte à chapa sendo alguns deles expostos na figura seguinte.

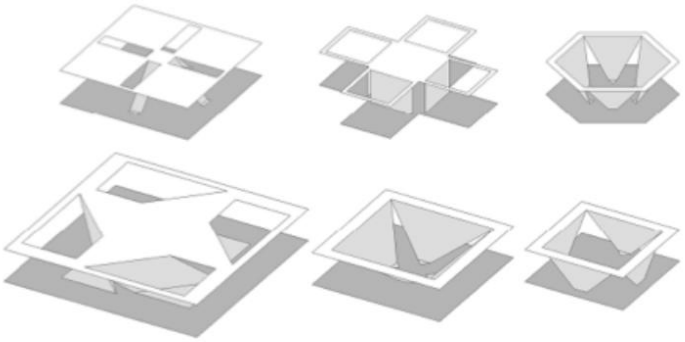


Figura 3: Exemplos de cortes em compósitos OpenCell (retirado de [7])

1.3 Optimização de Estruturas

A engenharia consiste num número de processos bem definidos, onde se inclui a análise, design, fabrico, pesquisa, desenvolvimento de sistemas e vendas. O desenvolvimento de sistemas é o maior campo profissional da engenharia. Os sistemas de fabrico e design têm sido desenvolvidos ao longo dos séculos. No entanto, a evolução destes sistemas é lenta e está dependente de custos materiais e humanos elevados [8].

O design de sistemas complexos requer um grande número de cálculos e capacidade de processamento de dados. Nos últimos anos houve uma grande evolução ao nível da tecnologia computacional o que tornou possível, utilizando os computadores actuais, realizar cálculos complexos e enormes processamentos de dados num curto espaço de tempo. A optimização de processos beneficiou muito com esta revolução. Actualmente é possivel desenvolver melhores sistemas através da análise e optimização de várias opções, num curto espaço de tempo. Isto conduz a sistemas com maior capacidade, melhor design e fáceis de operar a custos significativamente menores.

O design de sistemas pode ser formulado como um problema de optimização onde se pretende obter a melhor solução possível, satisfazendo uma série de constrangimentos.

Este processo tem início com a análise de várias opções sendo os vários subsistemas e seus componentes identificados e testados. Recorrendo a uma série de iterações, o processo de design decorre de forma a determinar se é aceitavel ou não. Sendo aceitavel, segue-se o processo de optimização que determina a melhor solução.

Tendo início com a identificação de uma necessidade, o processo deve ser bem organizado e seguir um modelo de evolução [8]. Os passos a seguir estão identificados na figura seguinte.

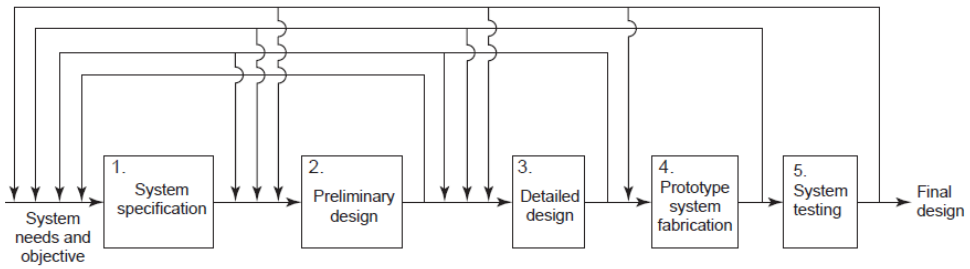


Figura 4: Modelo de evolução de um processo de design (retirado de [8])

O primeiro passo é definir as especificações necessárias do sistema.

O desenvolvimento do design preliminar do sistema é feito na segunda etapa do processo. Vários subsistemas são identificados e o seu design preliminar estimado. Esta fase deve ser curta.

A terceira fase consiste em desenvolver o design detalhado para os subsistemas utilizando um processo iterativo. Os parâmetros de design devem ser identificados e os requerimentos do sistema satisfeitos. Os subsistemas são desenhados para minimizar custos ou maximizar algum parâmetro pretendido.

O quarto e quinto passo podem, ou não, ser necessários. Estes consistem na fabricação e teste do sistema em questão. Estas fases são essenciais quando estão envolvidas vidas humanas.

Em suma, a optimização de estruturas consiste em minimizar, ou maximizar, uma determinada função objectivo atendendo aos constrangimentos definidos. Devem ainda ser escolhidos um conjunto de parâmetros (variáveis de projecto), que estão relacionadas com a função objectivo, e se pretende alterar de modo a obter a melhor solução final.

1.4 State of Art

Os primeiros passos da optimização estrutural consistiram na utilização de métodos analíticos e foram introduzidos já no final do século XIX. No entanto, esses métodos não eram muito eficientes, já que apenas permitiam resolver uma classe muito restrita de problemas [9].

O processo recente de optimização estrutural coincide com o desenvolvimento de computadores com grande capacidade de cálculo, que começaram a surgir por volta dos anos 60.

Em 1981, Cheng e Olhoff [10] desenvolveram um eficiente algoritmo numérico capaz de obter a configuração óptima para uma placa circular, utilizando a espessura da placa como variável de projecto.

Depois disto, Bendsoe e Kikuchi [11] trataram, pela primeira vez, o problema de optimização de topologia de um disco plano como um problema de distribuição de material com uma densidade de material variável. O método consiste na optimização da geometria através da distribuição de material, variando a microestrutura entre sólido e vazio. Um ano mais tarde, Bendsoe introduziu o método SIMP (Solid Isotropic Material with Penalization) [12], que foi implementado por Zhou e Rozvany, [13], em 1991.

Enquanto os problemas de optimização topológica eram tratados por Bendsoe e Kikuchi [11], apareciam novas extensões destes problemas com múltiplas cargas aplicadas [14], estruturas bi-material [15] e problemas de flexão de placas e cascas [16], [17] e [18]. Uma visão geral dos problemas de optimização topológica é apresentada por Bendsoe e Mota Soares [19], Rozvany e Bendsoe [20], Pedersen [21] e Olhoff e Rozvany [22].

Em 1997, Chung e Lee [23] estudaram a forma e localização óptima de reforços com o objectivo de aumentar a rigidez da estrutura, utilizando a técnica de optimização topológica. O processo de optimização consiste em encontrar a densidade de cada elemento que minimiza a *compliance* de toda a estrutura.

Dois anos depois, Krog e Olhoff [24] resolveram o problema de optimização da rigidez para diferentes tipos de cargas estáticas e modos de vibração livre, com base no método de homogeneização.

Stegmann e Lund [25], [26] e [27] em 2005 apresentaram um novo método de parametrização, o DMO (Discrete Material Optimization). Este método surge como uma extensão do trabalho desenvolvido por Sigmund e Torquato [28], que utilizaram o método de optimização de três fases para projectar a distribuição de material em compósitos com coeficientes de expansão térmica extremos.

Recentemente, vários são os trabalhos que utilizam o método DMO [29-31] devido à sua versatilidade e fácil implementação. Entre eles está o trabalho de Wang e Wang onde foi introduzido um novo método, o *Level-set method*, baseado nas mesmas ideias que Sigmund utilizou.

O método DMO é também utilizado neste trabalho como forma de determinar a melhor distribuição de material, que maximiza a rigidez da placa.

1.5 Motivação e Objectivos

Paineis Sanduíche são bastante utilizados na indústria mecânica e aeroespacial. Por isso, actualmente, qualquer tipo de progresso é extremamente útil para o crescimento destes ramos da engenharia. Sendo o OPENCELL uma recente variante de estruturas sanduíche, este desenvolvimento tem uma importância ainda maior.

O objectivo desta tese é desenvolver um modelo computacional que permita determinar a localização óptima de reforços, em paineis planos e cilíndricos, de modo a levar em conta um conjunto de requisitos:

- Carregamentos estáticos
- Deformada da estrutura
- Volume limitado de material disponível para o reforço

De forma a atingir o objectivo do trabalho pretende-se criar um modelo computacional para a análise estrutural, utilizando o programa ANSYS (software de elementos finitos) e um modelo computacional para a optimização, utilizando o MATLAB (software utilizado para o cálculo numérico).

A optimização deste painel é realizada utilizando o método de DMO, de modo a maximizar a rigidez da estrutura. A forma do corte e dobragem de uma das chapas do painel OPENCELL é escolhida a partir de um determinado leque de opções.

Numa primeira fase o estudo é feito para uma placa plana sendo, posteriormente realizado para um painel cilíndrico.

1.6 Organização da Tese

A tese em questão encontra-se dividida em 6 capítulos. No primeiro capítulo é apresentada uma breve introdução ao estado de arte e aos temas que irão ser abordados ao longo do trabalho.

O capítulo 2 apresenta a formulação teórica necessária para o problema de optimização, bem como uma breve introdução aos algoritmos de optimização.

No terceiro capítulo é apresentado o modelo computacional desenvolvido. Primeiro é explicado o funcionamento do modelo computacional para a análise estrutural, realizado através do programa ANSYS e, de seguida, apresenta-se o modelo computacional desenvolvido para a optimização, através do MATLAB.

O capítulo 4 expõe a discussão e análise de resultados obtidos ao longo do desenvolvimento do trabalho.

Por fim, no quinto e sexto capítulos são apresentadas, respectivamente, as conclusões e algum trabalho que pode ser desenvolvido futuramente.

Capítulo 2

Problema de Optimização

Neste capítulo é abordado o processo de formulação de um problema de optimização, sendo também, explicados os objectivos deste tipo de problemas. É feita uma descrição de alguns conceitos teóricos relevantes e apresentadas as variáveis de projecto, função objectivo e constrangimentos. Por fim, é feita uma introdução sobre algoritmos de optimização.

2.1 Formulação do problema de optimização

Uma correcta formulação do problema de optimização é essencial para a obtenção de uma adequada solução óptima, pois essa solução será apenas tão boa quanto a sua formulação [8]. Visto que o principal objectivo da formulação de um problema de optimização é traduzir o problema de uma linguagem comum para uma linguagem matemática, esta deve seguir algumas etapas essenciais. Assim, na maioria dos problemas de optimização devem ser utilizados os seguintes procedimentos:

- Objectivos do problema de optimização
- Identificação das variáveis de projecto
- Identificação da função objectivo
- Identificação dos constrangimentos

O processo de formulação começa com uma apresentação descritiva do problema. Nesta fase, são definidos os principais objectivos e requerimentos que devem ser cumpridos ao longo do projecto.

A segunda etapa consiste na identificação de um conjunto de variáveis que descrevam o sistema. Estas são denominadas variáveis de projecto. As variáveis de projecto podem assumir qualquer valor, desde que não violem os constrangimentos impostos, e devem ser independentes umas das outras. O número de variáveis utilizadas especifica o número de graus de liberdade do problema.

A terceira fase da formulação consiste na obtenção de um critério que permita comparar as diferentes soluções do projecto. Este critério deve ser uma função matemática a partir da qual é obtido um valor numérico que permita fazer essa comparação. Esta função é denominada função objectivo e deve possibilitar relacionar as variáveis de projecto com o que se pretende maximizar ou minimizar.

Por fim, a última etapa é aplicada apenas em certas ocasiões onde é necessário impôr algumas limitações ou restrições ao projecto. Estas restrições são chamadas constrangimentos.

Nos próximos subcapítulos são apresentados as várias etapas de formulação do problema desenvolvido nesta tese.

2.1.1 Objectivos

O objectivo principal do trabalho consiste em desenvolver um modelo computacional que permita determinar a melhor configuração de reforço, a aplicar em painéis planos e cilíndricos.

Desta forma, pretende-se realizar a optimização destes painéis utilizando o método de DMO, de uma forma um pouco diferente do habitual, de modo a maximizar a rigidez da estrutura. A explicação do modo de utilização do método de DMO é feita na secção 2.1.2.1 do trabalho. Para além das duas chapas que compõem o painel, não pode ser utilizado material adicional.

2.1.2 Variáveis de Projecto

O método dos elementos finitos é hoje aplicado num grande número de problemas de engenharia. Esta é, também, uma das metodologias mais utilizadas na análise numérica do comportamento mecânico de estruturas. O método consiste em dividir o domínio em subdomínios, chamados elementos finitos, e usando conceitos variacionais constroi-se uma solução aproximada em todo o domínio [32]. Atendendo ao tipo de estruturas que se pretende analisar neste trabalho, são utilizados elementos tipo casca (Shell) para se modelar a estrutura.

2.1.2.1 Método DMO

A ideia de parametrização utilizada no método DMO (*Discrete Material Optimization*) consiste numa extensão da ideia utilizada no STO *(Structural Topology Optimization)* mas, em vez de se escolher entre material ou vazio, o objectivo é agora decidir entre um número de materiais distintos. A ideia é então, para todos os elementos da estrutura encontrar um material distinto, de um conjunto pré-definido de materiais, de forma a minimizar a função objectivo.

Esta é a ideia base utilizada neste trabalho, no entanto, será utilizada de uma forma um pouco diferente.

A ideia consiste em utilizar diferentes configurações para o reforço da placa, em vez de diferentes materiais.

Tomemos como exemplo a figura que se segue, considerando o corte realizado. Pretende-se escolher qual a melhor opção de dobragem da placa de cima.

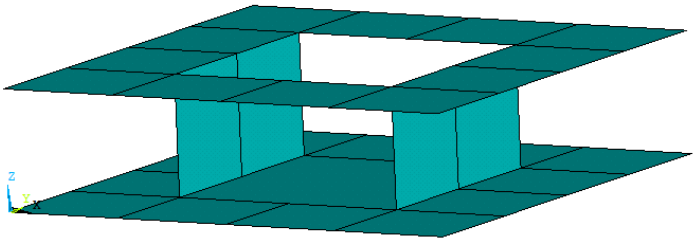


Figura 5: Estrutura OpenCell com corte em I

Consideremos então, como é possível observar na figura, três opções de dobragem diferente.

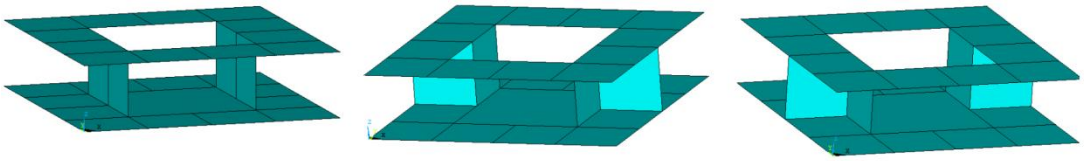


Figura 6: Exemplos de dobragens possíveis

Neste caso, o "material 1" será o tipo de dobragem da figura da esquerda, o "material 2" o tipo de dobragem da figura do meio e o "material 3" o tipo de dobragem da figura da direita. Assim, pretende-se definir qual o melhor "material" para os apoios verticais da placa, isto é, qual das três configurações possíveis minimiza a função objectivo.

2.1.2.2 Parametrização de uma estrutura

A parametrização do método DMO é feita através de elementos finitos. A matriz constitutiva, [D], é determinada a partir da soma de um conjunto de matrizes constitutivas pré-definidas, $[D]_i$, que correspondem cada uma a um candidato. Assim, a matriz [D] pode ser obtida através da seguinte expressão, para um número de candidatos N^c:

$$[D] = \sum_{i=1}^{N^c} w_i [D]_i = w_1[D]_1 + w_2[D]_2 + w_3[D]_3 + \ldots + w_{N^c}[D]_{N^c}\ , \quad 0 \leq w_i \leq 1 \qquad (2.1)$$

Observa-se, então, que o número de candidatos é igual ao número de variáveis de design.

As variáveis w_i são designadas funções de peso e irão estar directamente ligadas com as variáveis de projecto. As funções de peso assumem valores entre 0 e 1 de modo a garantir que não se obtêm propriedades do material sem qualquer significado físico, isto é, que cada matriz não contribua com mais que as suas propriedades físicas do material e também não sejam propriedades negativas.

O DMO permite que o optimizador utilizado conduza as funções de peso para os seus valores extremos (0 ou 1). Assim, o objectivo do método DMO consiste em obter apenas um $w_i = 1$, sendo todas as outras funções de peso iguais a 0. Qualquer elemento que possua valores intermédios para as funções de peso não possui significado físico, tornando, desta forma, o problema irrealista. O mesmo acontece para pesos superiores a 1 e inferiores a 0.

Podemos então verificar que, quando se pretende resolver problemas de optimização deste tipo, o somatório das funções de peso deve ser igual a 1.

$$\sum_{i=1}^{N^c} w_i = 1 \tag{2.2}$$

Os valores iniciais das variáveis de design podem ser quaisquer valores entre 0 e 1, no entanto, os valores devem ser escolhidos de forma a que sejam uniformes, de modo a que o ponto de início da optimização não favoreça nenhum dos candidatos.

Vários esquemas de interpolação têm sido desenvolvidos ao longo dos anos. De seguida são apresentados cinco esquemas, [25], sendo que os esquemas 1, 4 e 5 são utilizados nesta tese.

- Esquema 1

Este esquema é considerado o mais simples e aquele que possui a escolha de funções de peso mais óbvia. A ideia consiste em aplicar um factor de penalização, p, de modo a tentar penalizar os seus valores intermédios e obter apenas uma função de peso igual a 1 e as outras iguais a 0, [33].

$$[D] = (x_1)^p[D]_1 + (x_2)^p[D]_2 + (x_3)^p[D]_3 + \ldots + (x_{N^c})^p[D]_{N^c} \tag{2.3}$$

$$[D] = \sum_{i=1}^{N^c} w_i[D]_i = \sum_{i=1}^{N^c} (x_i)^p[D]_i\,, \quad p \geq 1\,, \quad 0 \leq x_i \leq 1 \tag{2.4}$$

Este é um esquema não muito eficiente, já que tem algumas dificuldades em assegurar que as funções de peso assumam valores próximos dos seus limites.

❖ Esquema 2

O problema relacionado com o esquema 1 foi tratado por Sigmund [28,34] que criou um novo esquema para dois materiais. Mais tarde, Stegmann e Lund fizeram uma adaptação desse esquema e que resultou no seguinte:

$$[D] = (x_0)^p((1-(x_1)^p)[D]_1 + (x_1)^p[D]_2) \;\leftrightarrow$$

$$\leftrightarrow [D] = (x_0)^p(w_1[D]_1 + w_2[D]_2)\;, \quad p \geq 1\;, \;\; 0 \leq x_0 \leq 1, \;\; 0 \leq x_1 \leq 1 \tag{2.5}$$

Este esquema é um pouco diferente do anterior na medida em que possui uma variável de projecto, x_0 , que posibilita a optimização de topologia, enquanto x_1 faz a optimização na escolha do candidato. A maior diferença entre estas duas formulações é o termo $(1-(x_1)^p)$ que permite um maior controlo dos valores das funções de peso. Assim, o aumento do valor de uma das funções leva à diminuição do valor da outra, permitindo obter o valor 1 para uma das funções e 0 para a outra.

Este é um esquema pouco utilizado pois apenas permite utilizar dois candidatos. De forma a solucionar este problema, Stegmann [25] criou o esquema que se segue.

❖ Esquema 3

O esquema 3 é uma simples extensão do esquema 2 que permite utilizar um qualquer número de candidatos.

$$[D] = (x_0)^p \sum_{i=1}^{N^c-1} w_i[D]_i + w_{N^c}[D]_{N^c} \tag{2.6}$$

$$[D] = (x_0)^p \sum_{i=1}^{N^c-1} \left((1-(x_i)^p) \prod_{j=1}^{i-1} (x_j)^p \right) [D]_i + \prod_{j=1}^{N^c-1} (x_j)^p [D]_{N^c}\;, \;\; p \geq 1\;, \;\; 0 \leq x_i \leq 1 \tag{2.7}$$

De modo a tornar mais perceptível a interpretação da expressão, apresenta-se de seguida o caso particular de utilização de 3 candidatos, $N^c = 3$.

$$[D] = (x_0)^p\{(1-(x_1)^p)[D]_1 + (x_1)^p(1-(x_2)^p)[D]_2 + (x_1)^p(x_2)^p[D]_3\}\,, \quad p \geq 1\;, \;\; 0 \leq x_i \leq 1 \tag{2.8}$$

Este é um esquema bastante eficiente quando se utilizam 2 ou 3 candidatos, no entanto, quando se aumenta o número de candidatos, o esquema deixa de ter capacidade de encontrar a solução óptima global, ficando apenas numa das soluções óptimas locais.

❖ Esquema 4

Como alternativa ao esquema anterior foi desenvolvido o esquema 4, que não é mais do que uma extensão do esquema 1. A diferença é que, agora, as variáveis de projecto afectam todos os pesos.

$$[D] = \sum_{i=1}^{N^c} w_i [D]_i = \sum_{i=1}^{N^c} \left(((x_i)^p) \prod_{j=1; j\neq i}^{N^c} (1-(x_j)^p) \right) [D]_i \,, \quad p \geq 1 \,, \; 0 \leq x_i \leq 1 \tag{2.9}$$

De seguida é apresentada a mesma expressão para o caso de três candidatos.

$$\begin{aligned} [D] = {} & (x_1)^p (1-(x_2)^p)(1-(x_3)^p)[D]_1 + (x_2)^p (1-(x_1)^p)(1-(x_3)^p)[D]_2 \\ & + (x_3)^p (1-(x_1)^p)(1-(x_2)^p)[D]_3 \,, \qquad p \geq 1 \,, \; 0 \leq x_i \leq 1 \end{aligned} \tag{2.10}$$

O termo $(1-(x_j)^p)$ é incluido neste caso para que um aumento de uma das funções de peso leve à diminuição de todas as outras. Este termo ajuda a deslocar as funções para valores de 1 ou 0.

A desvantagem em utilizar este esquema consiste no facto de o somatório das funções de peso nem sempre ser igual a um. Este facto pode levar a que seja necessário um maior número de iterações para atingir a solução óptima, à não convergência do método ou até mesmo à obtensão de resultados incorrecto.

❖ Esquema 5

Por último, o esquema 5 é uma extensão do esquema 4 onde é introduzido um denominador comum em todas as funções de peso, que assegura que o somatório das funções de peso é sempre igual a um.

$$[D] = \sum_{i=1}^{N^c} \frac{\hat{\mathrm{w}}_{\mathrm{i}}}{\sum_{k=1}^{N^c} \hat{\mathrm{w}}_k} [D]_i \tag{2.11}$$

$$onde \qquad \hat{\mathrm{w}}_i = (x_i)^p \prod_{j=1\,; j\neq i}^{N^c} (1-(x_j)^p) \,, \quad p \geq 1 \,, \; 0 \leq x_i \leq 1 \tag{2.12}$$

Esta formulação permite uma convergência mais rápida da solução óptima, no entanto apresenta uma maior dificuldade em assegurar que as funções de peso atinjam os seus valores limites de 1 ou 0. Isto ocorre devido à introdução do denominador que adimensionaliza o valor das funções de peso. Essencialmente, este factor conduz a um aumento do número de combinações favoráveis de variáveis de projecto, o que torna a interpolação menos nítida. Por outras palavras, este factor altera o efeito de penalização das variáveis de projecto e leva a

uma diminuição do “favoritismo” dos valores extremos (1 e 0) em relação aos valores intermédios.

2.1.2.3 Método de penalização explícito

Em certos casos, o optimizador não consegue escolher entre dois ou mais candidatos. Isto pode ocorrer, por exemplo, devido à simetria que a estrutura apresenta. Noutros casos, devido a algumas limitações do método DMO, apresentadas no sub-capítulo anterior, o programa prefere optar por soluções “cinzentas”, ou intermédias, o que tornaria o problema irrealista.

De modo a minimizar a possibilidade de o programa apresentar este tipo de soluções, irrealista face ao objectivo do trabalho, é possível implementar um método de penalização explícito. Este método permite reduzir a possibilidade do programa indicar valores intermédios para as funções de peso, em detrimento do pretendido (0 ou 1).

O método de penalização explícito consiste em somar um novo termo à função objectivo, que penalize os valores intermédios, isto é,

$$\hat{C} = C + A\,, \qquad A \geq 0 \tag{2.13}$$

Assim, sempre que as funções de peso assumem valores diferente de 0 e 1, ou o seu somatório é diferente de 1, o valor da função objectivo sofre um aumento significativo.

A função que se segue, pode ser utilizada como função como penalizadora.

$$A = k * \sum_{i=1}^{N^c} {w_i}^q (1 - w_i)^q\,, \qquad q \geq 0\,,\ k \geq 0\,,\ 0 \leq w_i \leq 1 \tag{2.14}$$

Nas figuras que se seguem é possivel observar a função penalizadora utilizando $k = 1$ e variando os valores de q.

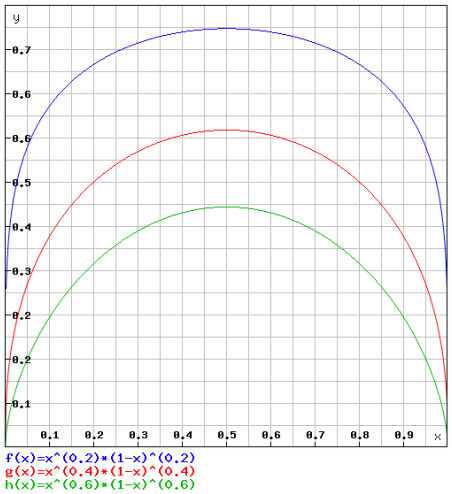


Figura 7: Penalização A com 0<q<1

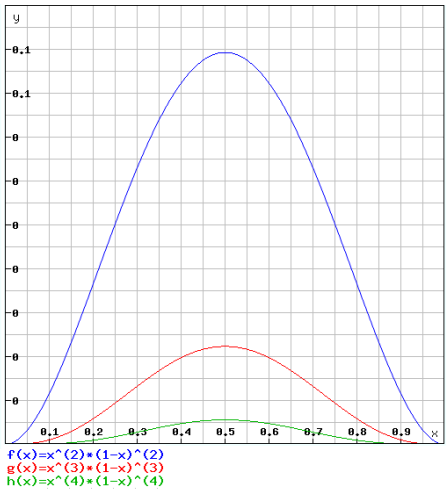


Figura 8: Penalização A com q>1

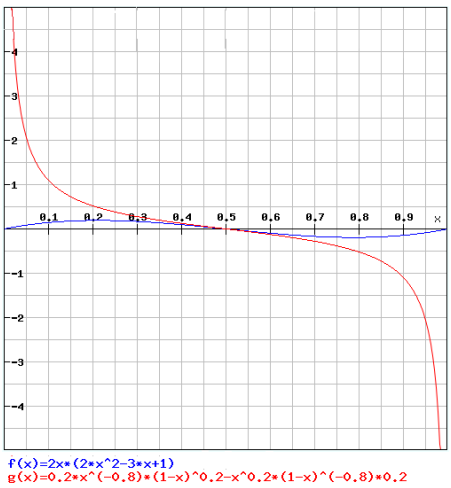


Figura 9: Derivada da função de penalização A com q=0.2 e q=2

Nos gráficos das figuras 7 e 8 é possível verificar que quando maior for o expoente q, menor será o valor da função. No entanto, como estamos a multiplicar a função por um factor k, a variação de q não é muito importante relativamente a este aspecto.

O principal factor a ter em conta, na escolha de q, é a forma como varia a função entre 0 e 1 e, ao mesmo tempo, o seu comportamento próximo desses mesmos extremos. Analisando as figuras 7 e 8 observa-se que, para valores de q superiores à unidade, a função penaliza pouco soluções próximas dos extremos, o que pode levar a que sejam escolhidas funções de peso ainda algo afastadas de 0 e 1. Veja-se o caso de $q = 4$, onde um $w = 0.15$ não sofre praticamente nenhuma penalização. No caso de se utilizar q inferior à unidade, verifica-se que a função já permite penalizar estas soluções próximas dos extremos. No entanto, se q tomar valores muito próximos de 0.5 a função praticamente não varia, o que pode levar a alguns problema no caso de se utilizar variáveis de projecto iniciais iguais a 0.5 .

Outro factor importante na escolha de q é o comportamento da função próximo dos seus extremos, pois verifica-se que, quanto mais afastado do valor 1 for o valor de q, mais a derivada da função se aproxima de infinito (ver figura 9). Isto pode causar alguns problemas de cálculo no optimizador.

Assim, todos estes aspectos devem ser considerados para uma escolha adequada do valor q .

2.1.3 Função Objectivo

A *compliance, C*, é o trabalho realizado pelas cargas aplicadas a uma estrutura, quando a mesma se encontra em equilíbrio. Através da minimização deste trabalho é possível não só determinar a rigidez máxima de uma estrutura, que se encontre em domínio elástico, mas também, de uma forma indirecta, minimizar os deslocamentos provocados pelas cargas. Tendo em conta os objectivos principais da tese, optou-se por escolher a *compliance, C*, como função objectivo.

A partir do método de elementos finitos [26, 27, 35] verifica-se que a matriz de rigidez global, $[K]$, é obtida através do somatório das matrizes de rigidez elementares, $[K]^e$, sendo estas obtidas pela expressão abaixo exposta:

$$[K]^e = \int_V [B]^T [D][B]\, dV \tag{2.15}$$

$$[K] = \sum_{e=1}^{N^e} [K]^e \tag{2.16}$$

onde $[B]$ é a matriz extensão-deslocamento e $[D]$ a matriz constitutiva.

A partir do vector de cargas, f, e da matriz de rigidez é possível obter o campo de deslocamentos, u, através da seguinte expressão.

$$[K]u = f \tag{2.17}$$

Sendo a *compliance, C*, o trabalho realizado pelas cargas aplicadas na estrutura, podemos relaciona-la com o campo de deslocamentos e o vector de cargas.

$$C = f^T u = u^T [K] u \tag{2.18}$$

Assim, conclui-se que a *compliance, C*, é igual a duas vezes a energia elástica de deformação total da estrutura, U.

$$C = 2U \tag{2.19}$$

Passando agora à determinação da função objectivo e do seu gradiente em relação às variáveis de projecto, verifica-se que:

$$C(u,x) = f^T u \tag{2.20}$$

$$\frac{dC}{dx}(u,x) = \frac{\partial C}{\partial x} + \frac{\partial C}{\partial u}\frac{\partial u}{\partial x} = f^T \frac{\partial u}{\partial x} \tag{2.21}$$
$$0 \leq x_i \leq 1$$

Observando a expressão (2.20) verifica-se que a função objectivo não depende explicitamente das variáveis de projecto, motivo pelo qual a derivada parcial $\frac{\partial C}{\partial x}$ é nula.

Apesar de não existir uma relação explícita que relacione a função objectivo ou o seu gradiente com as variáveis de projecto, facilmente que consegue determinar a função objectivo, C, através do campo de deslocamentos (2.18) ou mesmo da energia elástica de deformação (2.19). O mesmo já não se pode dizer do gradiente da função objectivo, pois não se conhece a relação $\frac{\partial u}{\partial x}$.

Com o objectivo de solucionar este problema, optou-se por recorrer a métodos analíticos [36] onde é possível determinar, de forma exacta, o gradiente da função objectivo.

2.1.3.1 Métodos analíticos de análise de sensibilidade

O método directo e o método adjunto são dois métodos analíticos que permitem obter o valor do gradiente da função objectivo. A dedução de ambos é feita a partir da mesma expressão (2.23). [36]. Começando por utilizar a expressão (2.17), verifica-se que:

$$[K]u = f \quad \leftrightarrow \quad [K]u - f = 0 \tag{2.22}$$

Derivando todos os termos em relação às variáveis de projecto obtem-se,

$$\frac{\partial([K]u)}{\partial x} - \frac{\partial f}{\partial x} = 0 \quad \leftrightarrow \quad \frac{\partial [K]}{\partial x}u + [K]\frac{\partial u}{\partial x} - \frac{\partial f}{\partial x} = 0 \tag{2.23}$$

Passemos agora à dedução do método directo.

- Método directo

De forma a determinar o gradiente pretendido, começa-se por reorganizar os termos da expressão anterior.

$$[K]\frac{\partial u}{\partial x} = \frac{\partial f}{\partial x} - \frac{\partial [K]}{\partial x}u \tag{2.24}$$

Neste caso, o termo $\frac{\partial f}{\partial x} - \frac{\partial [K]}{\partial x} u$ é denominado pseudo-carga [36], porque tanto a expressão (2.17) como a (2.24) apresentam a mesma estrutura.

Sendo o vector f independente das variáveis de projecto, o termo $\frac{\partial f}{\partial x}$ é nulo.

Assim, para obter as várias componentes do gradiente da função objectivo basta multiplicar o vector f^T pela respectiva componente do termo $\frac{\partial u}{\partial x}$. Desta forma obtem-se,

$$\frac{dC}{dx_i} = f^T \frac{\partial u}{\partial x_i}, \quad onde \quad \frac{\partial u}{\partial x_i} = [K]^{-1} \left(-\frac{\partial [K]}{\partial x_i} u \right), \quad i = 1,2,3, \dots, N^x \tag{2.25}$$

Mesmo conhecendo $[K]$ e u, é ainda preciso obter $\frac{\partial [K]}{\partial x}$ através do método DMO. Esta dedução será realizada mais a frente.

Assim, é necessário resolver N^x sistemas de equações para se determinar todas as componentes do gradiente da função objectivo, sendo que o número de quações a resolver aumenta proporcionalmente com o número de variáveis de projecto.

De referir que, em cada iteração do processo de optimização, estes cálculos têm que ser feitos. Consequentemente, um elevado número de equações a resolver levará a um maior tempo de análise e elevado custo computacional, motivo pelo qual o método directo pode tornar-se bastante demorado, se existir um grande número de variáveis de projecto.

- Método adjunto

Mais uma vez, utilizando a equação (2.23) como ponto de partida, multiplica-se a derivada por um vector arbitrário não nulo, denominado deslocamento adjunto, v, [36,40]

$$\begin{aligned} & v^T \frac{\partial [K]}{\partial x} u + v^T [K] \frac{\partial u}{\partial x} - v^T \frac{\partial f}{\partial x} = 0 \quad \leftrightarrow \\ \leftrightarrow \quad & v^T [K] \frac{\partial u}{\partial x} = v^T \frac{\partial f}{\partial x} - v^T \frac{\partial [K]}{\partial x} u \end{aligned} \tag{2.26}$$

Desta feita, impõe-se o deslocamento adjunto, v, como solução do problema

$$v^T [K] = \frac{\partial C}{\partial u} \tag{2.27}$$

Isto seria o mesmo que dizer que o deslocamento adjunto é o deslocamento da estrutura para uma força adjunta igual a $\frac{\partial C}{\partial u}$.

A partir das equações (2.21), (2.26) e (2.27) obtém-se:

$$\frac{dC}{dx}(u, x) = \frac{\partial C}{\partial x} + \frac{\partial C}{\partial u} \frac{\partial u}{\partial x} = \frac{\partial C}{\partial x} + v^T [K] \frac{\partial u}{\partial x} \quad \leftrightarrow \tag{2.28}$$

$$\leftrightarrow \quad \frac{dC}{dx}(u,x) = \frac{\partial C}{\partial x} + v^T \frac{\partial f}{\partial x} - v^T \frac{\partial [K]}{\partial x} u \tag{2.29}$$

Tendo em conta que $\frac{\partial C}{\partial u} = f^T$, (ver expressão (2.20)), a expressão (2.27) fica,

$$v^T [K] = f^T \tag{2.30}$$

Como a matriz de rigidez, $[K]$, é simétrica, a expressão acima é igual à equação de equilibrio e, por isso, conclui-se que

$$v = u \tag{2.31}$$

Mais uma vez, note-se que o vector de forças, f, é independente das variáveis de projecto, pelo que $\frac{\partial f}{\partial x} = 0$.

Anulando as derivadas parciais $\frac{\partial f}{\partial x}$ e $\frac{\partial C}{\partial x}$, pelos motivos já mencionados, e substituindo v por u na expressão (2.29), obtém-se finalmente,

$$\frac{dC}{dx} = -u^T \frac{\partial [K]}{\partial x} u \tag{2.32}$$

Para permitir o cálculo da derivada da função objectivo, utilizando mais do que uma variável de projecto, a expressão final fica,

$$\frac{dC}{dx_i} = -u^T \frac{\partial [K]}{\partial x_i} u \ , \qquad i = 1,2,3,\dots,N^x \tag{2.33}$$

Conhecendo o vector dos deslocamentos, u, fica apenas a faltar perceber a forma como a matriz de rigidez global se relaciona com as variáveis de projecto, $\frac{\partial [K]}{\partial x}$. Esta derivada será deduzida no sub-capítulo que se segue. Assim, torna-se possível calcular o valor da derivada da função objectivo.

Para concluir, no método directo seria necessário recorrer-se a um sistema matricial para solucionar o problema, o que teria repercursões ao nível do tempo de cálculo e capacidade computacional necessária. No método adjunto a expressão (2.33) permite obter o valor da derivada da função objectivo através de multiplicações simples, facilitanto o cálculo final. Assim sendo, o método adjunto torna-se muito mais eficiente ao nível de esforço computacional e tempo de computação, quando o número de variáveis de projecto começa a aumentar. Por estas razões, optou-se por utilizar o método adjunto neste trabalho.

2.1.3.2 Gradiente da matriz de rigidez global

Para se conseguir obter o valor da derivada da função objectivo é essencial, como se viu na secção anterior, calcular o gradiente da matriz de rigidez global, $\frac{\partial[K]}{\partial x}$. De seguida, é apresentada a metodologia que permite obter o valor desse gradiente. Numa primeira fase, é utilizado um número N^x de variáveis de projecto sendo, posteriormente, apresentada a metodologia para o caso particular de três candidatos.

Como já foi referido, a matriz constitutiva, $[D]$, é obtida a partir do somatório de todas as matrizes constitutivas dos candidatos. No caso de três candidatos, obtém-se a seguinte expressão:

$$[D] = \sum_{i=1}^{3} w_i [D]_i = w_1[D]_1 + w_2[D]_2 + w_3[D]_3 \tag{2.34}$$

Assim, verifica-se que o número de variáveis de projecto é igual ao número de candidatos e, por isso, o vector das variáveis de projecto é dado por,

$$x = \{x_1, x_2, x_3\}^T \tag{2.35}$$

Relativamente às funções de peso, verifica-se que estas dependem das variáveis de projecto. Assim, utilizando o esquema 5 exposto anteriormente, as expressões utilizadas para o caso de três candidatos são,

$$w_1 = \frac{\hat{w}_1}{\hat{w}_1 + \hat{w}_2 + \hat{w}_3} \tag{2.36}$$

$$w_2 = \frac{\hat{w}_2}{\hat{w}_1 + \hat{w}_2 + \hat{w}_3} \tag{2.37}$$

$$w_3 = \frac{\hat{w}_3}{\hat{w}_1 + \hat{w}_2 + \hat{w}_3} \tag{2.38}$$

Onde,

$$\hat{w}_1 = (x_1)^p [1 - (x_2)^p][1 - (x_3)^p] \tag{2.39}$$

$$\hat{w}_2 = (x_2)^p [1 - (x_1)^p][1 - (x_3)^p] \tag{2.40}$$

$$\hat{w}_3 = (x_3)^p [1 - (x_1)^p][1 - (x_2)^p] \tag{2.41}$$

Passando agora à matriz de rigidez global, $[K]$, esta é obtida a partir do somatório de todas as matrizes elementares, $[K]^e$. As matrizes rigidez elementares são obtidas, teoricamente, através da expressão,

$$[K] = \sum_{e=1}^{N^e} \int_V [B]^T [D][B]\, dV \tag{2.42}$$

Ou se considerarmos um número de candidatos igual a três,

$$[K] = \sum_{e=1}^{N^e} \sum_{i=1}^{3} \int_V [B]^T w_i [D]_i [B] \, dV \quad \leftrightarrow \tag{2.43}$$

$$\leftrightarrow \quad [K] = w_1 \sum_{e=1}^{N^e} \int_V [B]^T [D]_1 [B] \, dV + w_2 \sum_{e=1}^{N^e} \int_V [B]^T [D]_2 [B] \, dV + w_3 \sum_{e=1}^{N^e} \int_V [B]^T [D]_3 [B] \, dV \tag{2.44}$$

De modo a simplicar a leitura da expressão, pode-se escrever,

$$[K] = w_1 [K]^1 + w_2 [K]^2 + w_3 [K]^3 \tag{2.45}$$

Em que,

$$[K]^1 = \sum_{e=1}^{N^e} \int_V [B]^T [D]_1 [B] \, dV \tag{2.46}$$

$$[K]^2 = \sum_{e=1}^{N^e} \int_V [B]^T [D]_2 [B] \, dV \tag{2.47}$$

$$[K]^3 = \sum_{e=1}^{N^e} \int_V [B]^T [D]_3 [B] \, dV \tag{2.48}$$

No caso prático deste trabalho, tanto as matrizes de rigidez elementar como a matriz rigidez global são obtidas a partir do programa ANSYS. No caso dos 3 candidatos, exposto na secção 2.1.2.1, considere-se que a configuração 1 é composta pelas placas 1 e 2, a configuração 2 pela placa 1 e 3 e a configuração 3 pelas placas 2 e 3.

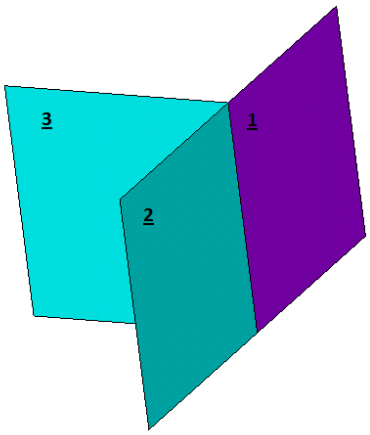


Figura 10: Configurações resultantes dos diferentes tipos de dobragem

Sabendo que w_1 está relacionado com a configuração 1, w_2 com a configuração 2 e w_3 com a configuração 3, a expressão (2.45) toma a seguinte forma,

$$[K] \text{ Placa } \underline{1} \;\leftrightarrow\; w_1\,[K]^+ + w_2\,[K]^+ + w_3\,[K]^- \tag{2.49}$$

$$[K] \text{ Placa } \underline{2} \;\leftrightarrow\; w_1\,[K]^+ + w_2\,[K]^- + w_3\,[K]^+ \tag{2.50}$$

$$[K] \text{ Placa } \underline{3} \;\leftrightarrow\; w_1\,[K]^- + w_2\,[K]^+ + w_3\,[K]^+ \tag{2.51}$$

Onde $[K]^+$ e $[K]^-$ simbolizam a matriz de rigidez de cada placa quando esta é, ou não, considerada para a análise. Isto é, no caso de se utilizar a configuração 1, a placa 3 não está a ser utilizada e por isso utiliza um material fraco que corresponde a $[K]^-$ (é como se praticamente não existisse). Para as restantes configurações o raciocínio a seguir é o mesmo.

Consideremos agora o gradiente da matriz de rigidez global, $\frac{\partial[K]}{\partial x}$. A expressão que nos permite obter o seu valor, para um número de candidatos, N^c, e um número de elementos por placa, N^e, é a seguinte,

$$\frac{\partial[K]}{\partial x} = \sum_{e=1}^{N^e}\sum_{i=1}^{N^c}\int_V [B]^T \frac{\partial w_i}{\partial x}[D]_i[B]\,dV \tag{2.52}$$

Considerando o caso de três candidatos, e utilizando as simplicações expostas atrás, verifica-se que o gradiente é obtido por,

$$\frac{\partial[K]}{\partial x} = \sum_{i=1}^{N^c}\frac{\partial w_i}{\partial x}[K]^i \quad \leftrightarrow \tag{2.53}$$

$$\leftrightarrow \quad \frac{\partial[K]}{\partial x} = \frac{\partial w_1}{\partial x}[K]^1 + \frac{\partial w_2}{\partial x}[K]^2 + \frac{\partial w_3}{\partial x}[K]^3 \tag{2.54}$$

Verifica-se então que o valor do gradiente da matriz de rigidez global é determinado através do somatório do produto entre o gradiente da função de peso e a matriz de rigidez correspondente a cada candidato. É possivel encontar um exemplo do grandiente das funções de peso no anexo A3, que se encontra nas últimas páginas deste trabalho.

Mais uma vez considerando o caso específico dos 3 candidatos, exposto na secção 2.1.2.1, a expressão que se utiliza para cada uma das placas é,

$$\frac{\partial[K]}{\partial x} \text{ Placa 1} \;\leftrightarrow\; \frac{\partial w_1}{\partial x_i}[K]^+ + \frac{\partial w_2}{\partial x_i}[K]^+ + \frac{\partial w_3}{\partial x_i}[K]^- \tag{2.55}$$

$$\frac{\partial[K]}{\partial x} \text{ Placa 2} \;\leftrightarrow\; \frac{\partial w_1}{\partial x_i}[K]^+ + \frac{\partial w_2}{\partial x_i}[K]^- + \frac{\partial w_3}{\partial x_i}[K]^+ \tag{2.56}$$

$$\frac{\partial[K]}{\partial x} \text{ Placa 3} \;\leftrightarrow\; \frac{\partial w_1}{\partial x_i}[K]^- + \frac{\partial w_2}{\partial x_i}[K]^+ + \frac{\partial w_3}{\partial x_i}[K]^+ \tag{2.57}$$

Sendo que $x_i = \{x_1, x_2, x_3\}^T$.

2.1.4 Constrangimentos

Em alguns problemas de optimização é necessário impôr limites ou constrangimentos ao projecto. Estes constrangimentos podem ser de dois tipos, constrangimentos de igualdade, h_j, ou constrangimentos de desigualdade, g_i, [25].

$$h_j(x) = 0, \quad j = 1,2,3,\dots,n \tag{2.58}$$

$$g_i(x) \leq 0, \quad i = 1,2,3,\dots,m \tag{2.59}$$

Existe ainda outro tipo de constrangimentos, que serão utilizados neste trabalho, e são denominados constrangimentos laterais. Neste caso são definidos valores limite e o optimizador apenas verifica se esse valor é ultrapassado ou não, garantindo desta forma, que a solução óptima não fica fora destes limites. O limite inferior imposto é zero e o superior é 1 e serão aplicados às variáveis de projecto,

$$0 < x_i < 1 \tag{2.60}$$

2.2 Formulação genérica utilizada

De seguida é apresentada, de forma resumida, a formulação utilizada neste trabalho.

Variáveis de projecto:

$$x \in [0,1], \quad x = \{x_1, x_2, x_3, x_4, x_5, \dots, x_{N^c},\}^T \tag{2.61}$$

Função objectivo:

$$min_{x \in [0,1]}\, C(u,x) \tag{2.62}$$

$$C(x,u) = f^T u \tag{2.63}$$

$$\frac{dC}{dx}(u,x) = f^T \frac{\partial u}{\partial x} \tag{2.64}$$

2.3 Algoritmos de optimização

Depois de elaborada a formulação do problema é necessário escolher o algoritmo de optimização. Neste sub-capítulo é feita uma breve introdução aos algoritmos de optimização.

Um algoritmo de optimização não é mais que um algoritmo que segue um processo iterativo onde a estrutura em questão é analisada, repetidamente, até que se obtenha a solução final que seja considerada óptima, de entre todas as possibilidades. De seguida

apresenta-se o fluxograma ilustrativo de um processo de optimização, de forma a facilitar a compreensão do que já foi dito.

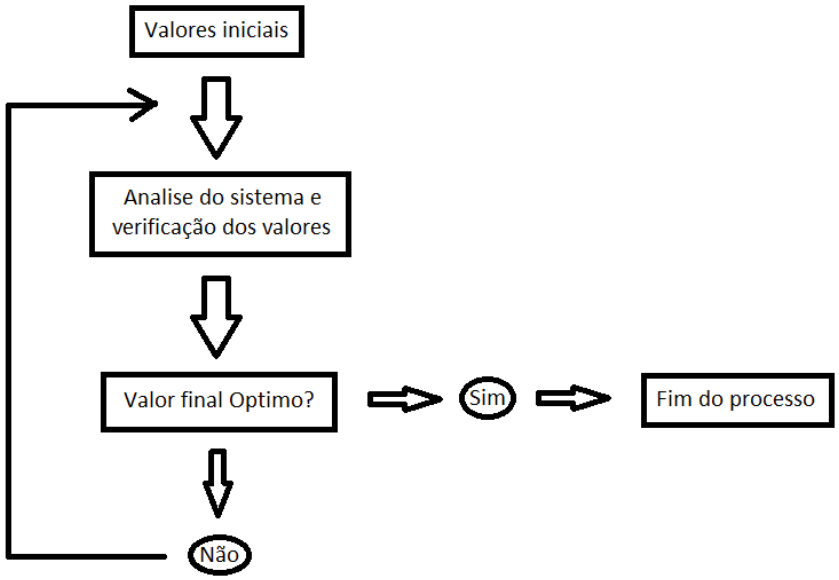


Figura 11: Fluxograma de um processo de optimização

Como é possivel verificar na imagem anterior, o processo começa com a atribuição de alguns valores iniciais para as variáveis. Segue-se então a análise do sistema e, posteriormente, é verificada a solução do problema. Se a solução for a pretendida (solução óptima) o processo é finalizado, se pelo contrário se verificar que a solução não é óptima, o processo é repetivo com outros valores iniciais.

Os algoritmos de optimização podem utilizar diferentes métodos com o objectivo de obter o valor óptimo para a função objectivo.

- Métodos de ordem zero
- Métodos de 1ª ordem
- Métodos de 2ª ordem

Os métodos de ordem zero são aqueles que utilizam apenas do valor da função objectivo para o processo iterativo. Já os métodos de 1ª ordem, para além do valor da função objectivo utilizam também o gradiente desta função em ordem às variáveis de projecto, isto é,

$$\nabla f(x) = \left(\frac{df}{dx}\right) = \left(\frac{df}{dx_1}, \frac{df}{dx_2}, \frac{df}{dx_3}, \dots, \frac{df}{dx_{N^x}}\right) \tag{2.65}$$

No caso dos métodos de 2ª ordem é necessário também o valor da matriz hessiana, ou seja,

$$\mathrm{H}(x) = \left(\frac{d^2 f}{dx^2}\right) = \begin{bmatrix} \frac{d^2 f}{dx^2} & \frac{d^2 f}{dx_1 x_2} & \cdots \\ \frac{d^2 f}{dx_1 x_2} & \ddots & \vdots \\ \cdots & \cdots & \frac{d^2 f}{dx_N x^2} \end{bmatrix} \tag{2.66}$$

Existem ainda duas classificações possíveis relativamente ao tipo de algoritmo que é utilizado.

- *Métodos Estocásticos (não determinísticos)*
- *Métodos Determinísticos*

Os métodos estocásticos ou probabilísticos usam apenas a função objectivo e introduzem parâmetros estocásticos no processo de optimização [41]. Como não utilizam a derivada da função objectivo são considerados métodos de ordem zero. Estes métodos apresentam algumas vantagens relativamente aos determinísticos.

- A função objectivo e os constrangimentos não necessitam de ter uma representação matemática.
- A função objectivo não necessita ser contínua ou diferenciável.
- Permitem utilizar parâmetros contínuos, discretos o uma combinação destes.
- Não apresentam restrições quanto ao ponto de partida dentro do espaço de busca da solução

Apesar disto, estes métodos necessitam de um tempo de processamento superior em comparação com os métodos clássicos.

Os métodos determinísticos geram uma sequência de soluções possíveis sendo que, na maioria das vezes, utilizam pelo menos a primeira derivada da função objectivo. Nestes métodos a função objectivo deve ser contínua e diferenciável e, tal como os constrangimentos, deve ser dada na forma de uma função matemática [41]. Nos métodos determinísticos a solução encontrada depende bastante do ponto de partida e, por isso, a solução óptima encontrada é muitas vezes local e não global. Assim, estes métodos apresentam teoremas que lhes garantem a convergência para uma solução óptima (não necessariamente global).

A maioria dos métodos clássicos são determinísticos e podemos dividi-los em dois grupos, de acordo com as características da função objectivo.

- *Lineares* – função objectivo e constrangimentos são funções lineares das variáveis
- *Não-Lineares* – função objectivo ou pelo menos um dos constrangimentos é uma função não-linear das variáveis de projecto.

O FAIPA (Feasible Arc Interior Point Algorithm) será um dos optimizadores utilizados neste trabalho. Este optimizador permite resolver problemas com constrangimentos não

lineares e requer um ponto de partida localizado no interior dos constrangimentos impostos [42]. O FAIPA localiza o mínimo da função objectivo utilizando as condições de óptimo de Karush-kuhn-Tucker.

Para além do FAIPA, foi utilizada também a função FMINCON, optimizador que pertence à *optimization toolbox* do MATLAB. Esta função também permite resolver problemas não lineares constrangidos. Neste caso será utilizado um algoritmo de optimização *large-scale*, que é um subespaço do método *trust-region* e que se baseia no método *interior-reflective Newton* [43]. A descrição destes métodos não será feita nesta tese, no entanto pode ser consultada nas referências [8] e [43].

Capítulo 3

Programa Computacional

Este capítulo consiste numa explicação do programa computacional desenvolvido. Numa primeira fase é descrito o funcionamente da parte de elementos finitos do programa, realizada através do software ANSYS. Posteriormente, é explicitado o algoritmo de optimização, tal como o funcionamente da interface entre o ANSYS e o MATLAB.

3.1 Análise ANSYS

A análise realizada em ANSYS consiste na resolução de um problema de elementos finitos, com o objectivo de se obter o valor da energia elástica de deformação. Esta energia é escrita num ficheiro *.txt* e utilizada pelo optimizador, de modo a que este calcule o valor da *Compliance, C e* faça com que esta função atinja o seu valor óptimo.

O ficheiro *.txt* utilizado para correr o software ANSYS recebe, a partir do MATLAB, os valores dos módulos de Young que são utilizados na análise. A forma como esses módulos de Young são calculados é explicada no subcapítulo 3.2 . O ANSYS actua sem a intervenção directa do utilizador, uma vez que a interface entre este e o MATLAB permite correr o ANSYS em *batch mode*. A forma como é realizada esta ligação é explicada no capítulo correspondente à interface entre ANSYS e MATLAB. Tanto os ficheiro de *input* como os ficheiros de *output* do programa ANSYS são ficheiros *.txt .*

Visto a análise ser realizada para uma placa plana, numa primeira fase e, posteriormente, para uma placa cilíndrica, ambos os casos são explicados nos subcapítulos seguintes. Os ficheiros do programa onde se recorre ao optimizador FAIPA e que corre a análise para um painel plano encontram-se na pasta “Programa Hugo Pina Casos de estudo”. Já os ficheiros onde é utilizada a função FMINCON, tanto na análise da estrutura plana como da cilíndrica encontran-se na pasta intitulada “Programa Hugo Pina_FMINCON”.

3.1.1 Pré-processamento

De forma a realizar uma correcta análise utilizando o software ANSYS é necessário seguir alguns passos essenciais, nesta fase de pré-processamento. Eles são os seguintes:

- Escolha do elemento a utilizar
- Definir as constantes do elemento

- Definir as propriedades de cada material
- Escolha da secção e geometria do elemento
- Definir os keypoints, linhas e áreas
- Associar os elementos às áreas
- Definir a malha
- Aplicar as cargas e condições de fronteira

A escolha do elemento certo para a análise em questão é essencial. Essa escolha depende não só do tipo de análise que se pretende fazer, mas também dos resultados que se quer obter. De entre as várias opções disponíveis no ANSYS, os elementos tipo casca (SHELL) são aqueles que mais se adequam ao que está a ser estudado neste trabalho. Dentro das opções possíveis, optou-se por utilizar o elemento SHELL181.

O elemento SHELL181 é constituido por quatro nós, cada um deles com seis graus de liberdade (deslocamentos em x, y e z e rotações em x, y e z).

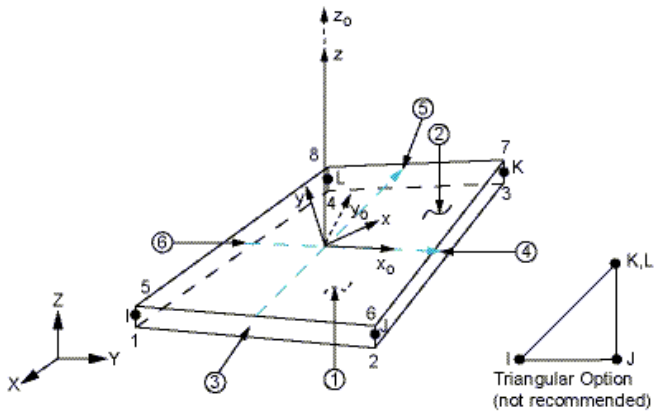


Figura 12: Elemento SHELL181

Este elemento é indicado para modelar compósitos sanduíche e baseia-se numa teoria de primeira ordem, a Teoria de Mindlin-Reissner para cascas. A definição deste elemento no ficheiro *.txt* é feita através do comando *ET*. Neste comando o primeiro espaço a preencher corresponde ao número do elemento e o segundo ao nome do mesmo.

As constantes do elemento e propriedades de cada material são definidades pelos comandos *R* e *MP*, respectivamente. No caso das propriedades do material, utilizando o comando *MP* é possivel definir o módulo de Young, coeficiente de Poisson, densidade do material, entre outros.

O passo seguinte consiste na escolha da secção e geometria do elemento. Nesta fase utiliza-se o comando *SECTYPE* para associar o tipo de secção com o seu número de

identificação. O comando *SECDATA* permite descrever a geometria da secção, isto é, define valores como a espessura, o número do material ou o número de pontos de integração.

Depois de definidas estas propriedades chega-se à fase de construção do modelo. Nesta fase, o primeiro passo é definir as coordenadas dos *keypoints* que serão a base do modelo. É importante referir que, na análise da placa cilíndrica utiliza-se o comando *CSYS* para permitir utilizar coordenadas cilindricas na definição dos *keypoints*.

Depois de definidos os *keypoints*, cria-se as linhas que unem os vários *keypoints.* Após a criação das linhas, é possível definir as áreas que farão parte do modelo. Estes passos são realizados através dos comandos *K, L, LARC* (para linha curvas), *A* (criar áreas a partir dos *keypoints*) *e AL* (criar áreas a partir das linhas), respectivamente. O comando *K* é seguido do número e das coordenadas do respectivo *keypoint*, o comando *L* e *LARC* são seguidos do número dos keypoints que se pretende ligar e, por fim, aos comandos *A* e *AL* seguem-se o número dos keypoints, ou linhas, que delimitam a respectiva área.

Após a criação das áreas pretende-se associar a cada área um tipo de material. Para isso, utiliza-se o comando *ASEL*, para selecionar as áreas pretendidas e o comando *AATT* para definir o material a utilizar nessas áreas.

A malha a utilizar deve ser escolhida de modo a que haja uma boa relação entre o tempo de computação e a precisão da solução. Assim, optou-se por utilizar apenas elementos rectangulares ou quadrangulares. O número de divisões das linhas é definido no terceiro campo do comando *L*, e permite obter uma malha com mais ou menos elementos, isto é, com maior ou menor precisão. O comando *AMESH* permite criar a malha da estrutura em análise.

Por fim, a última fase deste pré-processamento consiste na aplicação das cargas e das condições de fronteira à estrutura em análise. Os constrangimentos podem ser aplicados a linhas, áreas ou keypoints, sendo que se tem de seleccionar os elementos pretendido. No caso dos elementos seleccionados serem linhas, utiliza-se o comando *DL* para aplicar o constrangimento. Passando agora às cargas aplicadas, o comando *SFA* permite aplicar uma pressão uniforme ao longo das áreas seleccionadas.

Depois de realizada esta fase de pré-processamente deve-se fazer com que o programa calcule a solução do problema. Para isso utiliza-se os comandos, *FINISH - /SOL - /STATUS,SOLU* e por fim *SOLVE*. Após a definição destes comandos a solução é calculada e é possível passar à fase de pós-processamento.

No anexo A é possível observar o ficheiro APDL de um dos casos de estudo desta tese.

3.1.2 Pós-processamento

A fase de pós-processamento consiste em solicitar ao ANSYS que escreva num ficheiro *.txt* todos os valores que são necessários para o desenvolvimento do projecto. Desta forma, é necessário que o ANSYS escreva num ficheiro os seguintes dados:

- Matriz de rigidez de cada elemento
- Energia Elástica de deformação
- Nós correspondentes a cada elemento
- Graus de liberdade de cada nó

De referir que, para escrever num ficheiro .txt toda a informação necessária, obtida em ANSYS, é preciso aplicae os seguintes comandos:

*/POST1 - *CREATE,com - *CFOPEN -*

**GET - *VWRITE - código ASCII - *CFCLOS - *END - /INPUT,com*

Depois de abrir a directoria /POST1, o comando *CREATE,com permite abrir a macro do ficheiro, utilizando-se o *CFOPEN para abrir o referido ficheiro directoria pretendida. O comando *GET permite transformar valores em parâmetros escalar e o *VWRITE escreve a informação pretendido no ficheiro. Por fim, o comando *CFCLOS fecha o ficheiro e o *END fecha a macro do mesmo.

De seguida é explicado como se obtém cada um dos valores referidos acima.

3.1.2.1 Matrizes de Rigidez

As matrizes de rigides de cada elemento são essenciais para o cálculo do gradiente da matriz de rigidez global e, consequentemente, da derivada da função objectivo. A partir da análise realizada no software ANSYS é possível obter estas matrizes.

De modo a obter as matrizes de rigidez de cada elemento é necessário utilizar os comandos seguintes para criar um ficheiro *.emat* (ficheiro com a matriz dos elementos):

/SOL - ematwrite,yes - /STATUS,SOLU - SOLVE

Depois disto, utiliza-se o *DMAT e *PRINT para, respectivamente, se criar uma matriz densa e escrever os valores dessa matriz num ficheiro *.txt* .

Utilizando o comando *HBMAT*, da directoria */AUX2*, é possível obter a matriz de rigidez global no formato Harwell-Boeing. Este formato apresenta a matriz na forma de matriz esparsa. Depois de escrita no ficheiro *.txt* a matriz é então modificada no MATLAB e apresentada na sua forma normal.

3.1.2.2 Energia elástica de deformação

Como já foi explicado no capítulo 2, a função objectivo é obtida através da seguinte expressão:

$$C = f^T u = u^T [K] u = 2U \tag{3.1}$$

Assim, a forma mais fácil de obter o valor da *compliance C* é através da energia. Como já foi referido, a energia pode ser obtida a partir da análise realiza em ANSYS. Através do comando *ETABLE* é possível obter o valor desta energia para cada um dos elementos da estrutura. De seguida, aplica-se o comando *SSUM* para que seja feito o somatório da energia de todos os elementos, pois

$$U = \sum_{e=1}^{N^e} U^e \tag{3.2}$$

A energia é, por fim, escrita no ficheiro e utilizada pelo MATLAB.

3.1.2.3 Nós por elemento

Os número dos nós que pertencem a cada elemento é utilizado pelo MATLAB para o cálculo do vector dos graus de liberdade de cada elemento, u_e. De forma a obter este parâmetro, é necessário realizar um pequeno ciclo no ANSYS. Deve-se começar por utilizar o comando *CM*, que permite agrupar vários items num só componente. De seguida utiliza-se o comando **GET* seguido do **DIM*. Este último permite definir os parâmetros e dimensões de uma matriz.

Após estes comandos iniciais, utiliza-se o comando **DO* para iniciar o ciclo. Dentro do ciclo usa-se o *CMSEL* para selecionar o conjunto de componentes pretendido e o comando **GET* para passar esses componentes para o local pretendido na matriz. Por fim, utiliza-se o comando **ENDDO* para terminar o ciclo.

3.1.2.4 Graus de liberdade

Os graus de liberdade referentes a cada nó são utilizados pelo MATLAB para construir o vector u_e. Estes valores são obtidos no ANSYS através de um ciclo idêntico ao dos nós por elemento sendo que, neste caso, os items utilizados serão os graus de liberdade de cada nó e não o número dos nós que fazem parte de cada elemento.

3.2 Algoritmo de Optimização

O algoritmo de optimização utilizado neste trabalho recorre ao método adjunto e ao DMO para determinar a melhor forma de utilizar os recortes da chapa superior da estrutura, de modo a maximizar a rigidez da mesma. O processo em questão é apresentado na figura que se segue, sendo que em (a) é apresentado o algoritmo de optimização e em (b) é clarificado a implementação do método adjunto e DMO.

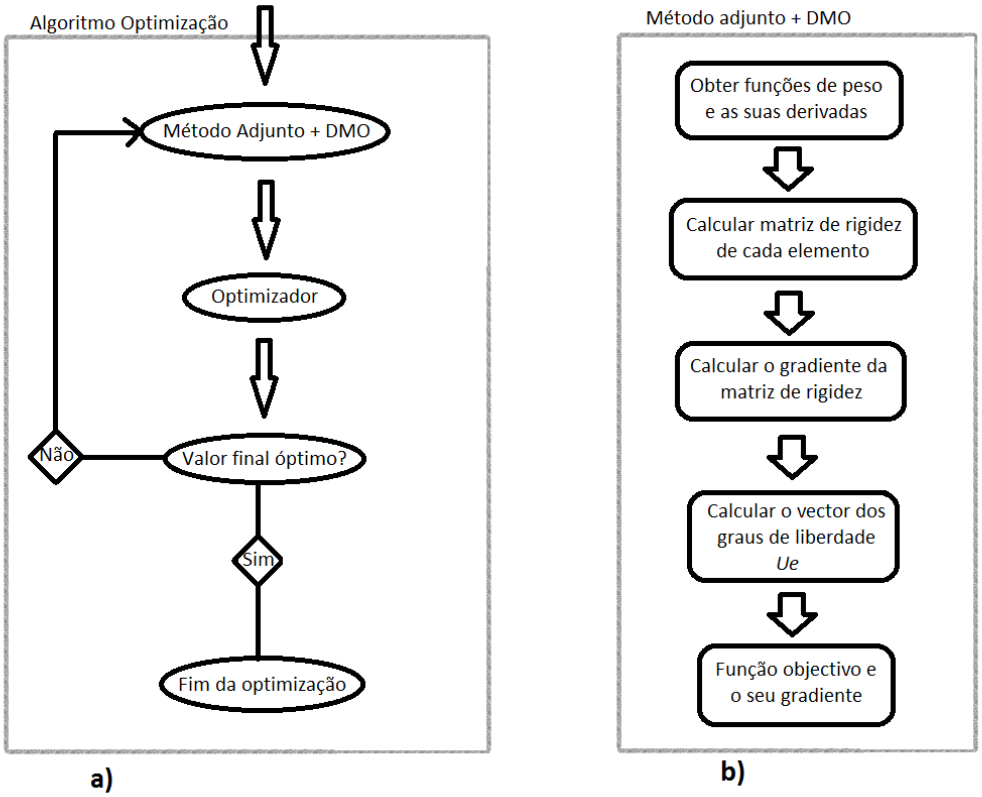


Figura 13: a) Algoritmo de optimização b) Método adjunto + DMO

Como já foi referido no capítulo anterior, um dos optimizador utilizado neste trabalho é o FAIPA [44], (Feasible Arc Interior Point Algorithm). Este optimizador permite resolver problemas com constrangimentos não lineares e requer um ponto de partida localizado no interior dos constrangimentos impostos de forma a resolver as condições de óptimo de Karush-Kuhn-Tucker [42].

Para além do FAIPA, recorreu-se também à utilização do optimizador FMINCON da toolbox do MATLAB [45]. Esta função também permite resolver problemas não lineares constrangidos. Ambos os optimizadores necessitam que se defina parâmetros como critérios de paragem, algoritmo de optimização, entre outros. Pelos motivos acima mensionados optou-se por utilizar estes optimizadores, em deterimento de outros.

3.3 Interface ANSYS – MATLAB

Neste subcapítulo é apresentada a interface entre os dois programas utilizados nesta tese, o ANSYS e o MATLAB. Esta interface, realizada a partir do MATLAB, permite que os dados obtidos neste programa sejam utilizados no ANSYS e os *outputs* obtidos em ANSYS utilizados para os cálculos do algoritmo de optimização, no MATLAB.

De seguida é apresentado um diagrama que esquematiza a forma como interagem os dois softwares.

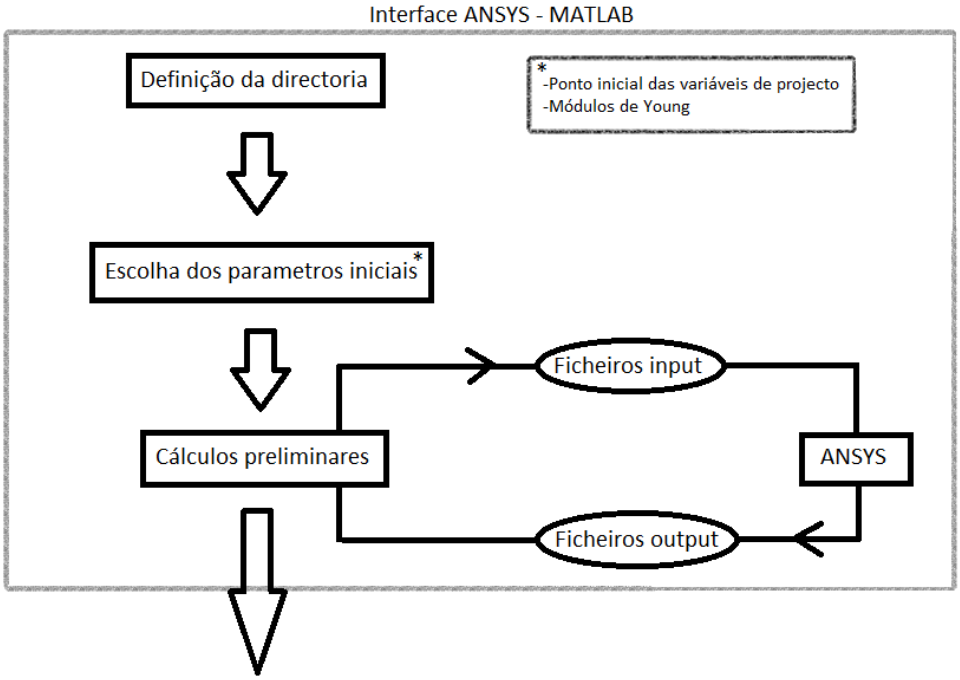


Figura 14: Interface ANSYS - MATLAB

No MATLAB foram desenvolvidas várias funções que permitem realizar todos os cálculos preliminares necessários para a resolução do problema. Esses cálculos ocorrem após a definição da directoria e escolha dos parâmetros inciais. Utilizando o FAIPA como optimizador, o programa é composto pelo ficheiro *fun999,* sendo este constituido pelas seguintes funções:

- *MatrizK_Ncand_AND_Vector_F_Ncand*
- *DMO_Scheme_5_Funcao_AND_Derivadas*
- *SacaEnergia*

O ficheiro designado *MatrizK_Ncand_AND_Vector_F_Ncand* permite, a partir dos valores obtidos do ANSYS, calcular o as matrizes de rigidez de cada elemento e a matriz de rigidez global.

O ficheiro *DMO_Scheme_5_Funcao_AND_Derivadas* calcula o valor das funções de peso e das suas derivadas em ordem às variáveis de projecto. Neste caso, o número presente no nome do ficheiro varia consoante o esquema do DMO que se está a utilizar.

Por último, a função *SacaEnergia* solicita que o ANSYS corra em *batch mode* e permite obter o valor da energia elástica de deformação.

Depois de obtidos todos os parâmetros referidos atrás, são feitos os cálculos necessários para se obter o valor da função objectivo - *compliance, C*.

O ficheiro *gfun999* permite obter o valor da derivada da função objectivo em ordem às variáveis de projecto. Este ficheiro é composto pelas seguintes funções:

- *MatrizK_Ncand_AND_Vector_F_Ncand*
- *DMO_Scheme_5_Funcao_AND_Derivadas*
- *VectorU*
- *DERIVADA_MatrizK_ORDEM_x*

O ficheiro *VectorU* permite obter o vector dos deslocamentos de cada elemento, a partir dos resultados obtidos no ANSYS.

A função *DERIVADA_MatrizK_ORDEM_x* recebe as matrizes de rigidez de cada elemento e as derivadas das funções de peso e calcula a derivada da matriz de rigidez em ordem a cada uma das variáveis de projecto.

Após estes cálculos, o ficheiro *gfun999* calcula o valor da derivada da função objectivo em ordem às variáveis de projecto.

Na fase seguinte do trabalho, utiliza-se o optimizador FMINCON. Neste caso, as seguintes funções encontram-se dentro do script *prob_999* :

- *MatrizK_Ncand_AND_Vector_F_Ncand*
- *sacaKelem (caso painel cilíndrico)*
- *METODO_ADJUNTO_Funcao_Objectivo_Compliance*

O ficheiro *METODO_ADJUNTO_Funcao_Objectivo_Compliance* é composto por várias funções,

- *DMO_Scheme_5_Funcao_AND_Derivadas (número dependo do esquema DMO utilizado)*
- *SacaEnergia*
- *VectorU*
- *DERIVADA_MatrizK_ORDEM_x*

As funções utilizadas neste caso servem o mesmo propósito que no caso em que se recorre ao FAIPA.

3.4 Como correr o programa

É possível correr o programa desenvolvido nesta tese em qualquer computador. Para isso é necessário ter instalados os softwares ANSYS e MATLAB.

Antes de correr qualquer dos casos analisados é necessário alterar as directorias utilizadas para guarda os ficheiros de *input, output* e as directorias para correr o software ANSYS em *batchmode.* Isto deve ser feito tanto nos ficheiros APDL denominados "SacaEnergia1buraco" como nas funções *fun999* e *gfun999* (caso do FAIPA) ou *prob999* (caso do FMINCON).

Depois de alteradas as directorias, é possivel correr qualquer análise. Para analisar os estudos realizados com recurso ao FAIPA deve-se abrir a pasta "Programa Hugo Pina Casos de estudo", escolher o caso que se pretende analisar, abrir a função *prob999* e correr o programa.

No caso dos estudos realizados com recurso ao FMINCON o procedimento é idêntico. Neste caso deve abrir-se a pasta "Programa Hugo Pina_FMINCON", escolher o caso que se pretende analisar, abrir a função *prob999* e correr o programa.

Capítulo 4

Resultados e Discussão

Neste capítulo são expostos os resultados obtidos ao longo do desenrolar desta dissertação. Os resultados são apresentados segundo a sequência natural de trabalho, isto é, dos casos mais simples para os mais complexos. Inicialmente, apresenta-se o caso em que são utilizadas apenas três variáveis de projecto. De seguida, são expostos os resultados obtidos para outros casos de estudo, onde é utilizado um maior número de variáveis de projecto.

Os problemas em estudo têm como objectivo optimizar os reforços e buracos numa estrutura tridimensional, composta por dois paineis, sendo o principal objectivo maximizar a rigidez da estrutura.

O modelo de estudo dos próximos subcapítulos será uma placa quadrada com $4\,m$ de lado, composta por dois painéis paralelos com $1\,m$ de distância entre ambos. Ao painel superior da chapa é aplicado um corte tipo H ou I, sendo os recortes dobrados e soldados ao painel inferior (reforços). O furo aplicado ao painel também será quadrado, tendo $2\,m$ de lado.

Nos casos de estudo onde se pretende estudar o efeito do aumento do número de buracos da placa, o modelo acima descrito é repetido n vezes, nas direcções x e y, mantendo a estrutura quadrada.

Como já foi referido no capítulo anterior, optou-se por utilizar o elemento SHELL181, contituido por quatro nós e seis graus de liberdade por nó. Para os dois painéis será utilizado um módulo de elasticidade $E = 200x10^9$. Já para os reforços, será utilizado $E = 200x10^9$, no caso de terem função de peso $w_i = 1$, e $E = 200x10^3$ para $w_i = 0$.

Numa primeira fase será utilizado o optimizador FAIPA. Neste caso, o programa prossegue com o processo iterativo até que uma das condições de paragem seja atingida (ex: variação da função objectivo, ou da norma do gradiente, inferior a 10^{-5}). Os valores de paragem irão variar de análise para análise, tendo em conta o valor da função objectivo obtido em cada caso. Se, por exemplo, a função objectivo de determina análise for na ordem de 10^{-2} a condição de paragem para a função objectivo será uma variação inferior a 10^{-7}.

Todas as análises foram realizadas num computador com as seguintes características:

- Intel(R) Core(TM) i7-2600K CPU@3.40GHz 8GB memória

4.1 Caso 1 - Utilização do método DMO

4.1.1 Modelo com 3 variáveis de projecto

Inicialmente, optou-se por desenvolver um programa que utilizasse apenas três variáveis de projecto, de modo a que fosse mais fácil verificar se este funcionava correctamente ou não. Com apenas três opções de escolha é possivel correr cada uma das configurações possibilidades e verificar se o programa converge para a melhor solução.

De modo a diminuir o tempo de cálculo e processamento do programa, a malha desenvolvida nesta fase utiliza apenas um elemento por área gerada.

Na imagem que se segue é possível verificar os três candidatos que se utilizou neste primeiro caso. O corte na placa superior é sempre igual, sendo que o que varia é a forma como se dobra os reforços.

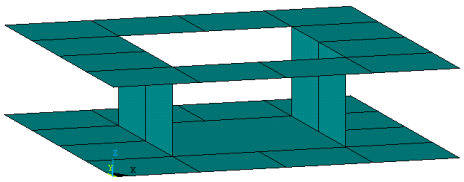


Figura 15: Candidato 1

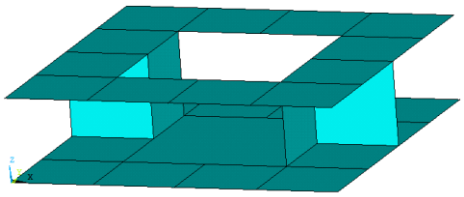


Figura 16: Candidato 2

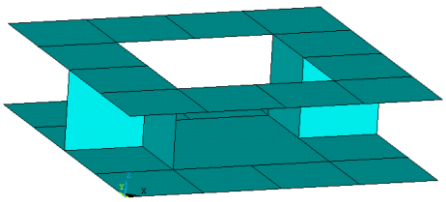


Figura 17: Candidato 3

Nesta primeira fase começou-se por aplicar uma pressão uniforme, de $150\ N/m^2$, na placa inferior (z=0) e encastrar a placa superior (z=1) nas suas extremidades y=0 e y=4.

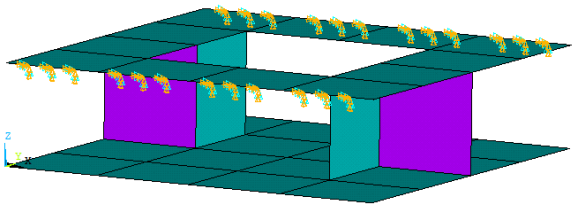


Figura 18: Estrutura com a placa superior encastrada em y=0 e y=4

Depois de algumas análises, verificou-se que os resultados obtidos não eram concordantes com o previsto. Constatou-se que as soluções não convergiam e os seus valores encontravam-se afastados das soluções pretendidas, isto é, uma das três funções de peso igual a 1 e as restantes iguais a 0.

Observou-se que, o facto de apenas a placa superior estar constrangida poderia estar a criar problemas na análise, levando o optimizador a preferir soluções "cinzentas" ou intermédias, em detrimento das soluções 1 ou 0. Neste caso, a placa não constrangida apresenta deformadas muito elevadas, o que pode ser um problema para a análise.

De modo a solucionar este problema, optou-se por constrangir a placa inferior da mesma forma que a superior, como é possível observar na imagem seguinte. Com esta alteração nas condições de fronteira, as soluções começaram a enquadrar-se com o previsto.

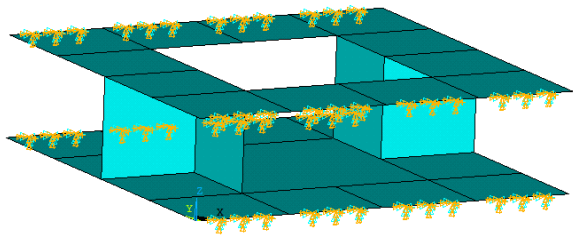


Figura 19: Estrutura com as placas superior e inferior encastradas em y=0 e y=4

De seguida são apresentadas algumas análises e respectivos resultados obtidos neste caso.

4.1.1.1 Escolha do esquema de DMO

Nesta fase inicial é importante analisar e verificar qual o esquema de DMO que melhor se enquadra no problema. Assim, realizaram-se algumas análises de modo a verificar qual a melhor opção.

Começou-se por utilizar o esquema 4 do DMO, com um factor de penalização igual a dois, $p = 2$, e os seguintes valores iniciais para as variáveis de projecto, $x_0 = [0.5\ 0.5\ 0.5]$. Neste caso, verificou-se que o optimizador tem alguma dificuldade em escolher entre o segundo e terceiro candidatos, apresentando o primeiro candidato como a pior solução. Esta dificuldade deve-se à simetria que o problema apresenta, que faz com que a solução óptima seja obtida tanto para o candidato dois, como para o três.

Posto isto, optou-se por "favorecer" um destes dois candidatos, utilizando-se como ponto de partida os valores $x_0 = [0.7\ 0.6\ 0.4]$. Neste caso, o programa já consegue apresentar uma solução óptima.

De forma a ser possível comprovar que o programa está a funcionar de acordo com o previsto, testou-se cada um dos 3 candidatos directamente no software ANSYS, tendo-se obtido, em cada um dos casos, os seguintes valores para a função objectivo.

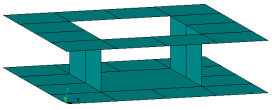
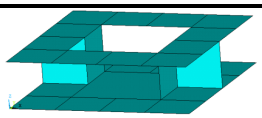
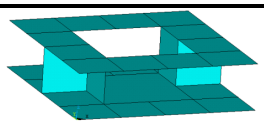
 $w1 = 1$	***C = 0.34672992900486 J***
 $w2 = 1$	***C = 0.01752390021923 J***
 $w3 = 1$	***C = 0.01752390021923 J***

Tabela 1: Solução obtida directamente do ANSYS

De seguida, procedeu-se à análise do problema através do programa desenvolvido. No caso em questão, $x_0 = [0.7\ 0.6\ 0.4]$, e utilizando o esquema 4 do DMO, o programa apresenta o candidato dois como a melhor solução. Aplicando o esquema 5 ao mesmo problema, os candidatos dois e três evidenciam valores idênticos nas funções de peso, o que denota uma incapacidade do programa em escolher entre estas duas opções devido à simetria do problema. Tendo em conta este problema, a solução do problema utilizando o esquema 5 torna-se irrealista. Os resultados obtidos nestas análises encontram-se expostos na tabela seguinte.

$p = 2$ $x_0 = [0.7\ 0.6\ 0.4]$	$C\ [J]$	Iterações até convergir
DMO 4	0.0184568736671	32
DMO 5	0.0147606985622	35

Tabela 2: Solução da função objectivo usando o esquema 4 e 5 do DMO

$p = 2$ $x_0 = [0.7\ 0.6\ 0.4]$	$w1$	$w2$	$w3$
DMO 4	1.51308e-11	**0.954098**	3.81115e-4
DMO 5	3.12506e-9	0.500319	0.499681

Tabela 3: Solução das funções de peso usando o esquema 4 e 5 do DMO

Pela análise das tabelas anteriores verifica-se que o resultado obtido através do esquema 4 do DMO apresenta um erro de 5.3% em relação ao valor óptimo, calculado

directamente do software ANSYS. Já o esquema 5 denota novamente uma incapacidade em escolher apenas um dos candidatos, preferindo apresentar uma solução “cinzente”.

4.1.1.2 Escolha do factor de penalização *p*

Como já foi referido na secção 2.1.2.2, o factor de penalização utilizado no método DMO permite penalizar os valores intermédios das funções de peso, levando a que estes se aproximem dos seus valores extremos, 1 e 0. Assim, é importante averiguar qual o factor de penalização mais adequado ao problema em questão.

Tal como nas análises anteriores, optou-se por utilizar como valores iniciais para as variáveis de projecto, $x_0 = [0.7\ 0.6\ 0.4]$. A análise foi iniciada com factores de penalização mais baixos, $p = 2$, e foi sendo aumentado sucessivamente. Os resultados obtidos estão expostos na tabela que se segue.

DMO 4 $x_0 = [0.7\ 0.6\ 0.4]$	$C\ [J]$	$w1$	$w2$	$w3$	Iterações até convergir
$p = 2$	0.0184568736671	1.51308e-11	**0.954098**	3.81115e-4	32
$p = 4$	0.0219889203449	1.92972e-3	0.782485	3.62952e-4	27
$p = 6$	0.0231597068029	2.12322e-3	0.732977	3.94464e-5	28

Tabela 4: Função objectivo e funções de peso com p=2, p=4 e p=5

Observa-se que o caso em que a solução se aproxima mais do previsto é para $p = 2$, apesar de convergir mais lentamente. Os restantes ensaios denotam alguma dificuldade em aproximar as funções de peso dos seus extremos. Isto deve-se à limitação que o esquema 4 do DMO apresenta, isto é, não garante que o somatório das funções de peso seja igual a 1. Assim, verifica-se que o aumento de p induz um afastamento do valor da função de peso do melhor candidato do seu valor extremo, 1.

4.1.2 Modelo com 6 variáveis de projecto

Após as análises realizadas com três variáveis de projecto passou-se ao estudo do modelo composto por seis variáveis de projecto. As condições de fronteira e o número de elementos foram mantidos iguais. Neste caso existe um maior número de combinações possíveis, no entanto este número continua a ser reduzido. Assim, é possível mais uma vez verificar os resultados directamente no software ANSYS e observar se estes estão de acordo com o previsto.

No caso em análise as seis variáveis de projecto são divididas em dois grupos de três variáveis, sendo estes grupos independentes entre si, isto é, pretende-se que cada um dos grupos tenha apenas uma função de peso a 1 e as restantes a 0. Na imagem que se segue é possivel observar as três configurações possíveis para os reforços, tanto no lado esquerdo como no lado direito da placa.

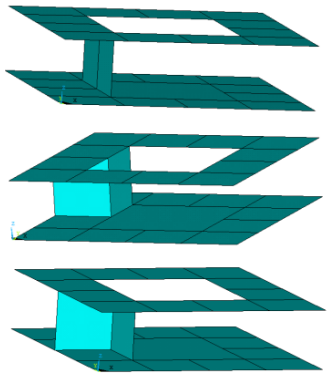
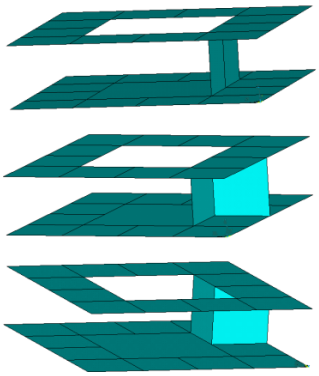
Lado esquerdo		*Lado direito*	
	w_1		w_4
	w_2		w_5
	w_3		w_6

Tabela 5: Candidatos possíveis com 6 variáveis de projecto

Como forma de verificar se o programa permite obter a solução óptima prevista recorreu-se, mais uma vez, a uma análise directa no software ANSYS. Na tabela seguinte estão expostos os resultados obtidos para cada um dos candidatos.

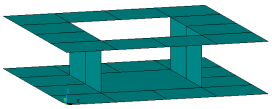
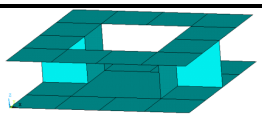
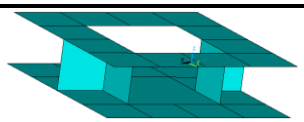
 $w1 = 1$ e $w4 = 1$	***C = 0.34672992900486 J***
 $w1 = 1$ e $w5 = 1$ // $w1 = 1$ e $w6 = 1$ $w2 = 1$ e $w4 = 1$ // $w3 = 1$ e $w4 = 1$	***C = 0.18754972130672 J***
 $w2 = 1$ e $w5 = 1$ // $w3 = 1$ e $w6 = 1$	***C = 0.01752390021923 J***
 $w2 = 1$ e $w6 = 1$ // $w3 = 1$ e $w5 = 1$	***C = 0.01783825647957 J***

Tabela 6: Solução obtida directamente do ANSYS

De referir que, devido à simetria do problema, existem várias combinações com soluções idênticas, tal como é possivel observar pela tabela acima.

Constata-se que a solução óptima, quando se utilizam seis variáveis de projecto, é o caso em que temos $w_2 = 1$, $w_5 = 1$ e as restantes funções de peso iguais a 0 (ou por simetria $w_3 = 1$, $w_6 = 1$). Esta é a solução que se espera atingir com o programa desenvolvido.

4.1.2.1 Escolha do esquema de DMO

Tal como no caso de três variáveis, começou-se por analisar os resultados obtidos quando se utilizava o mesmo valor inicial para todas as variáveis de projecto, $x_0 = [0.5\ 0.5\ 0.5\ 0.5\ 0.5\ 0.5]$. Neste caso, tanto utilizando o esquema 4 como o esquema 5 do DMO, o programa apresenta dificuldade em optar pelas soluções w_2, w_3 e w_5, w_6 . Os resultados obtidos estão expostos da tabela seguinte,

$p = 4$ $x_0 = [0.5\ 0.5\ 0.5\ 0.5\ 0.5\ 0.5]$	$C\ [J\ ]$	Iterações até convergir
DMO 4	0.0261831643937	21
DMO 5	0.0147606978855	22

Tabela 7: Solução da função objectivo usando o esquema 4 e 5 do DMO

$p = 4$ $x_0 = [0.5\ 0.5\ 0.5\ 0.5\ 0.5\ 0.5]$	$w1$	$w2$	$w3$	$w4$	$w5$	$w6$
DMO 4	3.32176e-7	0.25000	0.25000	3.32176e-7	0.25000	0.25000
DMO 5	9.37710e-9	0.50000	0.50000	9.37710e-9	0.50000	0.50000

Tabela 8: Solução das funções de peso usando o esquema 4 e 5 do DMO

De forma a contornar este problema optou-se, tal como na secção 4.1.1, por realizar a análise para os valores iniciais $x_0 = [0.7\ 0.6\ 0.4\ 0.7\ 0.6\ 0.4]$, tendo-se alcançado os seguintes resultados.

$p = 4$ $x_0 = [0.7\ 0.6\ 0.4\ 0.7\ 0.6\ 0.4]$	$C\ [J\ ]$	Iterações até convergir
DMO 4	0.0182767566535	35
DMO 5	0.0147606985908	36

Tabela 9: Solução da função objectivo usando o esquema 4 e 5 do DMO

$p = 4$ $x_0 = [0.7\ 0.6\ 0.4\ 0.7\ 0.6\ 0.4]$	$w1$	$w2$	$w3$	$w4$	$w5$	$w6$
DMO 4	6.52548e-5	**0.95711**	7.20908e-5	6.52548e-5	**0.95711**	7.20908e-5
DMO 5	1.20433e-7	0.50014	0.49986	1.20433e-7	0.50014	0.49986

Tabela 10: Solução das funções de peso usando o esquema 4 e 5 do DMO

Observa-se que, utilizando o esquema 5 do DMO, o programa continua sem conseguir escolher apenas um dos candidatos, continuando a preferir apresentar soluções intermédias e irreais, tendo em conta o problema. Como foi referido na secção 2.1.2.2 o esquema 5 do DMO garante que o somatorio das funções de peso é igual a 1. Este facto leva a que o programa não consiga optar por uma de duas soluções idênticas. Já o esquema 4, não tendo esta limitação, apresenta maior "liberdade" na escolha do valor das variáveis utilizadas em cada iteração, conseguindo optar por apenas uma das soluções.

No caso de se optar por utilizar o esquema 4 do DMO verifica-se que o programa já é capaz de escolher apenas um candidato. O programa opta pelo candidato w_2, para o lado esquerdo da placa, e o w_5, para o lado direito. Tal como seria pretendido, a solução obtida para as funções de peso em w_1, w_3, w_4 e w_6 é praticamente zero, sendo 0.957 para w_2 e w_5. O erro obtido, relativamente à solução calculada directamente do ANSYS, é de 4.3%.

Com isto, é possível afirmar que o resultado obtido vai de encontro ao previsto, apesar da função de peso ainda não ter atingido o valor 1.

4.1.2.2 Escolha do factor de penalização *p*

Com base nos resultados obtidos nas análises anteriores e de modo a garantir soluções realista, optou-se por utilizar o esquema 4 do DMO e $x_0 = [0.7\ 0.6\ 0.4]$ como valores iniciais para as variáveis de projecto. De referir que a análise foi iniciada com factores de penalização mais baixos, $p = 2$, e foi sendo aumentado sucessivamente. Os resultados obtidos estão expostos na tabela que se segue.

DMO 4 $x_0 = [0.7\ 0.6\ 0.4\ 0.7\ 0.6\ 0.4]$	$C\ [J]$	Iterações até convergir
$p = 2$	0.0234678555057	18
$p = 4$	0.0182767566535	32
$p = 6$	0.0189585027464	39
$p = 8$	0.0186547357501	32

Tabela 11: Solução da função objectivo com p=2, p=4, p=6 e p=8

DMO 4 $x_0 = [0.7\ 0.6\ 0.4\ 0.7\ 0.6\ 0.4]$	$w1$	$w2$	$w3$	$w4$	$w5$	$w6$
$p = 2$	3.37102e-3	0.71393	4.56423e-3	3.37102e-3	0.71393	4.56423e-3
$p = 4$	6.52548e-5	**0.957111**	7.20908e-5	6.52548e-5	**0.957111**	7.20908e-5
$p = 6$	2.62447e-4	**0.92021**	1.89564e-5	2.62447e-4	**0.92021**	1.89564e-5
$p = 8$	6.56693e-4	**0.94809**	1.65628e-5	6.56693e-4	**0.94809**	1.65628e-5

Tabela 12: Solução das funções de peso com p=2, p=4, p=6 e p=8

No caso de termos $p = 2$, verifica-se que o programa não consegue obter uma convergência total da solução. A função objectivo apresenta um erro de 33.9% em relação ao valor previsto e, apesar de indicar os candidatos 2 e 5 como solução óptima, as respectivas funções de peso têm o valor de 0.71, ficando ainda muito afastado da solução prevista.

Para $p = 4, 6, 8$ o programa já apresenta soluções mais próximas do previsto, sendo $p = 4$ a melhor das três soluções, isto é, aquela que apresenta menor erro na função objectivo e valores das funções de peso, $w2$ e $w5$, mais próximos da unidade.

4.1.3 Modelo com 12 variáveis de projecto

Após as análises realizadas com três e seis variáveis de projecto passou-se ao estudo do modelo composto por doze variáveis de projecto. Nesta análise pretende-se que o programa escolha dois dos quatros lados do corte para reforçar. Assim, pretende-se que escolha um lado, entre os lados 1 e 2, e outro entre as opções 3 e 4.

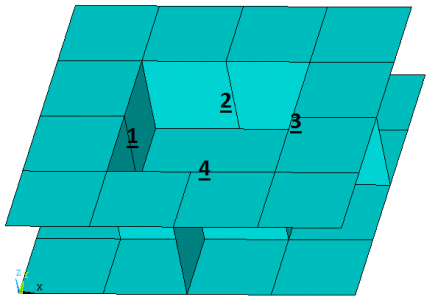


Figura 20: Numeração dos 4 lados de reforço

Cada um dos lados tem, como nas análises anteriores, três candidatos possíveis. Assim, tem-se doze variáveis, independentes seis a seis, isto é,

$$\sum_{i=1}^{6} w_i = 1 \quad ; \quad \sum_{i=7}^{12} w_i = 1 \tag{4.1}$$

A tabela seguinte ilustra cada um dos candidatos e respectivas funções de peso.

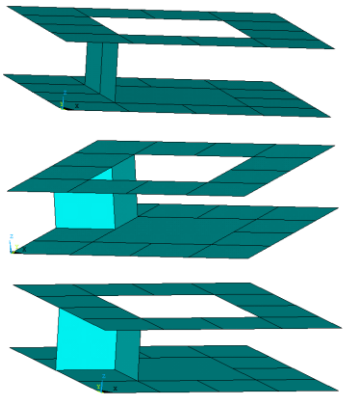
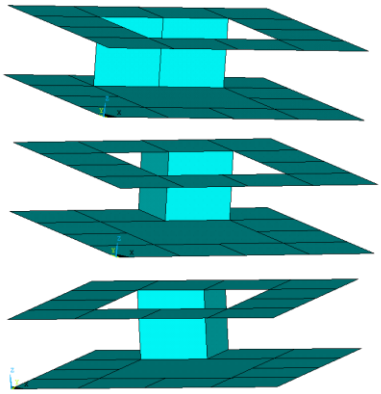
Lado 1		Lado 2	
	w_1 w_2 w_3		w_4 w_5 w_6

Tabela 13: Candidatos possíveis com 12 variáveis de projecto, para os lados 1 e 2

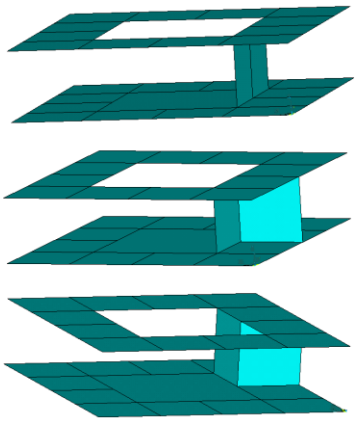
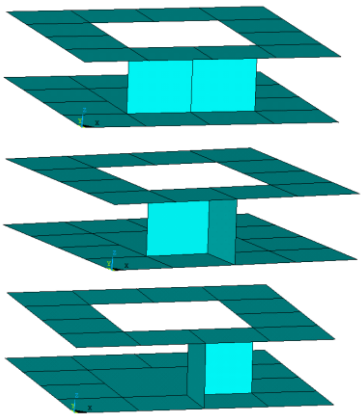
Lado 3		Lado 4	
	w_7 w_8 w_9		w_{10} w_{11} w_{12}

Tabela 14: Candidatos possíveis com 12 variáveis de projecto, para os lados 3 e 4

As condições de fronteira e o número de elementos foram mantidos iguais. Devido ao elevado número de combinações possíveis, apenas serão apresentados alguns resultados obtidos directamente do software ANSYS. Na tabela seguinte podemos observar esses resultados.

$w5 = 1$ e $w11 = 1$ // $w6 = 1$ e $w12 = 1$	***C = 1.12160545799150 J***
$w4 = 1$ e $w10 = 1$	***C = 0.33622288677114 J***
$w2 = 1$ e $w8 = 1$ // $w3 = 1$ e $w9 = 1$	***C = 0.017522491731224 J***
$w2 = 1$ e $w9 = 1$ // $w3 = 1$ e $w8 = 1$	***C = 0.01783825647957 J***
$w2 = 1$ e $w11 = 1$	***C = 1.54708366899886 J***
$w1 = 1$ e $w7 = 1$	***C = 0.34439493157110 J***

Tabela 15: Solução obtida directamente do ANSYS

Devido à simetria verifica-se, novamente, que vários candidatos têm valor óptimo idêntico. Constata-se ainda que, a solução óptima, quando se utilizam doze variáveis de projecto, é o caso em que temos $w_2 = 1$, $w_8 = 1$ e as restantes funções de peso iguais a 0 (ou por simetria $w_3 = 1$, $w_9 = 1$). Esta é a solução que se espera atingir com o programa desenvolvido.

4.1.3.1 Escolha do esquema de DMO e factor p

Iniciou-se a análise deste problema com um valor inicial igual, para todas as variáveis de projecto, $x_0 = 0.5$. Neste caso, tanto a solução apresentada pelo esquema 4 como pelo esquema 5 do DMO apresentam valores intermédios para as funções de peso.

$p = 4$ $x_0 = 0.5$ para todas as variáveis	$C\ [J]$	Iterações até convergir
DMO 4	0.0034734726280	34
DMO 5	0.0022414036954	39

Tabela 16: Solução da função objectivo usando o esquema 4 e 5 do DMO

Utilizando o esquema 4 do DMO, o programa apresenta os candidatos w_1, w_5, w_6, w_7, w_{11}, w_{12} com funções de peso muito próximas de zero e os restantes candidatos com os valores,

$$w_4 = 0.41069 = w_{10} \tag{4.2}$$

$$w_2 = w_3 = 0.05747 = w_8 = w_9 \tag{4.3}$$

É evidente que a solução obtida é bastante diferente da esperada e completamente irrealista. Foram, também, realizadas análises variando o factor de penalização, utilizando

diferentes valores iniciais para as variáveis de projecto, entre outras. No entanto, a solução final obtida foi sempre do mesmo género.

Resumindo, com o aumento do número de variáveis de projecto, verificou-se que o programa apresenta cada vez mais dificuldade em obter a solução óptima global. As soluções obtidas nestas análises evidenciam a dificuldade que o optimizador tem em sair dos mínimos locais encontrados e que, na maioria dos casos, têm soluções inferiores ao valor óptimo previsto, mas que são soluções irrealistas, pois as funções de pesos assumem valores longe dos extremos 1 e 0.

O esquema 5 do DMO evidencia incapacidade em optar por dois candidatos com soluções idênticas, preferindo indicar valores intermédios como solução. O Esquema 4, apesar de apresentar resultados mais realistas que o esquema 5, denota alguma dificuldade em aproximar as funções de peso dos seus valores extremos.

De forma a solucionar o problema dos valores intermédios, decidiu-se utilizar uma função penalizadora.

4.2 Caso 2 - Método DMO + Método penalização explícito

A utilização de um método de penalização explícito permite obter uma penalização adicional quando o programa opta por escolher valores intermédios para as funções de peso. Por outras palavras, esta é uma forma de acentuar a necessidade do programa desenvolvido chegar a valores óptimos com uma das funções de peso igual a 1 e as restantes iguais a 0. Nas análises realizadas nesta secção da tese é utilizada a função penalizadora (2.14) apresentada na secção 2.1.2.3 .

4.2.1 Modelo com 3 variáveis de projecto

No caso da utilização de três variáveis de projecto, a utilização desta penalização explícita tem como principal objecto solucionar o problema que o esquema 5 do DMO apresenta. Assim, espera-se que este método permita ao programa escolher um dos candidatos dois e três como solução óptima e deixar de apresentar uma solução intermédia.

De seguida apresentam-se os resultados obtidos em diversas análises onde se utilizou o método de penalização explícito como forma de penalização dos valores intermédios.

$p=2 \quad k=0.1$ $q=1$	$C\ [J]$	Iterações até convergir
$DMO\ 5-x_0=[0.5\ 0.5\ 0.5]$	0.0147606984632	12
$DMO\ 5-x_0=[0.7\ 0.7\ 0.7]$	0.0175249208098	8
$DMO\ 4-x_0=[0.7\ 0.7\ 0.7]$	0.0338825573708	37

Tabela 17: Solução da função objectivo usando o esquema 4 e 5 do DMO

$p=2 \quad k=0.1$ $q=1$	$w1$	$w2$	$w3$
$DMO\ 5-x_0=[0.5\ 0.5\ 0.5]$	0.0000	0.50000	0.50000
$DMO\ 5-x_0=[0.7\ 0.7\ 0.7]$	0.0000	0.0000	**1.00000**
$DMO\ 4-x_0=[0.7\ 0.7\ 0.7]$	3.55376e-4	0.17690	0.17690

Tabela 18: Solução das funções de peso usando o esquema 4 e 5 do DMO

Como é possivel verificar nas tabelas 17 e 18 quando se assume um valor inicial das variáveis de projecto igual a 0.5 o programa continua sem conseguir optar pelos candidatos 2 e 3, preferindo apresentar uma solução "cinzenta". Na realidade esta solução até apresenta função objectivo menor que no caso óptimo, demonstrando o porquê do programa preferir escolher uma solução intermédia. No entanto, este problema deixa de existir quando se utiliza 0.7 para valor inicial. Iniciar o processo iterativo com diferentes valores para as variáveis de projecto irá fazer com que o "caminho" percorrido ao longo da análise seja diferente. Assim, quando se começa com 0.5 nos três candidatos o processo iterativo segue um "caminho" onde poderá existir um mínimo local, acabando por convergir numa solução "cinzenta". Iniciando o processo em 0.7 o processo seguirá outro percurso, conseguindo convergir na solução pretendida ($w_3 = 1$; $w_1 = w_2 = 0$). O erro obtido em relação à solução tirada directamente do ANSYS é da ordem de 0.006%, o que representa a qualidade da solução obtida.

Quanto à escolha do esquema DMO a utilizar, verifica-se que o esquema 5 é a melhor opção já que o 4, utilizando a penalização de valores intermédios, não garante que o somatório das funções de peso é 1 e, por isso, não permite atingir a solução óptima pretendida. Neste caso, seria necessário a utilização de outra função penalizadora que obrigasse a que essa condição fosse satisfeita.

4.2.2 Modelo com 6 variáveis de projecto

A análise realizada com seis variáveis de projecto evidenciou, mais uma vez, o problema identificado na secção 4.2.1.. Assumindo 0.5 como valor inicial das variáveis de projecto, o programa continua sem conseguir optar pelas soluções 2/3 e 5/6, que apresentam função objectivo idênticas devido à simetria da placa e das condições de fronteira aplicadas.

Utilizou-se por isso 0.7 como valor inicial das variáveis de projecto. Neste caso observou-se que o programa já consegue apresentar resultados coerentes e de acordo com o previsto.

$p = 2 \quad k = 10$ $q = 1 \quad DMO\ 5$	$C\ [J]$	Iterações até convergir
$x_0 = [0.7\ 0.7\ 0.7\ 0.7\ 0.7\ 0.7]$	0.09368349487977	65

Tabela 19: Solução da função objectivo usando o esquema 5 do DMO

$p = 2 \quad k = 10$ $q = 1 \quad DMO\ 5$	$w1$	$w2$	$w3$	$w4$	$w5$	$w6$
$x_0 = [0.7\ 0.7\ 0.7$ $0.7\ 0.7\ 0.7]$	**0.99904**	5.9131e-4	3.7349e-4	**0.99904**	3.7349e-4	5.9131e-4

Tabela 20: Solução das funções de peso usando o esquema 5 do DMO

Tal como no caso de três variáveis de projecto, a utilização da função penalizadora permite solucionar o problema de se obter solução óptima com valores das funções de peso intermédios. No entanto, o problema pressiste se se utilizarem valores próximos de 0.5 como valor inicial das variáveis de projecto. Para além disso, a solução que o programa obtem não é a mínima, se compararmos com a que se obtem directamente do ansys. Factores como a existência de mínimos locais ou mesmo a malha utilizada, que neste caso é bastante grosseira, podem influenciar a solução final.

4.2.3 Modelo com 12 variáveis de projecto

No caso de se utilizar doze variáveis de projecto o programa parece convergir para uma solução que não é a óptima. Como é possível verificar na tabela 15, a solução óptima seria para os casos $w_2 = 1$, $w_8 = 1$ ou $w_3 = 1$, $w_9 = 1$. Tal como na análise realizada em 4.1.3.1, o programa tende a apresentar o caso $w_4 = 1$, $w_{10} = 1$ como melhor solução,

$p = 4$ k=0.1 $x_0 = 0.5$ para todas as variáveis	$C\ [J]$	Iterações até convergir
DMO 5	0.00544335291988	42

Tabela 21: Solução da função objectivo usando o esquema 5 do DMO

O programa apresenta os candidatos w_1, w_5, w_6, w_7, w_{11}, w_{12} com funções de peso muito próximas de zero e os restantes candidatos com os valores,

$$w_4 = 0.95387 = w_{10} \tag{4.4}$$

$$w_2 = w_3 = 0.02301 = w_8 = w_9 \tag{4.5}$$

Neste caso, obtém-se uma solução que é de facto menor que o valor esperado (ver tabela 15). No entanto, esperava-se que à solução obtida correspondessem à escolha dos candidatos w_2 e w_8 ou w_3 e w_9, o que de facto não acontece. Tendo-se obtido um valor da função objectivo inferior ao previsto, verifica-se que o programa opta por escolher este mínimo local, em detrimento da solução prevista, porque este denota ser melhor opção em termos de aumento de rigidez da estrutura. Tentou-se ainda alterar o valor da penalização, o que não se traduziu numa melhoria dos resultados.

Assim, conclui-se que, nas condições idêntificadas na secção 4.2.2, a utilização de uma função penalizadora de valores intermédios permite solucionar o problema apresentado devido à simetria da placa.

4.3 Caso 3 - Utilização de duas funções penalizadores

De forma a verificar se os problemas que o programa exibia se deviam aos esquemas de DMO utilizados, optou-se por implementar uma abordagem mais simples que os esquemas 4 e 5 do método do DMO.

Considere-se que cada função de peso w_i depende apenas de uma variável de projecto, isto é,

$$w_i = x_i{}^p\,, \qquad p \geq 1\,,\ 0 \leq x_i \leq 1 \tag{4.6}$$

Neste caso, idêntico ao esquema 1 do método DMO, serão consideradas duas funções penalizadoras, a primeira para penalizar os valores intermédios (soluções “cinzentas”) e a segunda para garantir que o somatório das funções de peso seja sempre 1. As funções são representadas por,

$$A = k * \sum_{i=1}^{N^c} w_i{}^q (1 - w_i)^q\,, \qquad q \geq 0\,,\ k \geq 0\,,\ 0 \leq w_i \leq 1 \tag{4.7}$$

$$B = \frac{c}{2} * (\sum_{i=1}^{N^c} w_i - 1)^2\,, \qquad c \geq 0\,,\ 0 \leq w_i \leq 1 \tag{4.8}$$

Assim, à função objectivo será somada a contribuição de cada uma desta funções penalizadoras,

$$\hat{C} = C + A + B\,, \qquad A \geq 0\,,\ B \geq 0 \tag{4.9}$$

Neste caso, os estudos realizados evidenciaram resultados pouco coerentes em relação ao esperado. Começou-se com o valor inicial de 0.5 para todas a variáveis de projecto e verificou-se que o programa ficava muito tempo no line search, acabando por ir sempre parar aos valores iniciais. Testou-se, de seguida, para valores iniciais das variáveis de projecto iguais 0.7 e identificou-se exactamente o mesmo problema. Para outros valores iniciais testados o programa evidenciou o mesmo problema, pelo que se conclui que esta alternativa não é a indicada para solucionar o problema.

4.4 Caso 4 - FMINCON

Todos os casos de estudo já apresentados foram realizados com recurso ao optimizador FAIPA. Devido à grande quantidade de problemas que foram sempre aparecendo nos casos anteriores, decidiu-se alterar o optimizador utilizado de forma a verificar se as dificuldades obtidas se deviam ao próprio optimizador ou ao programa em si.

Optou-se por utilizar o FMINCON, optimizador da tollbox do MATLAB, e o programa foi adaptado para que a optimização fosse realizada por este optimizador. Tal como acontece para o FAIPA, as condições de paragem do optimizador irão variar consoante a solução que se obtem.

4.4.1 Esquema 5 do DMO + Método penalização explícito

Nos casos de estudo já realizados observou-se que a melhor solução era obtida quando se utilizava o método DMO em conjunto com um método de penalização de valores intermédios. Desta forma, o primeiro caso que se irá estudar, com recurso ao optimizador FMINCON, será exactamente este. As condições de fronteira e cargas aplicadas serão idênticas às aplicadas anteriormente. Os resultados expostos nas tabelas seguintes são aqueles que consideraram mais importantes nas muitas análises realizadas.

Começou-se pelo caso de três variáveis de projecto. Na tabela seguinte é possível observar a solução obtida e compará-la com a solução do optimizador FAIPA, exposto no sub-capítulo 4.2 .

$x_0 = [0.5\ 0.5\ 0.5]$ $q = 1 \quad p = 2 \quad DMO\ 5$	$C\ [J]$	Iterações até convergir	Tempo de análise
FAIPA / $k = 0.1$	0.0147606984633	10	89.4seg
FAIPA / $k = 0.1$ / $x_0 = [0.7\ 0.7\ 0.7]$	0.0175249207579	8	94.7 seg
FMINCON / $k = 0.01$	0.0175491731224	64	176.4 seg
ANSYS	0.0175239002192	-	-

Tabela 22: Solução da função objectivo usando o FAIPA e o FMINCON

$x_0 = [0.5\ 0.5\ 0.5]$ $q = 1 \quad p = 2 \quad DMO\ 5$	$w1$	$w2$	$w3$
FAIPA / $k = 0.1$	1.2363e-8	0.5000	0.5000
FAIPA / $k = 0.1$/ $x_0 = [0.7\ 0.7\ 0.7]$	1.28073e-16	2.79105e-9	**1.0000**
FMINCON / $k = 0.01$	2.9606e-15	2.9606e-15	**1.0000**

Tabela 23: Solução das funções de peso usando o FAIPA e o FMINCON

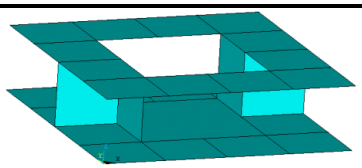
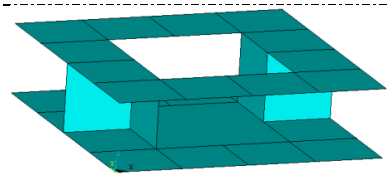
	Soluções
FAIPA / $p = 2$ / $k = 0.1$ $x_0 = [0.7 \ldots 0.7]$ DMO5 $w3 = 1$	
FMINCON /$p = 2$ / $k = 0.01$ $x_0 = [0.5 \ldots 0.5]$ DMO5 $w3 = 1$	

Tabela 24: Solução obtida usando o FAIPA e o FMINCON

Os resultados obtidos evidenciam uma melhoria relativamente ao caso em que se utiliza o FAIPA. Utilizando o optimizador da toolbox do MATLAB consegue-se, com um valor menor na constante da expressão da penalização, solucionar o problema das soluções intermédias. Enquanto que utilizando o FAIPA se teve que alterar o valor inicial das variáveis de projecto, pois mesmo aumentando a penalização o programa não conseguiu sair do mínimo local encontrado, obtendo uma solução irrealista para $x_0 = [0.5\ 0.5\ 0.5]$, com o FMINCON atinge-se a solução óptima com esses mesmos valores. De referir que o tempo de cálculo no FMINCON é superior aos casos em que se recorre ao optimizador FAIPA.

Depois da análise realizada com três variáveis de projecto pretende-se provar que o programa funciona com mais variáveis de projecto. Assim, analisou-se o caso de seis variáveis e comparou-se com a solução obtida através do FAIPA. Os resultados obtidos foram os seguintes.

$q = 1 \quad DMO\ 5$	$C\ [J]$	Iterações até convergir	Tempo de análise
FAIPA / $p = 2$ / $k = 10$ $x_0 = [0.7\ 0.7\ 0.7\ 0.7\ 0.7\ 0.7]$	0.09368349487977	65	300.9 seg
FMINCON / $p = 2$ / $k = 10$ $x_0 = [0.5\ 0.5\ 0.5\ 0.5\ 0.5\ 0.5]$	0.1323060416045	161	529.3 seg

Tabela 25: Solução da função objectivo usando o FAIPA e o FMINCON

$q=1 \quad DMO\ 5$	$w1$	$w2$	$w3$	$w4$	$w5$	$w6$
FAIPA / $p=2$ / $k=10$ $x_0=[0.7\ldots\ 0.7]$	**0.99904**	5.9131e-4	3.7349e-4	**0.99904**	3.7349e-4	5.9131e-4
FMINCON /$p=2$ / $k=10$ $x_0=[0.5\ldots 0.5]$	**0.99891**	5.2748e-4	5.6119e-4	**0.99891**	5.6137e-4	5.2763e-4

Tabela 26: Solução das funções de peso usando o FAIPA e o FMINCON

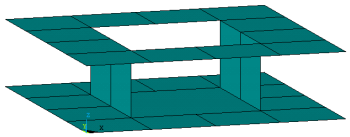
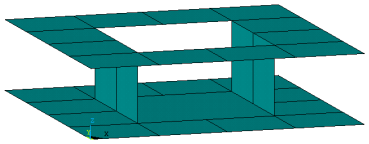
	Soluções
FAIPA / $p=2$ / $k=10$ $x_0=[0.7\ldots\ 0.7]$ DMO5 $w1=1=w4$	
FMINCON /$p=2$ / $k=10$ $x_0=[0.5\ldots 0.5]$ DMO5 $w1=1=w4$	

Tabela 27: Solução obtida usando o FAIPA e o FMINCON

Mais uma vez, é evidente que utilizando o FMINCON se consegue obter a solução óptima pretendida com os valores $x_0=[0.5\ldots\ 0.5]$, o que no caso do FAIPA não foi possível. Mais uma vez, o programa opta por escolher os candidatos w_1 e w_4 como solução. Esta não era a solução prevista, tendo em conta os resultados obtidos directamente do ANSYS, no entanto, estes resultados podem dever-se à utilização de uma malha grosseira, com apenas um elemento por área. Este aspecto, aliado à existência de mínimos locais próximos dos valores extremos ($w_i=1$ e restantes $w_i=0$), faz com que as soluções sofram grandes variações em zonas próximas desses pontos.

Obtidas as soluções nas análises com três e seis variáveis, prosseguiu-se para o estudo do problema com doze variáveis de projecto. Neste caso realizaram-se algumas análises de modo a verificar se com a variação de certos parâmetros, a solução se aproximava do valor esperado. De seguida são apresentados os resultados obtidos no caso de se variar o expoente, p, utilizado no esquema 5 do método DMO.

$x_0=[0.5\ldots\ 0.5]$ $DMO\ 5 \quad k=10 \quad q=1$	$C\ [J]$	Iterações até convergir	Tempo de análise
$p=2$	0.0626190714661	195	1899.6 seg
$p=4$	0.0701945886194	208	2039.1 seg
$p=6$	0.0829095886420	221	2135.1 seg
$p=8$	0.0651790623693	260	2519.2 seg
$p=12$	0.0910617783432	221	2098.9 seg

Tabela 28: Solução da função objectivo usando p=2, p=4, p=6, p=8 e p=12

$x_0 = [0.5 \dots 0.5]$ $DMO\ 5$ $k = 10 \quad q = 1$	$w1$	$w2$	$w3$	$w4$	$w5$	$w6$
$p = 2$	**0.99487**	1.8789e-3	2.3722e-3	1.1619e-4	3.7929e-4	3.7929e-4
$p = 4$	**0.99653**	1.0245e-3	1.5667e-2	5.5747e-5	4.1173e-4	4.1171e-4
$p = 6$	**0.99876**	4.6835e-4	7.5332e-4	3.3160e-7	9.8642e-6	9.8636e-6
$p = 8$	**0.99785**	7.1141e-4	1.1517e-3	2.9399e-6	1.4189e-4	1.4188e-4
$p = 12$	**0.99896**	2.4114e-4	7.7436e-4	3.1635e-8	1.2124e-5	1.2122e-5

Tabela 29: Solução das funções de peso usando p=2, p=4, p=6, p=8 e p=12

$x_0 = [0.5 \dots 0.5]$ $DMO\ 5$ $k = 10 \quad q = 1$	$w7$	$w8$	$w9$	$w10$	$w11$	$w12$
$p = 2$	**0.99844**	5.6045e-4	4.2990e-4	8.1155e-5	2.4291e-4	2.4291e-4
$p = 4$	**0.99874**	5.6809e-4	3.7147e-4	2.0207e-5	1.4952e-4	1.4953e-4
$p = 6$	**0.99864**	7.7501e-4	4.3643e-4	3.4129e-6	7.2026e-5	7.2029e-5
$p = 8$	**0.99785**	1.1458e-3	7.1231-4	2.9772e-6	1.4338e-4	1.4338e-4
$p = 12$	**0.99896**	7.7281e-4	2.4054e-4	3.1580-8	1.2177e-5	1.2178e-5

Tabela 30: Solução das funções de peso usando p=2, p=4, p=6, p=8 e p=12

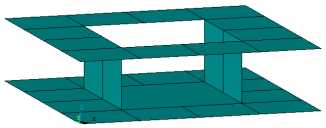
	Soluções
$DMO\ 5 \quad k = 10 \quad q = 1$ $p = 2, 4, 6, 8, 12$ $w1 = 1 = w7$	

Tabela 31: Solução obtida usando p=2, p=4, p=6, p=8 e p=12

Observa-se que não existe uma melhoria significativa que justifique a utilização de um factor p superior a 2. O caso em que se usa $p = 2$ permite obter uma solução com funções de peso muito próximas dos extremos, com um menor tempo de cálculo e número de iterações.

De seguida pretende-se verificar a influência da função penalizadora na solução obtida. Assim, irá variar-se o parâmetro k, que multiplica por essa função.

$x_0 = [0.5 \dots 0.5]$ $DMO\ 5 \quad p = 2 \quad q = 1$	$C\ [J]$	Iterações até convergir	Tempo de análise
$k = 0.01$	0.0026962669611	208	1959.1 seg
$k = 0.1$	0.0026962669611	208	1965.3 seg
$k = 1$	0.0406926163437	208	1956.0 seg
$k = 10$	0.0626190714661	195	1899.6 seg
$k = 100$	0.1825703497452	208	1964.6 seg

Tabela 32: Solução da função objectivo usando $k = 0.01$, $k = 0.1$, $k = 1$, $k = 10$ e $k = 100$

$x_0 = [0.5 \ldots 0.5]$ $DMO\ 5 \quad p = 2$ $q = 1$	$w1$	$w2$	$w3$	$w4$	$w5$	$w6$
$k = 0.01$	0.35951	0.14154	0.15288	0.09942	0.12332	0.12332
$k = 0.1$	0.35951	0.14154	0.15288	0.09942	0.12332	0.12332
$k = 1$	**0.99546**	1.6939e-3	2.2195e-3	8.3151e-5	2.7284e-4	2.7281e-4
$k = 10$	**0.99487**	1.8789e-3	2.3722e-3	1.1619e-4	3.7929e-4	3.7929e-4
$k = 100$	**0.99959**	1.1214e-4	1.4636e-4	2.1850e-5	6.5178e-5	6.5177e-5

Tabela 33: Solução das funções de peso usando $k = 0.01$, $k = 0.1$, $k = 1$, $k = 10$ e $k = 100$

$x_0 = [0.5 \ldots 0.5]$ $DMO\ 5 \quad p = 2$ $q = 1$	$w7$	$w8$	$w9$	$w10$	$w11$	$w12$
$k = 0.01$	0.35948	0.15288	0.14254	0.09942	0.12335	0.12335
$k = 0.1$	0.35948	0.15288	0.14254	0.09942	0.12335	0.12335
$k = 1$	**0.99403**	2.1904e-3	1.6866e-3	3.3713e-4	8.8037e-4	8.8037e-4
$k = 10$	**0.99844**	5.6045e-4	4.2990e-4	8.1155e-5	2.4291e-4	2.4291e-4
$k = 100$	**0.99976**	8.6869e-5	6.6547e-5	1.3011e-5	3.8844e-5	3.8844e-5

Tabela 34: Solução das funções de peso usando $k = 0.01$, $k = 0.1$, $k = 1$, $k = 10$ e $k = 100$

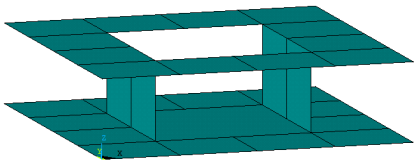
	Soluções
$DMO\ 5 \quad q = 1 \quad p = 2$ $k = 0.01\,, 0.1\,, 1\,, 10\,, 100$ $w1 = 1 = w7$	

Tabela 35: Solução obtida usando $k = 0.01$, $k = 0.1$, $k = 1$, $k = 10$ e $k = 100$

Pelos resultados exposto nas tabelas 32, 33 e 34 podemos verificar que, um aumento no valor da função penalizadora permite aproximar as funções de peso dos seus valores extremos. Sendo a penalização dos valores intermédios o principal objectivo da função penalizadora, podemos dizer que esta cumpre o seu papel. No entanto, tendo em conta os valores obtidos directamente do software ANSYS, seria de esperar que o programa optasse pelos candidatos w_2 / w_8 ou w_3 / w_9 e não por w_1 e w_7 , até porque o valor da função objectivo obtida neste caso é superior ao caso em que se tem $w_3 = 1$ e $w_9 = 1$.

A razão para este facto já foi explicada atrás. Mais uma vez, o optimizador encontra um mínimo local, que não é exactamente um valor extremo mas é próximo, e mesmo com o aumento do valor da função penalizadora tem dificuldade em optar por outra solução. Assim, acaba por escolher os candidatos w_1 e w_7 . A existência destes mínimos locais fica evidente se observarmos as tabelas 32, 33 e 34 onde se verifica uma função objectivo significativamente

inferior nos casos em que se obtêm soluções intermédias ou "cinzentas" (com k=*0.01* e $k = 0.1$).

Tendo em conta que os resultados obtidos para três e seis variáveis de projecto estão de acordo com o esperado, decidiu-se testar três casos mais complexos, de forma a observar se o programa conseguia obter a solução óptima. Estes casos serão estudados com recurso a seis variáveis de projecto. De seguida é apresentada a lista das três variantes do problema a estudar.

- Utilização de 4 elementos por área gerada
- Utilização de 9 furos idênticos
- Alteração das condições de fronteira – $x = 0$ e $y = 4$

O primeiro caso consiste em refinar a malha utilizada, passando a ter quatro elementos por área, em vez de um elemento. Com esta análise pretende-se mostrar que o programa funciona para malhas com mais que um elementos, de modo a que seja possível obter resultados completamente convergidos com recurso a malhas mais complexas. Como o número de candidatos é pequeno, é possível confirmar se os resultados obtidos estão de acordo com o previsto através da tabela seguinte, onde estão expostas as soluções obtidas directamente do ANSYS para os casos extremos onde temos uma função de peso igual a 1 e as restantes iguais a 0.

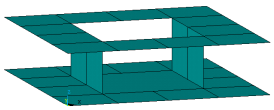
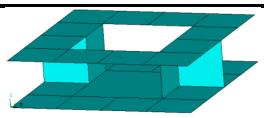
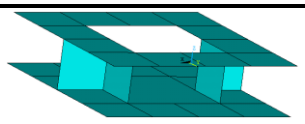
 $w1 = 1$ e $w4 = 1$	***C = 1.42712507735993 J***
 $w1 = 1$ e $w5 = 1$ // $w1 = 1$ e $w6 = 1$ $w2 = 1$ e $w4 = 1$ // $w3 = 1$ e $w4 = 1$	***C = 1.56634908795839 J***
 $w2 = 1$ e $w5 = 1$ // $w3 = 1$ e $w6 = 1$	***C = 1.62963453740768 J***
 $w2 = 1$ e $w6 = 1$ // $w3 = 1$ e $w5 = 1$	***C = 1.73830806719036 J***

Tabela 36: Solução obtida directamente do ANSYS com uma malha de 4 elementos por área

Passemos então às soluções obtidas pelo programa. Neste caso, chegou-se os seguintes resultados:

$p = 2 \quad DMO\ 5$ $x_0 = [0.5 \dots 0.5]$	$C\ [J]$	Iterações até convergir	Tempo de análise
$q = 1\ /\ k = 0.1$	1.2458607792887	133	454.9 seg
ANSYS	1.4271250773599	-	-

Tabela 37: Solução da função objectivo usando uma malha de 4 elementos

$p = 2 \quad DMO\ 5$ $x_0 = [0.5 \dots 0.5]$	$w1$	$w2$	$w3$	$w4$	$w5$	$w6$
$q = 1\ /\ k = 0.1$	**0.99123**	1.1345e-12	8.7691e-3	**0.99117**	8.8281e-3	4.1034e-10

Tabela 38: Solução das funções de peso usando uma malha de 4 elementos

O programa apresenta os candidatos w_4 e w_1 como solução óptima. Este resultado está de acordo com o esperado uma vez que, através da tabela 36 verifica-se que esta é a melhor solução. De referir que, com o aumento do número de elementos, isto é, utilizando uma malha mais refinada, as soluções óptimas sofrem grandes alterações. Neste caso os candidatos w_4 e w_1 passam a ser a escolha mais indicada, o que não acontecia com uma malha de um elemento por área. Este facto explica os resultados obtidos em algumas análises anteriores, como para seis variáveis de projecto (tabela 27) ou doze variáveis de projecto (sub-capítulo 4.2.3), onde as funções de peso assumiam valores próximos de 1 para candidatos que, segundo as soluções obtidas directamente do ANSYS, não seriam os ideais.

O segundo estudo consiste em aumentar o número de furos na chapa superior do modelo, utilizando as mesmas condições de fronteira. Assim, pretende-se verificar que o software desenvolvido consegue obter soluções reais para casos com vários buracos. Os resultados apresentados nas tabelas seguintes foram obtidos para o caso de uma chapa com nove furos idênticos, isto é, todos com o mesmo tipo de configuração de reforço.

$p = 2 \quad DMO\ 5$ $x_0 = [0.5 \dots 0.5]$	$C\ [J]$	Iterações até convergir	Tempo de análise
$q = 1\ /\ k = 50$	48.651831168447	112	404.3 seg

Tabela 39: Solução da função objectivo usando estrutura com 9 furos idênticos

$p = 2 \quad DMO\ 5$ $x_0 = [0.5 \dots 0.5]$	$w1$	$w2$	$w3$	$w4$	$w5$	$w6$
$q = 1\ /\ k = 50$	**0.99880**	6.0396e-4	5.9952e-4	**0.99880**	5.9951e-4	6.0395e-4

Tabela 40: Solução das funções de peso usando estrutura com 9 furos idênticos

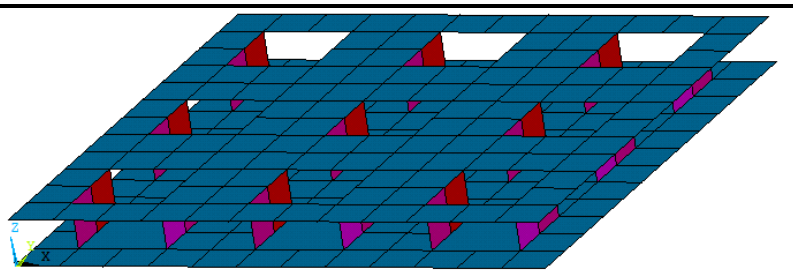
Soluções	
$p = 2 \quad DMO\ 5$ $w1 = 1 = w4$	

Tabela 41: Solução obtida usando estrutura com 9 furos idênticos

Como é possível observar, o programa apresenta os candidatos w_4 e w_1 como solução, isto é, coloca todos os reforços na mesmo direcção, perpendiculares ao lados encastrados da placa.

O último estudo realizado neste sub-capítulo consiste em aplicar uma alteração às condições de fronteira da placa. Passamos então a ter a placa encastrada em $x = 0$ e $y = 4$ em vez de $y = 0$ e $y = 4$. Podemos observar as novas condições de fronteira na imagem seguinte.

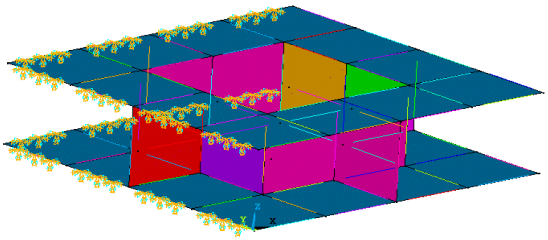


Figura 21: Estrutura com as placas encastradas em x=0 e y=4

Procedeu-se então à análise deste caso, tendo-se obtido os seguintes resultados.

$p = 2 \quad DMO\ 5$ $x_0 = [0.5 \dots 0.5]$	$C\ [J]$	Iterações até convergir	Tempo de análise
$q = 1\ /\ k = 0.1$	0.5800650258787	112	366.1 seg
$q = 1\ /\ k = 1$	0.5130776253298	112	364.9 seg
$q = 1\ /\ k = 50$	0.1470445712371	119	376.1 seg

Tabela 42: Solução da função objectivo usando estrutura encastrada em x=0 e y=4

$p = 2 \quad DMO\ 5$ $x_0 = [0.5 \dots 0.5]$	$w1$	$w2$	$w3$	$w4$	$w5$	$w6$
$q = 1\ /\ k = 0.1$	0.48736	2.3713e-2	0.48893	**0.92012**	1.0382e-5	7.9873e-2
$q = 1\ /\ k = 1$	0.48232	3.5071e-2	0.48261	**0.99723**	3.4308e-7	2.7682e-3
$q = 1\ /\ k = 50$	**0.99965**	1.6646e-4	1.8421e-4	**0.99965**	1.8416e-4	1.6641e-4

Tabela 43: Solução das funções de peso usando estrutura encastrada em x=0 e y=4

Podemos verificar que o optimizador escolhe o candidato w_4 no lado direita, mas tem dificuldade em optar por um dos candidatos, w_1 e w_3, no lado esquerdo da placa, quando se utilizam penalizações mais baixas. Aumentando a penalização é possível obter a solução óptima pretendida. Observa-se ainda que a solução óptima apresenta uma função objectivo três a quatro vezes inferior aquela que se obteve no outros dois casos.

Em suma, foi visível que com a utilização do FMINCON em deterimento do FAIPA conseguiu-se uma melhoria na qualidade dos resultados obtidos.

4.4.2 Esquema 4 do DMO + Método Penalização Explícito

Nesta secção pretende-se verificar se o esquema 4 do DMO, utilizado em conjunto com uma penalização explícita, pode ser uma boa opção para este tipo de optimizações.

Quanto se testou a utilização do esquema 4 do DMO, sem recurso a funções penalizadoras, verificou-se que a solução óptima era obtida para valores das funções de peso muito próximos de zero e todos iguais entre si. Isto acontece porque, como já foi dito, o esquema 4 do método DMO não garante que o somatório das funções de peso seja um, o que pode conduzir a este tipo de resultados.

De forma a contornar este problema optou-se por utilizar, em conjunto com o esquema 4, as duas funções penalizadoras referidas na secção 4.3 . Desta forma, pretende-se garantir que o somatório das funções de peso seja próximo de um e que não existam soluções intermédias e irrealistas. Numa primeira análise utilizou-se apenas a função penalizadora que garante o somatório das funções de peso igual a um, tendo-se depois utilizado as duas funções penalizadoras em conjunto. Assim, obtiveram-se os seguintes resultados:

$p = 2 \quad DMO\,4 \quad q = 1$ $x_0 = [0.5 \ldots 0.5]$	$C\ [J\ ]$	Iterações até convergir	Tempo de análise
Penalização B $c = 10$	0.0570462249436	112	346.4 seg
Penalização A+B $k = 1\ /\ c = 10$	0.0621627448180	112	339.0 seg

Tabela 44: Solução da função objectivo usando esquema 4 do DMO com penalizações

$p=2\ \ DMO\ 4\ \ q=1$ $x_0=[0.5\ldots0.5]$	$w1$	$w2$	$w3$	$w4$	$w5$	$w6$
Penalização B $c=10$	**0.85092**	2.1441e-3	2.2220e-3	**0.86576**	2.3542e-3	1.1501e-3
Penalização A+B $k=1\ /\ c=10$	**0.85293**	2.0367e-3	1.5785e-3	**0.86167**	1.4000e-3	1.0075e-3

Tabela 45: Solução das funções de peso usando esquema 4 do DMO com penalizações

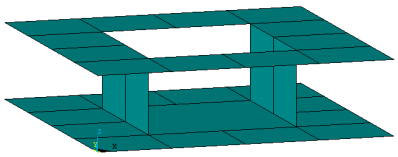
Soluções	
$p=2\ \ DMO\ 4$ Penalização B $/\ c=10$ Penalização A + B $/\ k=1\ /\ c=10$ $w1=1=w4$	

Tabela 46: Solução obtida usando esquema 4 do DMO com penalizações

Tanto no caso em que se utiliza apenas a penalização B, como no caso em que se utilizam as duas penalizações, A e B, o programa não consegue obter funções de peso iguais a um, para os candidatos escolhidos. Tentou-se ainda aumentar o valor das penalizações sem que isso provocasse uma melhoria nos resultados. Assim, apesar de se idênticar claramente a escolha dos candidatos óptimos, a solução apresenta funções de peso com valores afastados dos extremos, sendo por isso irreais.

4.4.3 Esquema 1 do DMO + duas funções penalizadores

Neste sub-capítulo são apresentadas as soluções obtidas com recurso ao esquema exposto na secção 4.3. Sendo mais simples que o esquema 4 e 5 do método DMO, espera-se que convirja mais rapidamente para uma solução óptima.

$p=1\ \ DMO\ 1$+Penalizações $x_0=[0.5\ldots0.5]$	$C\ [J]$	Iterações até convergir	Tempo de análise
6 variáveis de projecto $q=1\ /\ k=1\ /\ c=50$	0.0202840368013	112	199.5 seg
12 variáveis de projecto $q=1\ /\ k=0.01\ /\ c=0.1$	0.1316010657272	208	364.8 seg

Tabela 47: Solução da função objectivo usando esquema 1 do DMO com penalizações

DMO 1+Penalizações $x_0 = [0.5 \dots 0.5] \quad p = 1$	$w1$	$w2$	$w3$	$w4$	$w5$	$w6$
6 variáveis de projecto $q = 1 / \ k = 1 \ / c = 50$	2.1655e-2	2.8239e-4	**0.83288**	2.1776e-2	1.4854e-5	**0.83264**
12 variáveis de projecto $q = 1 / k = 0.01 / c = 0.1$	4.3050e-2	4.5594e-2	4.5124e-2	4.7294e-2	4.5535e-2	**1.0000**

Tabela 48: Solução das funções de peso usando esquema 1 do DMO com penalizações

DMO 1+Penalizações $x_0 = [0.5 \dots 0.5] \quad p = 1$	$w7$	$w8$	$w9$	$w10$	$w11$	$w12$
6 variáveis de projecto $q = 1 / \ k = 1 \ / c = 50$	-	-	-	-	-	-
12 variáveis de projecto $q = 1 / k = 0.01 / c = 0.1$	4.3043e-2	4.5180e-2	4.5530e-2	4.7294e-2	4.5535e-2	**1.0000**

Tabela 49: Solução das funções de peso usando esquema 1 do DMO com penalizações

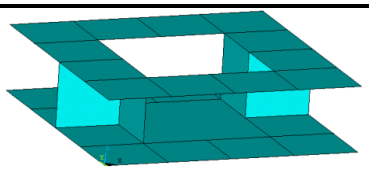
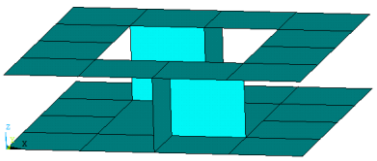
	Solução
6 variáveis de projecto $q = 1 / k = 1 / c = 50$ $w3 = 1 = w6$	
12 variáveis de projecto $q = 1 / k = 0.01 / c = 0.1$ $w6 = 1 = w12$	

Tabela 50: Solução obtida usando esquema 1 do DMO com penalizações

Pelas tabelas 48 e 49 verifica-se que, para doze variáveis de projecto, o software opta pelos candidatos w_6 e w_{12} como solução. Já para o caso em que se utilizam seis variáveis de projecto, o programa apresenta os candidatos w_3 e w_6 como solução, no entanto, com valores das funções de peso ainda um pouco afastados de um.

Devido à existência de um mínimo local, a energia é substancialmente menor no caso de 6 variáveis de projecto. No entanto, se tivermos em conta as penalizações impostas, a função objectivo atinge valores duas ordens de grandeza superiores (penalização $A = 0.32133216363$ e penalização $B = 1.05676931029$ - valores obtidos na iteração 112). Desta forma, utilizando seis variáveis de projecto não foi possível conseguir resultados completamente convergidos, mesmo utilizando maiores penalizações.

Quanto ao tempo necessário ao processo de optimização, verifica-se que é substancialmente menor, quando comparado com os casos onde se utiliza o esquema 4 e 5 do DMO.

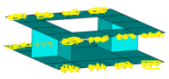
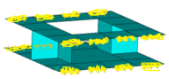
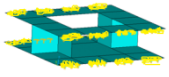
6 variáveis de projecto		
	Nº Iterações	Tempo de análise (seg)
DMO 5 + A	161	529.3
DMO 4 + A+ B	112	339.0
DMO 1 + A+ B	112	199.5
12 variáveis de projecto		
	Nº Iterações	Tempo de análise (seg)
DMO 5 + A	208	1956.0
DMO 1 + A+ B	208	364.8

Tabela 51: Número de iterações e tempos de análises usando os esquemas 1, 4 e 5 do DMO

Verificando que os resultados obtidos neste sub-capítulo são idênticos aos obtidos com o esquema 5 do DMO e tendo em conta a grande diferença ao nível do tempo de optimização, optou-se por utilizar nas próximas análises o esquema 1 apresentado na secção 4.3 em conjunto com as duas funções penalizadoras.

4.4.4 Utilizando diferenças finitas

Nesta secção é feito um estudo equivalente ao realizado na secção anterior, mas com o cálculo das derivadas da função objectivo realizado por diferenças finitas. Esta análise é feita utilizando 6 variáveis de projecto, com os mesmos candidatos apresentados nos casos anteriores, e com recurso ao esquema 1 do método DMO (secção 4.3), em conjunto com as duas funções penalizadoras.

DMO 1+Penalizações $x_0 = [0.5 \dots 0.5] \quad p = 1$ $q = 1$	Condições de fronteira	$C\ [J]$	Iterações até convergir	Tempo de análise
1 Elemento / 1 Furo $k = 0.01$ / $c = 0.5$		0.0176151711104	203	343.4 seg
4 Elemento / 1 Furo $k = 4$ / $c = 60$		1.2612593408490	203	402.7 seg
1 Elemento / 9 Furos $k = 50$ / $c = 300$		48.009650689943	216	534.2 seg

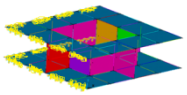
1 Elemento / 1 Furo $k = 1$ / $c = 5$ C.F. $x = 0$ $y = 4$		0.5354557556962	138	269.0 seg

Tabela 52: Solução da função objectivo usando esquema 1 do DMO com penalizações

DMO 1+Penalizações $x_0 = [0.5 \dots 0.5]$ $p = 1$ $q = 1$	$w1$	$w2$	$w3$	$w4$	$w5$	$w6$
1 Elemento / 1 Furo $k = 0.01$ / $c = 0.5$	0.0000	**1.0000**	0.0000	0.0000	**1.0000**	1.9058e-2
4 Elemento / 1 Furo $k = 4$ / $c = 60$	**1.0000**	0.0000	2.0101e-3	**1.0000**	2.0106e-3	1.4901e-8
1 Elemento / 9 Furos $k = 50$ / $c = 300$	**1.0000**	7.8614e-3	0.0000	**1.0000**	0.0000	7.8611e-3
1 Elemento / 1 Furo C.F. $x = 0$ $y = 4$ $k = 1$ / $c = 5$	0.0000	0.0000	**1.0000**	2.7500e-2	0.0000	**1.0000**

Tabela 53: Solução das funções de peso usando esquema 1 do DMO com penalizações

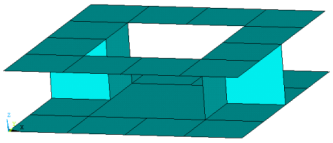
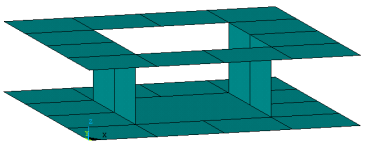
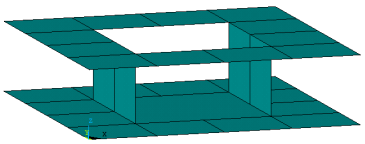
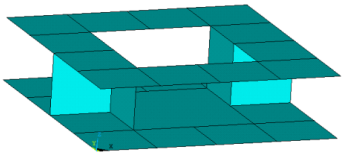
	Soluções
1 Elemento / 1 Furo $k = 0.01$ / $c = 0.5$ $w2 = 1 = w5$	
4 Elemento / 1 Furo $k = 4$ / $c = 60$ $w1 = 1 = w4$	
1 Elemento / 9 Furos $k = 50$ / $c = 300$ $w1 = 1 = w4$	
1 Elemento / 1 Furo C.F. $x = 0$ $y = 4$ $w3 = 1 = w6$	

Tabela 54: Solução obtidas usando esquema 1 do DMO com penalizações

Os resultados mostram soluções óptimas de acordo com as previstas e completamente convergidos. Em todos os casos são identificados os melhores candidatos (com função de peso igual a 1), e praticamente todos os restantes candidatos apresentam função de peso igual

a zero. De referir a evidente alteração na escolha da solução óptima, quando se refina a malha utilizada ou se aumenta o número de furos da placa. Como esta análise foi realizada apenas para seis variáveis de projecto, o caso em que as condições de fronteira deixam de ser simétricas não tem implicações na escolha dos candidatos e na simetria dos reforços, pois estamos a limitar a sua localização a apenas dois lados. Prevê-se por isso, que essa simetria deixe de existir quando o estudo for feito para doze variáveis de projecto.

Passemos então à análise onde se utilizam doze variáveis de projecto. No caso das condições de fronteira simétricas, prevê-se que os reforços sejam colocados nos lados mais próximos dos encastramentos pois parece ser a melhor opção para aumentar a rigidez da placa.

Neste caso, começou-se pelo estudo do caso mais simples, onde se utiliza apenas um furo e um elemento por área, com a placa encastrada em $y = 0$ e $y = 4$. Tendo-se obtido os resultados esperados, prosseguiu-se a análise, agora utilizando quatro elementos por área gerada. Mais uma vez, os resultados obtidos enquadram-se com o espectavel e, assim, passou-se ao estudo da placa com doze variáveis de projecto, dezasseis furos e um elementos por área. Novamente, obtiveram-se resultados satisfatórios e, por isso, decidiu-se repetir a última análise mas desta vez refinando a malha, passando a ter quatro elementos por área. Os resultados obtidos nas análises anteriores encontram-se expostos nas tabelas seguintes.

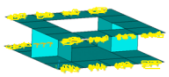
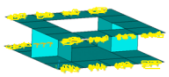
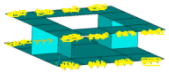
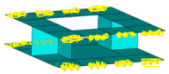
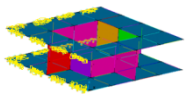
DMO 1+Penalizações $x_0 = [0.5 \dots 0.5] \quad p = 1$ $q = 1$	Condições de fronteira	$C\ [J]$	Iterações até convergir	Tempo de análise
1 Elemento / 1 Furo $k = 0.01$ / $c = 0.01$		0.0044172244466	317	1117.7 seg
4 Elemento / 1 Furo $k = 0.05$ / $c = 0.05$		0.1160305411498	850	3187.0 seg
1 Elemento / 16 Furo $k = 50$ / $c = 300$		12.549670767363	593	3050.0 seg
4 Elemento / 16 Furo $k = 50$ / $c = 300$		21.732721963113	841	15906.3 seg
4 Elemento / 16 Furo C.F. $x = 0$ $y = 4$ $k = 50$ / $c = 300$		36.573789368341	553	3015.6 seg

Tabela 55: Solução da função objectivo usando esquema 1 do DMO com penalizações

DMO 1+Penalizações $x_0 = [0.5 \dots 0.5] \quad p = 1$ $q = 1$	$w1$	$w2$	$w3$	$w4$	$w5$	$w6$
1 Elemento / 1 Furo $k = 0.01$ / $c = 0.01$	0.0000	3.1880e-2	3.1893e-2	**1.0000**	0.0000	0.0000
4 Elemento / 1 Furo $k = 0.05$ / $c = 0.05$	2.3862e-2	1.4793e-2	1.5077e-2	0.0000	**1.0000**	5.4829e-2
1 Elemento / 16 Furo $k = 50$ / $c = 300$	2.5111e-2	0.0000	2.9965e-3	3.8767e-3	**1.0000**	0.0000
4 Elemento / 16 Furo $k = 50$ / $c = 300$	3.3937e-2	6.4564e-4	2.3807e-3	0.0000	**1.0000**	2.2921e-2
4 Elemento / 16 Furo C.F. $x = 0$ $y = 4$ $k = 50$ / $c = 300$	0.0000	7.9032e-3	**1.0000**	0.0000	3.7507e-2	1.2588e-2

Tabela 56: Solução das funções de peso usando esquema 1 do DMO com penalizações

DMO 1+Penalizações $x_0 = [0.5 \dots 0.5] \quad p = 1$ $q = 1$	$w7$	$w8$	$w9$	$w10$	$w11$	$w12$
1 Elemento / 1 Furo $k = 0.01$ / $c = 0.01$	0.0000	3.1882e-2	3.1894e-2	**1.0000**	0.0000	1.4901e-8
4 Elemento / 1 Furo $k = 0.05$ / $c = 0.05$	2.0539e-2	1.4819e-2	1.4929e-2	0.0000	5.5166e-2	**1.0000**
1 Elemento / 16 Furo $k = 50$ / $c = 300$	4.8919e-2	1.6385e-3	2.4901e-3	9.1490e-4	0.0000	**1.0000**
4 Elemento / 16 Furo $k = 50$ / $c = 300$	3.3865e-2	6.4536e-4	2.3814e-3	2.2847e-2	**1.0000**	0.0000
4 Elemento / 16 Furo C.F. $x = 0$ $y = 4$ $k = 50$ / $c = 300$	1.3903e-2	5.0361e-2	1.3805e-2	**1.0000**	2.0885e-2	0.0000

Tabela 57: Solução das funções de peso usando esquema 1 do DMO com penalizações

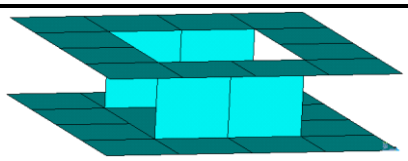
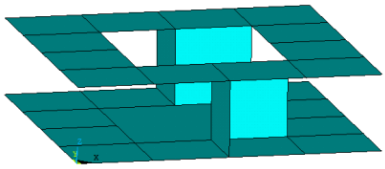
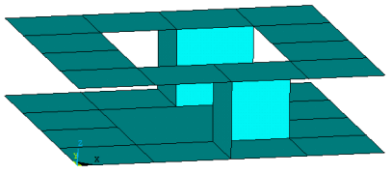
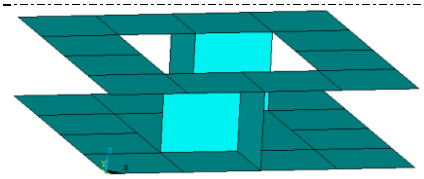
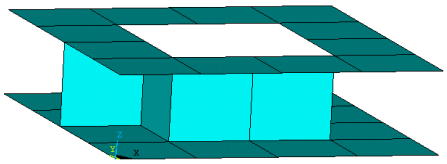
Soluções	
1 Elemento / 1 Furo $k = 0.01$ / $c = 0.01$ $w4 = 1 = w10$	
4 Elemento / 1 Furo $k = 0.05$ / $c = 0.05$ $w5 = 1 = w12$	
1 Elemento / 16 Furo $k = 50$ / $c = 300$ $w5 = 1 = w12$	
4 Elemento / 16 Furo $k = 50$ / $c = 300$ $w5 = 1 = w11$	
4 Elemento / 16 Furo C.F. $x = 0$ $y = 4$ $k = 50$ / $c = 300$ $w3 = 1 = w10$	

Tabela 58: Solução obtidas usando esquema 1 do DMO com penalizações

Em todos os casos analisados se obtiveram resultados coerentes. Tal como pretendido, o programa opta sempre por escolher apenas um candidato, apresentando a sua função de peso igual a 1, sendo as dos restantes candidatos muito próximas de 0.

Novamente, com o refinamento da malha utilizada, o programa tende a indicar diferentes soluções óptimas. Isto verifica-se tanto para um como para dezasseis furos. Para além disto e tal como se previa, observa-se que todos os reforços foram colocados nos lados mais próximos dos encastramentos, para condições de fronteira simétricas. No caso em que se altera as condições de fronteira a melhor solução é aquela em que se coloca os reforços em lados adjacentes.

De referir que, as condições de fronteira aplicadas ao problema evideciam uma elevada influência no valor da função objectivo. Quando deixamos de ter condições simétricas dá-se um aumento de aproximadamente 68% no valor da função objectivo, o que se traduz numa redução brusca na rigidez da placa.

Tendo em conta os resultados obtidos neste sub-capítulo será exposta, no capítulo seguinte, uma comparação entre os resultados apresentados pelo FAIPA e o FMINCON com o cálculo da derivada da função objectivo feito por diferenças finitas.

4.4.5 FMINCON vs FAIPA

De forma a verificar qual o optimizador que melhor se enquadra à resolução do problema em questão, optou-se por fazer uma comparação dos resultados obtidos utilizando o FAIPA e o FMINCON. Nestas análises será utilizado o método apresentado na secção 4.3 , pois foi aquele que permitiu obter a melhor relação entre precisão da solução e tempo de computação. Pela mesma razão, as derivadas da função serão calculados por diferenças finitas.

De seguida são apresentados os resultados obtidos para os casos de seis e doze varáveis de projecto, quando temos uma placa com um furo e uma malha de um elemento por área.

DMO 1+Penalizações $x_0 = [0.5 \dots 0.5]$ $p = 1$ $q = 1$ 1 Elemento 1 Furo	$C\ [J]$	Iterações até convergir	Tempo de análise
FAIPA 6 var $k = 0.01$ / $c = 0.5$	0.0246806968743	208	630.0 seg
FAIPA 12 var $k = 0.01$ / $c = 0.5$	0.0052516214128	1275	3833.7 seg
FMINCON 6 var $k = 0.01$ / $c = 0.5$	0.0176151711104	203	343.4 seg
FMINCON 12 var $k = 0.01$ / $c = 0.01$	0.0031417289928	317	1117.7 seg

Tabela 59: Solução da função objectivo usando o FAIPA e o FMINCON

DMO 1+Penalizações $x_0 = [0.5 \dots 0.5]$ $p = 1$ $q = 1$ 1 Elemento 1 Furo	$w1$	$w2$	$w3$	$w4$	$w5$	$w6$
FAIPA 6 var $k = 0.01$ / $c = 0.5$	5.5815e-14	0.50631	0.50632	4.8255e-14	0.50632	0.50631
FAIPA 12 var $k = 0.01$ / $c = 0.5$	1.4586e-10	2.2512e-2	2.2656e-2	**0.97456**	1.4586e-10	1.4586-10
FMINCON 6 var $k = 0.01$ / $c = 0.5$	0.0000	**1.0000**	0.0000	0.0000	**1.0000**	1.9058e-2
FMINCON 12 var $k = 0.01$ / $c = 0.01$	0.0000	3.1880e-2	3.1893e-2	**1.0000**	0.0000	0.0000

Tabela 60: Solução das funções de peso usando o FAIPA e o FMINCON

DMO 1+Penalizações $x_0 = [0.5 \dots 0.5]$ $p = 1$ $q = 1$ 1 Elemento 1 Furo	$w7$	$w8$	$w9$	$w10$	$w11$	$w12$
FAIPA 6 var $k = 0.01$ / $c = 0.5$	-	-	-	-	-	-
FAIPA 12 var $k = 0.01$ / $c = 0.5$	1.4586e-10	2.2750e-2	2.2424e-2	**0.97457**	1.4586e-10	6.0556e-6
FMINCON 6 var $k = 0.01$ / $c = 0.5$	-	-	-	-	-	-
FMINCON 12 var $k = 0.01$ / $c = 0.01$	0.0000	3.1882e-2	3.1894e-2	**1.0000**	0.0000	1.4901e-8

Tabela 61: Solução das funções de peso usando o FAIPA e o FMINCON

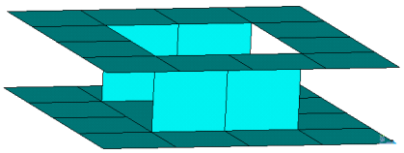
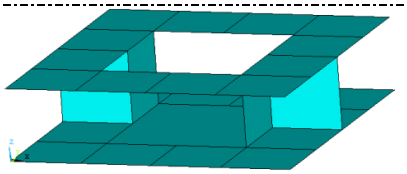
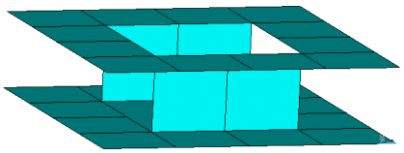
	Soluções
FAIPA 6 var $k = 0.01$ / $c = 0.5$	---
FAIPA 12 var $k = 0.01$ / $c = 0.5$ $w4 = 1 = w10$	
FMINCON 6 var $k = 0.01$ / $c = 0.5$ $w2 = 1 = w5$	
FMINCON 12 var $k = 0.01$ / $c = 0.01$ $w4 = 1 = w10$	

Tabela 62: Solução obtidas usando o FAIPA e o FMINCON

Analisando os resultados obtidos para seis varáveis de projecto, verifica-se que o optimizador FAIPA não consegue optar por apenas um dos candidatos w_2/w_3 e w_5/w_6 , preferindo apresentar uma solução "cinzenta" e irrealista face ao que se pretende. Já o FMINCON, escolhe os candidatos w_2 e w_5 como solução óptima e apresenta funções de peso com os valores pretendidos de 1 e 0.

Quanto ao estudo realizado para doze variáveis de projecto, ambos os optimizadores conseguem apresentar uma boa solução. No entanto, utilizando o FMINCON consegue-se uma

solução mais precisa, com funções de peso mais próximas dos seus valores extremos e menor valor da função objectivo.

Quanto ao tempo de análise, verifica-se que a utilização de diferenças finitas no FAIPA torna-o bastante mais lento que o FMINCON. Assim, necesssita aproximadamente o dobro do tempo, para o estudo do caso de seis variáveis de projecto, e mais do triplo do tempo para a análise do modelo com doze variáveis de projecto.

Tendo em conta a precisão dos resultados obtidos e os custos ao nível do tempo de computação, será muito mais benéfico utilizar o FMINCON em análises com funções para as quais não se dispõe de derivadas explícitas, pois para além de apresentar melhores soluções, tem um tempo de análise claramente inferior.

4.5 Caso 5 - 192 variáveis de projecto

De maneira a mostrar que esta nova forma de utilizar o método DMO funciona em casos mais complexos, decidiu-se realizar uma análise onde são utilizadas 192 variáveis de projecto. Este caso consiste em analisar uma estrutura quadrada com 16 furos independentes, cada um deles com 12 variáveis de projecto. As configurações possíveis são idênticas às apresentadas nos casos anteriores onde eram utilizadas 12 variáveis de projecto, sendo os lados 1/2 e 3/4 dependentes entre si, isto é, o optimizador escolhe apenas um lado para reforçar (ex: escolhe lado 1 ou o 2).

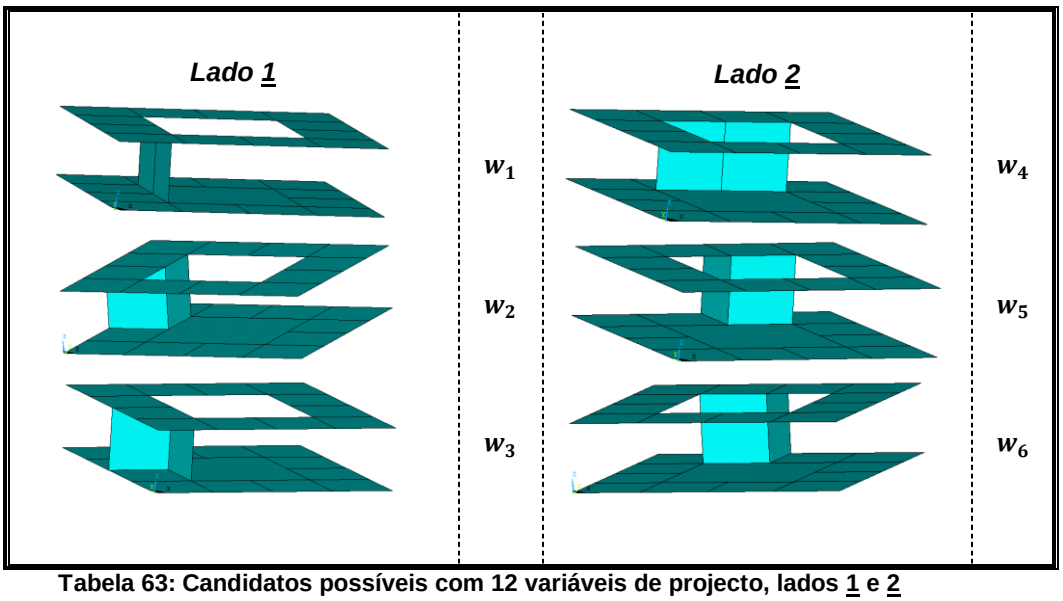


Tabela 63: Candidatos possíveis com 12 variáveis de projecto, lados 1 e 2

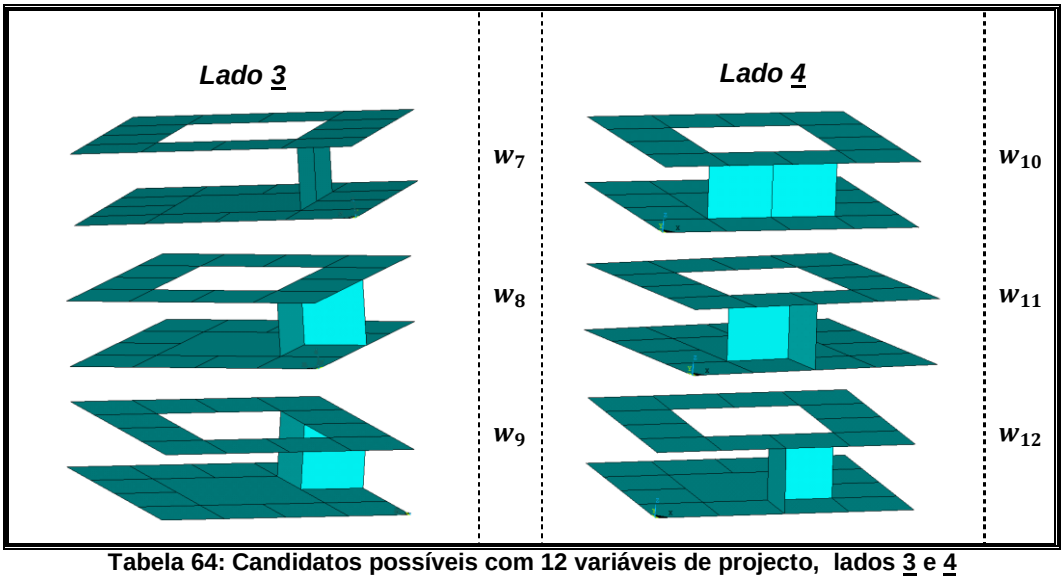


Tabela 64: Candidatos possíveis com 12 variáveis de projecto, lados 3 e 4

Este estudo será realizado com o cálculo das derivadas por diferenças finitas, de forma a analisar qual o custo computacional de uma análise mais complexa. Será utilizada a função FMINCON para realizar a optimização do problema. De forma a diminuir o tempo de análise optou-se por utilizar o esquema 1 do DMO (ver secção 4.3) em conjunto com as duas funções de penalização indicas.

De forma a simplicar a apresentação dos resultados obtidos é apresentada a próxima imagem, onde são numerados todos os furos.

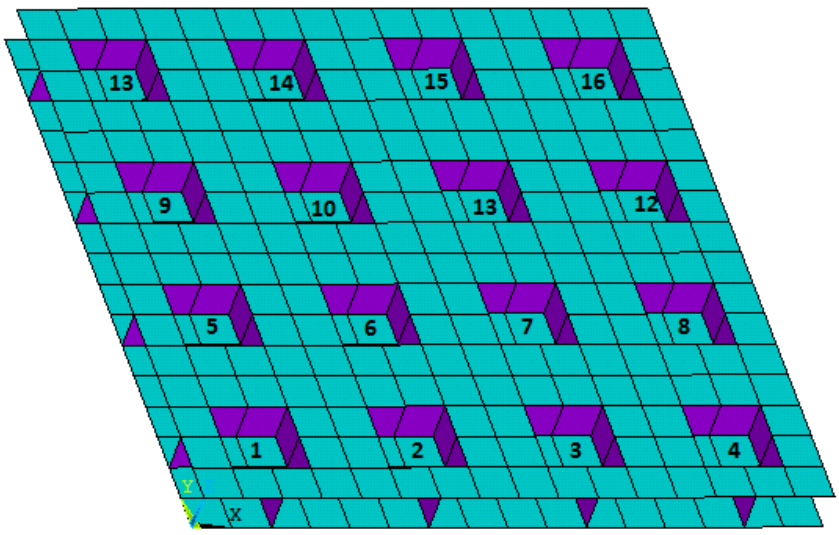


Figura 22: Numeração dos recortes de reforço

Devido à complexidade da análise, este caso de estudo denotou uma convergência muito lenta, assim, terminou-se o processo por se ter atingido o número limite de iterações.

Após a análise ter terminado obtiveram-se os seguintes resultados.

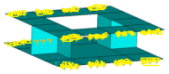
$x_0 = [0.5 \dots 0.5] \quad p = 1$ $q = 1$ 1 Elemento	Condições de fronteira	$C\ [J]$	Iterações até convergir	Tempo de análise
$k = 0.01$ / $c = 0.5$		379.7605463146778	31120	1167872 seg

Tabela 65: Solução da função objectivo – placa com 192 variáveis

	$x_0 = [0.5 \dots 0.5]$		$p = 1$		$q = 1$	$k = 50$		$c = 300$	1Elemento			
	$w1$	$w2$	$w3$	$w4$	$w5$	$w6$	$w7$	$w8$	$w9$	$w10$	$w11$	$w12$
Furo 1	0	0.48498	0	0	0	0.48498	0	**1**	0	0	0	0
Furo 2	0.50154	0.50154	0	0	0	0	**1**	0	0	0	0	0
Furo 3	0.21381	0.21381	0.21381	0	0.21381	0	0.50983	0.50983	0	0	0	0
Furo 4	0.21256	0.21256	0.21256	0.21256	0	0	**1**	0	0	0	0	0
Furo 5	0.30576	0.30576	0	0	0.30576	0	0	0	0	**1**	0	0
Furo 6	0	0	0	0	0	**1**	0.29089	0.29089	0	0	0	0.29089
Furo 7	0	0	0	0	0	**1**	0	0	0	0	**1**	0
Furo 8	**1**	0	0	0	0	0	0	0	0	0	**1**	0
Furo 9	0	0	0	**1**	0	0	0	0	0	0	0	**1**
Furo 10	0	0	0	0	0	**1**	0	0	0	0	**1**	0
Furo 11	0	0	0	0	0	**1**	0	0	**1**	0	0	0
Furo 12	0.49870	0	0	0.4870	0	0	0	0	0	0	**1**	0
Furo 13	0	0	0	0	0	**1**	**1**	0	0	0	0	0
Furo 14	**1**	0	0	0	0	0	0	0.49432	0.49432	0	0	0
Furo 15	0	0	0	0	**0.99628**	0	0	0	0	0	0	**1**
Furo 16	0	0	0.49866	0	0	0.49866	**1**	0	0	0	0	0

Tabela 66: Solução das funções de peso – placa com 192 variáveis

Solução Final

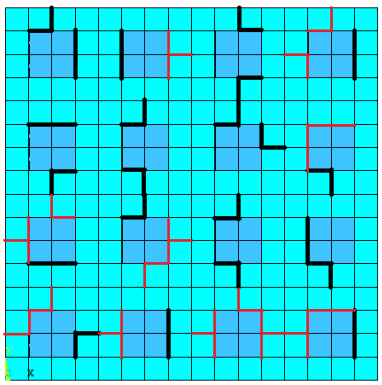


Tabela 67: Configuração final

Podemos verificar que, tal como pretendido, existe uma clara convergência das variáveis de projecto para os seus valores extremos de 1 e 0. Na imagem anterior podemos observar a preto os reforços com $w_i = 1$, sendo que a vermelho estão representados os candidatos com valores intermédios, isto é, que ainda não convergiram. Tendo em conta a evolução da análise prevê-se que, se esta não tivesse sido parada, se teria atingido a solução final totalmente convergida, isto é, com a escolha de uma, e apenas uma, configuração para cada um dos reforços.

Tendo em conta o tempo de análise, podemos concluir que o esquema 1 do método DMO é eficaz neste tipo de análises. Isto aliado ao facto de ser o método mais simples e com menor custo computacional, faz com que seja a melhor opção para análises com elevado número de variáveis de projecto. Quanto ao cálculo da derivada da função objectivo por diferenças finitas, verificou-se que quanto maior for o número de variáveis de projecto utilizadas mais lenta será a convergência. Por isso, apesar de ser eficaz, é preferivel utilizar métodos analíticos (ex: método adjunto) para calcular o gradiente da função objectivo pois estes permitem obter um menor tempo de análise. No entanto será uma boa opção para casos em que não se dispõe da derivada explicita.

4.6 Caso 6 - Casca cilíncrica

4.6.1 Casca cilíncrica com 1 furo

Depois de verificada e confirmada a viabilidade do método em estruturas planas, prosseguiu-se com o estudo para o caso de estruturas cilíndricas, de forma a mostrar que o método também é aplicável a estruturas não planas. Nesta secção o gradiente da função objectivo será calculado com recurso a métodos analíticos (método adjunto) e será utilizado a função optimizadora FMINCON. A primeira análise será realizado para uma placa cilíndrica escastrada apenas numa das extremidades. A placa possui um furo com 12 configurações de reforço possíveis, sendo as hipóteses de reforço idênticas aos casos anteriores. É ainda aplicada uma pressão à placa superior da estrutura (a vermelho na figura 23). Tendo em conta os resultados obtidos até aqui, optou-se por utilizar o esquema 5 do método de DMO em conjunto com a penalização de valores intermédios.

Na imagem seguinte é possivel observar o painel que se vai análisar.

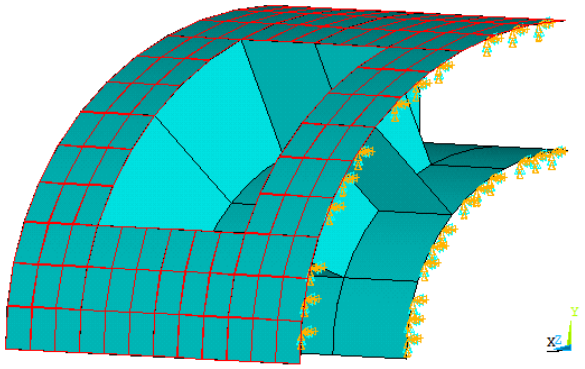


Figura 23: Estrutura cilíndrica

Assim, realizou-se uma análise estática à estrutura, tendo-se utilizado uma pressão de $150\ N/m^2$. A malha escolhida para a análise das estruturas cilíndricas utilizada nove elementos por área gerada. A tabela seguinte apresenta os resultados obtidos neste caso.

$x_0 = [0.5 \ldots 0.5]$ $p = 2 \quad q = 1$	$C\ [J\ ]$	Iterações até convergir	Tempo de análise
DMO5 $k = 50$	0.15475996779229	325	9824 seg

Tabela 68: Solução da função objectivo – placa cilíndrica com pressão aplicada

$x_0 = [0.5 \ldots 0.5]$ $p = 2 \quad q = 1$	$w1$	$w2$	$w3$	$w4$	$w5$	$w6$
DMO5 $k = 50$	**0.99972**	1.9359e-4	8.4632e-5	0.0000	0.0000	0.0000

Tabela 69: Solução das funções de peso – placa cilíndrica com pressão aplicada

$x_0 = [0.5 \ldots 0.5]$ $p = 2 \quad q = 1$	$w7$	$w8$	$w9$	$w10$	$w11$	$w12$
DMO5 $k = 50$	0.0000	0.0000	0.0000	0.0000	0.0000	**1.0000**

Tabela 70: Solução das funções de peso – placa cilíndrica com pressão aplicada

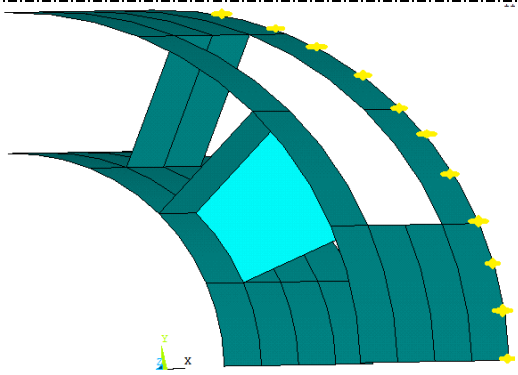
Solução


Tabela 71: Configuração final – placa cilíndrica com pressão aplicada

A partir da imagem da tabela 71 podemos verificar que o método funciona para estruturas cilíndricas. Neste caso é escolhido um reforço perpendicular ao encastramento e outro reforço em L no lado oposto ao encastramento.

O próximo caso de estudo consiste em aplicar uma pequena alteração ao caso anterior. No lugar da pressão aplicada anteriormente à estrutura será utilizada uma carga pontual no lado esquerdo da estrutura e perpendicular à mesma. Essa carga terá o valor de $1500\ N$. Podemos observar, a vermelho, a nova carga imposta à placa.

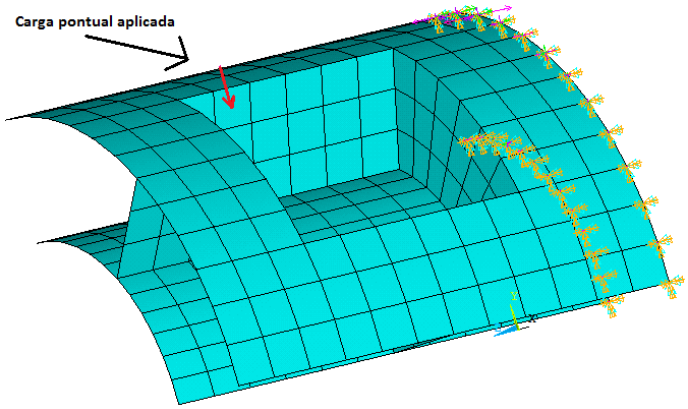


Figura 24: Estrutura cilíndrica com carga pontual

Neste caso, prevê-se que o optimizador altere os reforços escolhidos no caso anterior. Teoricamente será previsível que seja aplicado um reforço em L no lado em que a carga é aplicada, de forma a diminuir a deformada da estrutura. De seguida são apresentados os resultados obtidos.

$x_0 = [0.5 \dots 0.5]$ $p = 2 \quad q = 1$	$C\ [J]$	Iterações até convergir	Tempo de análise
DMO5 $k = 50$	0.12521829931845	338	6413 seg

Tabela 72: Solução da função objectivo – placa cilíndrica com carga pontual aplicada

$x_0 = [0.5 \dots 0.5]$ $p = 2 \quad q = 1$	$w1$	$w2$	$w3$	$w4$	$w5$	$w6$
DMO5 $k = 50$	0.0000	**1.0000**	0.0000	0.0000	0.0000	0.0000

Tabela 73: Solução das funções de peso – placa cilíndrica com carga pontual aplicada

$x_0 = [0.5 \dots 0.5]$ $p = 2 \quad q = 1$	$w7$	$w8$	$w9$	$w10$	$w11$	$w12$
DMO5 $k = 50$	0.0000	**1.0000**	0.0000	0.0000	0.0000	0.0000

Tabela 74: Solução das funções de peso – placa cilíndrica com carga pontual aplicada

Solução

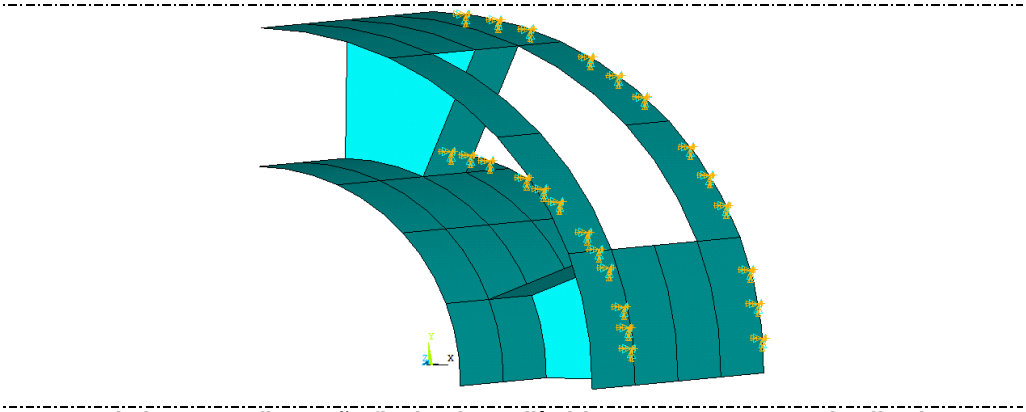


Tabela 75: Configuração final - placa cilíndrica com carga pontual aplicada

De acordo com o previsto, verifica-se que é escolhido um reforço em L no local de aplicação da carga. Mais uma vez, é evidente a qualidade da solução obtida.

4.6.2 Casca cilíncrica com 4 furos

O próximo caso de estudo pretende provar que esta forma de utilizar o método DMO pode ser utilizada em estruturas não plana e com maiores dimensões. Aqui será utilizada uma placa cilíndrica com 4 furos. Devido ao elevado tempo de análise para casos com mais de 12 variáveis de projecto, a todos os furos serão aplicados reforços iguais, o que equivale a dizer que se utilizam novamente 12 variáveis de projecto, sendo as configurações de reforço idênticas ao caso anterior.

Será, de novo, utilizado o método adjunto para calcular o gradiente da função objectivo, a função optimizadora FMINCON e o esquema 5 do método DMO em conjunto com a penalização A, de valores intermédios. A estrutura cilíndrica é escastrada apenas numa das extremidades sendo aplicada uma pressão de $150\,N/m^2$ na placa superior da estrutura e utilizada uma malha com nove elementos por área.

Nestas novas condições os resultados obtidos foram os seguintes.

$x_0 = [0.5 \ldots 0.5]$ $p = 2 \quad q = 1$		$C\ [J]$	Iterações até convergir	Tempo de análise
DMO5	$k = 200$	2.29714982586376	1066	28854 seg

Tabela 76: Solução da função objectivo - placa cilíndrica com 4 furos

$x_0 = [0.5 \ldots 0.5]$ $p = 2 \quad q = 1$		$w1$	$w2$	$w3$	$w4$	$w5$	$w6$
DMO5	$k = 200$	2.4716e-4	1.8292e-4	**0.9991**	1.4221e-4	1.4597e-4	1.4422e-4

Tabela 77: Solução das funções de peso - placa cilíndrica com 4 furos

$x_0 = [0.5 \dots 0.5]$ $p = 2 \quad q = 1$		$w7$	$w8$	$w9$	$w10$	$w11$	$w12$
DMO5	$k = 200$	9.4015e-7	1.1252e-6	1.4715e-7	6.4248e-4	**0.9993**	2.2460e-5

Tabela 78: Solução das funções de peso - placa cilíndrica com 4 furos

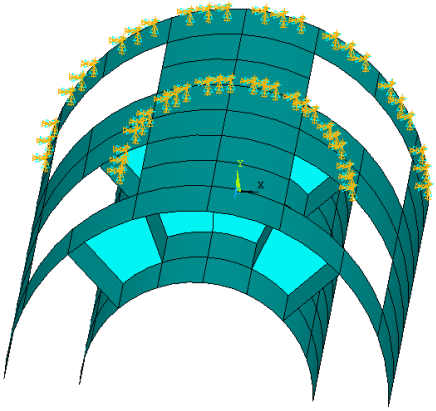
Solução


Tabela 79: Configuração final - placa cilíndrica com 4 furos

Neste caso a solução de reforço escolhida foi $w_3 = 1$ e $w_{11} = 1$.

Mostra-se assim que o método produz bons resultados em estruturas cilíndricas com múltiplos recortes de reforço e variáveis de projecto.

4.6.3 Estudo de convergência

É importante referir que a malha utilizada foi escolhida de modo a que o problema não apresentasse um elevado custo ao nível do tempo de cálculo. Assim, foi realizado um estudo de convergência de forma a verificar o efeito da malha utilizado na qualidade da solução obtida. O estudo compara o valor da energia elástica para três configurações diferentes, no caso de uma placa cilíndrica com um furo, na qual é aplicada uma pressão idêntica à das secções anteriores.

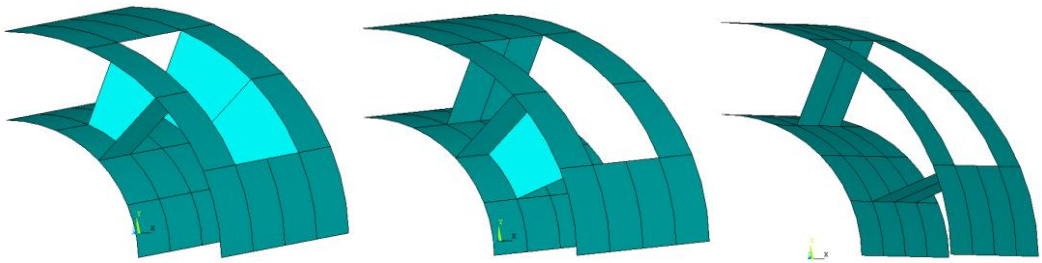


Figura 25: Configurações -- w4 e w11 -- w1 e w12 -- w1 e w7 respectivamente

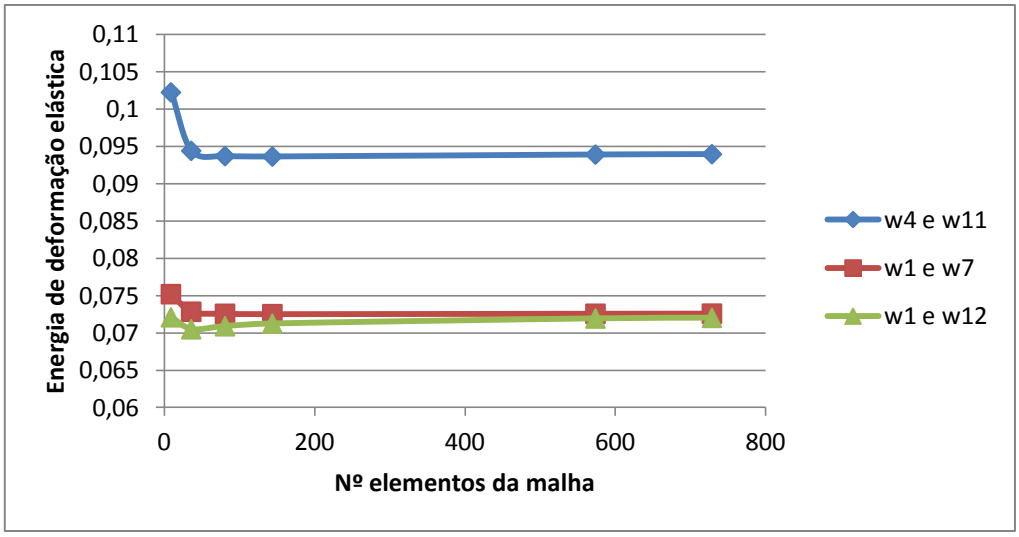


Figura 26: Estudo convergência da malha

É possível verifficar que a configuração óptima identificada no subcapítulo 4.6.1 é efectivamente a melhor opção de entre as três, tanto no caso de se utilizar uma malha com 9 elementos como quando se utiliza uma malha com 729 elementos. Para malhas mais grosseiras verifica-se que a solução é um pouco diferente da que se obtem com malhas mais refinadas e, por isso, quando se pretende fazer estudos mais precisos (ex: estudar tensões) convém utilizar malhas mais refinadas. No entanto, malhas menos refinadas podem ser utilizadas para idêntificar qual a configuração óptima, pois o candidato a configuração óptima não é alterado com o refinamento da malha.

Capítulo 5

Conclusões

Nesta dissertação de mestrado pretendia-se desenvolver um modelo computacional de optimização discreta que permitisse escolher, a partir de um conjunto de possibilidades, a melhor forma de reforçar um painel OPENCELL plano cilindrico ou plano. Este reforço tinha como objectivo maximizar a rigidez da estrutura. Para formular o problema de optimização utilizou-se o método DMO de um modo diferente do habitual. Neste caso, as variáveis de projecto representam diferentes configurações de reforço, em vez dos tradicionais tipos de materiais.

De forma a verificar se esta nova formulação do método DMO era viável, começou-se por estudar casos mais simples, com menor número de variáveis de projecto. Com a obtenção de resultados concordantes com o previsto, seguiram-se os exemplos mais complexos, onde se aumenta o número de configurações possiveis para os reforços.

A partir dos resultados obtidos, verificou-se que todos os esquemas do DMO testados apresentam prós e contras. Inicialmente, o esquema 4 do DMO parecia ser o mais adequado já que, ao contrário do esquema 5, permitia obter alguns resultados coerentes, com valores das funções de peso próximos de 1 e 0. No entanto, o esquema apresentava limitações quando o valor inicial das variáveis de projecto era igual para todas elas e quando se encontrava próximo de 0.5. Para além disso, com o aumento do número de variáveis de projecto, o esquema voltava a apresentar soluções “cinzentas”.

Com a introdução de uma penalização explícita de valores intermédios, o esquema 5 do DMO começa a exibir as soluções que se pretendia, com as funções de peso a assumirem valores próximos dos seus extremos. Ainda assim, o programa continuava a apresentar soluções irreais, para valores iniciais das variáveis de projecto próximos de 0.5. Este problema foi resolvido com a utilização do optimizador FMINCON da toolbox do MATLAB em detrimento do FAIPA, optimizador que se utilizou inicialmente. Quanto ao esquema 4, a utilização desta penalização não permitiu solucionar os problemas evidenciados anteriormente.

Foi ainda testado o esquema 1 do DMO com recurso a duas penalizações. Foi utilizada uma penalização de valores intermédios e outra função penalizadora que garantisse que a soma das funções de peso fosse igual a 1. Com recurso a este esquema e ao optimizador FMINCON conseguiram-se resultados idênticos aos obtidos com o esquema 5. Para além disso, devido a sua simplicidade, relativamente ao esquema 5, o tempo de análise neste caso é substancialmente inferior. Isto faz com que seja uma melhor opção para estudos com elevado número de variáveis de projecto, onde o custo computacional e o tempo de análise são elevados.

Os resultados obtidos com recurso aos dois optimizadores, o FAIPA e o FMINCON, permitiram mostrar que o FMINCON é o mais indicado para este tipo de análises. Apesar de ser um pouco mais lento na maioria dos casos, permitiu resolver o problema evidenciado pelo esquema 5 do DMO quando se assumia 0.5 como valor inicial das variáveis de projecto.

Tal como seria de esperar, o aumento do número de variáveis de projecto torna o problema mais complexo, e isso tem repercussões no tempo necessário à análise do problema, aumentando-o significativamente.

Relativamente às condições de fronteira testaram-se dois casos diferentes para o painel plano, um simétrico (placa encastrada em $y = 0$ e $y = 4$) e outro sem ser simétrico (placa encastrada em $x = 0$ e $y = 4$). De acordo com o previsto, o caso em que temos condições de fronteira simétricas tem tendência a apresentar soluções de reforços simétricas, pois a carga aplicada é igual em toda a placa. O mesmo já não é verdade no segundo caso, pois o facto de a placa não estar apoiada em lados oposto cria diferentes tensões ao longo da mesma o que faz com que os reforços ideais não apresentem simetria. Para além disto, quando deixamos de ter condições simétricas verifica-se um aumento de aproximadamente 68% no valor da função objectivo, o que se traduz numa redução brusca na rigidez da placa. Este facto também seria de prever já que, o constrangimento de lados adjacentes de uma placa quadrangular permite que hajam maiores deslocamentos na mesma, se compararmos com caso em que se constrangem lados opostos.

Posteriormente foi estudado um painel cilíndrico com recurso à função FMINCON. Por ser o esquema que permitiu obter melhores resultados nos estudos realizados no painel plano, optou-se por utilizar o esquema 5 do método DMO, em conjunto com a penalização de valores intermédios. Mais uma vez, obtiveram-se resultados positivos já que em todos os casos estudados foi sempre encontrada a solução óptima de reforço da estrutura. Verificou-se ainda que a alteração da carga aplicada à estrutura altera o reforço da mesma.

Concluindo, pode-se afirmar que esta forma inovadora de utilizar o DMO é viável, permitindo obter bons resultados tanto para painéis planos como cilíndricos, sendo esquema 5, em conjunto com a penalização de valores intermédios, o mais indicado para este tipo de análises. Quando se pretende aumentar muito o número de variáveis de projecto, a utilização de um esquema mais simples, como o caso do esquema 1 conjuntamente com as duas funções penalizadoras (equações (4.7) e (4.8)), traz benefícios ao nível do custo computacional.

Capítulo 6

Trabalho Futuro

Como já foi referido, os paineis OPENCELL são uma inovadora e recente classe de estruturas sanduíche. A forma como se utilizou o método DMO nesta tese é também inovadora, pois usou-se configurações como candidatos, em vez da tradicional utilização da densidade de material numa placa. Assim sendo, existe ainda um enorme campo de análise e estudo, não só ao nível deste tipo de painéis, mas principalmente nesta forma de utilização do DMO. Sugerem-se os seguintes trabalhos futuros:

- Estudo de diferentes tipos de corte (ex: triangular), de padrões de reforço (repetir sequência de 2, 4 ou mais furos idênticos) ou tipo de reforço (ex: dobrar reforço a 45º), com uma análise semelhante aquela que aqui foi realizada, de forma a verificar qual a melhor forma de reforçar a estrutura.
- Seria interessante estudar outra forma de organizar os reforços, permitindo com o corte e dobragem assegurar o reforço dos quatro lados do recorte quadrangular.
- Estudo e análise de vibrações da estrutura, em vez da maximização da rigidez.
- Análise semelhante à realizada neste trabalho, mas utilizando o método DMO como um “switch”, permitindo escolher desta forma, qual o lado do corte quadrangular a reforçar.
- Análise semalhante à realizada nesta tesa, mas utilizando outro tipo de funções de peso, no lugar do método DMO, que se possam adquar melhor ao problema.
- Estudar a influência de diferentes tipos de carga aplicada e condições de fronteira (cargas pontuais, diferentes pressões, placa encastrada, placa simplesmente apoiada, entre outras).
- Estudar estruturas com aplicação de multiplas cargas.
- Estudo de casos mais complexos, com elevado número de variáveis de projecto.

Bibliografia

[1] Vinson, Jack R., (1999), “The Behavior of Sandwich Structures of Isotropic and Composite Materials”, CRC Press, USA.

[2] Allen H., (1993), “Analysis and Design of Structural Sandwich Panels”, Pergamon Press, UK.

[3] Gibson, L.J., Ashby M.F., (1999), “Cellular Solids - Structures and Properties”, Second Edition, Cambridge University Press, UK.

[4] Freitas, Manuel., (2011) “Compósitos Sandwich”, Materiais Compósitos laminados, IST.

[5] Almeida, I.A., Correia, J.R., Branco, F.A., Gonilha, J.A., “Comportamento estrutural de painéis sanduíche compósitos para aplicações na indústria da construção”, Revista da Associação Portuguesa de Análise Experimental de Tensões, Universidade Técnica de Lisboa, IST, ISSN 1646-7078

[6] Petrone, G., D’Alessandro, V., Franco F., Mace B., Rosa S., (2014), “Modal Characterisation of Recyclable Foam Sandwich Panels”, Composite Structures 113, 362-368

[7] Valente, António. OpenCell Technology (Tech & Applications Perspective).

[8] Arora, J. (2004). “Introduction to Optimum Design”, Second Edition, The University of Iowa, Elsevier, UK.

[9] Coutinho, K.D. (2006), “Método de Otimização Topológica em Estruturas Tridimensionais”. Tese de Mestrado em Engenharia Mecânica, Universidade Federal do Rio Grande do Norte.

[10] Cheng K.T. and Olhoff N., (1981), “An investigation concerning optimum design of solid elastic plates”, International Journal of Solids and Structures 17, 305-323.

[11] Bendsoe M.P., Kikuchi N., (1988), “Generating optimal topologies in structural design using a homogenization method”, Computer Methods in Applied Mechanics and Engineering 71, 197-224.

[12] Bendsoe M.P., (1989), “Optimal shape design as a material distribution problem”, Structural Optimization 1, 193-202.

[13] Zhou, M. and G.I.N. Rozvany (1991), “The COC algorithm - Part II. Topological, geometrical and generalized shape optimization”, Computer Methods in Applied Mechanics and Engineering 89, 309-336.

[14] Diaz A., Bendsoe M.P., (1992), “Shape optimization of structures for multiple loading situations using a homogenization method”, Structural Optimization 4, 17-22.

[15] Olhoff N., Thomsen J., Rasmussen J., (1993), “Topology optimization of bi-material structures”, In: Pedersen P, editor. Optimal Design with Advanced Materials. Amsterdam: Elsevier, p. 191-206.

[16] Suzuki K., Kikuchi H., (1991), Layout optimization using the homogenization method: generalized layout design of three-dimensional shells for car bodies. In: Rozvany G.I.N., editor. Optimization of large structural systems, p. 26-110 Nato-ASI, Lecture Notes.

[17] Soto C.A., Diaz A.R., (1993), “Optimum layout of plate structures using homogenization”, In: Bendsoe M.P., Mota Soares C.A., editors, Topology design of structures, Dordrecht: Kluwer, p. 387-407 Nato-ASI.

[18] Soto C.A., Diaz A.R., (1993), “A model for layout optimization of plate structures”, In: Pedersen P., editor, Optimal design with advanced materials, Amsterdam: Elsevier, p. 285-337.

[19] Bendsoe M.P., Mota Soares C.A., (1993) editors. “Topology design of structures”, Dordrecht: Kluwer.

[20] Rozvany G.I.N., Bendsoe M.P., Kirsch U., (Rev. 1995), “Layout optimization of structures”, Applied Mechanics Reviews 48, 41-119.

[21] Pedersen P., (1993), “Optimal design with advanced materials”, Elsevier, Amsterdam.

[22] Olhoff N, Rozvany G.I.N., (1995), editors. WCSMO-1 “Proceedings of First World Congress of Structural and Multidisciplinary Optimization”, Goslar, Germany. Oxford: Pergamon.

[23] Chung J. and Lee K., (1997), “Optimal design of rib structures using the topology optimization technique”, Proceedings of Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science 211, 425–437.

[24] Krog L.A. and Olhoff N., (1999), “Optimum topology and reinforcement design of disk and plate structures with multiple stiffness and eigenfrequency objectives”, Computers & Structures 72, 535–563.

[25] Stegmann, J. (2005). “Analysis and optimization of laminated composite shell structures”, PhD Thesis, Institute of Mechanical Engineering, Aalborg University.

[26] Lund, E. and Stegmann, J. (2005), “Discrete material optimization of general composite shell structures”, Topological Design Optimization of Strutures, Machines and Material: Status and Perspective, 89-98.

[27] Lund, E. and Stegmann, J. (2005), “On structural optimization of composite shell structures using a discrete constitutive parametrization”, Wind Energy 8, 109-124.

[28] Sigmund, O. and Torquato, S. (1997), “Design of material with extreme thermal expansion using a three-phase topology optimization method”, Journal of the Mechanics and Physics of Solids 45, 1037-1067.

[29] Ferreira, R. T. L. and Rodrigues, H.C, Guedes, J. M., Hernandes, J. A. (2013), “Hierarchical optimization of laminated fiber reinforced composites”, Preprint submitted to Composite Structures, January 23.

[30] Ferreira, R.T.L. and Hernandes, J.A. (2013), “New approximations for sequential optimization with discrete material interpolations”, in: Proceedings of the 10th World Congress on Structural and Multidisciplinary Optimization, May 19-24, Florida, USA.

[31] Vatanabe, S.L., Paulino, G.H., Silva, E.C.N. (2013), “Design of functionally graded piezocomposites using topology optimization and homogenization – Toward effective energy harvesting materials”, Computer Methods in Applied Mechanics and Engineering,

[32] Soares, Carlos A. M. (1982), “Elementos Finitos em Mecânica dos Sólidos”, departamento de engenharia mecânica, Instituto Superior Técnico

[33] Bendsoe, M.P. and O. Sigmund (2003), “Topology optimization - theory, methods and Applications”, Second Edition, Springer, USA.

[34] Gibiansky, L. V. and Sigmund, O. (2000), “Multiphase composites with extremal bulk modulus”, Journal of Mechanics and Physics of Solids 48, 461-498.

[35] Reddy, J. N. (2006), “An Introduction to Finite Element Method”, Third Edition, McGran-Hill, USA.

[36] Cristensen, P.W. and Klarbring, A. (2009), “An Introduction to Structural Optimization”, Springer, Canada.

[37] Bragança, Maria T. C. (2012), “Optimização da configuração de reforços numa classe de painéis planos considerando efeitos de plasticidade”, Tese Mestrado em Engenharia Mecânica, Instituto Superior Técnico, Universidade de Lisboa.

[38] Ferreira, Ana L.C. (2013), “Optimização de sequências de laminação para painéis planos reforçados por fibras”, Tese Mestrado em Engenharia Aeroespacial, Instituto Superior Técnico, Universidade de Lisboa.

[39] Silva, Edson B.L. (2009), “Optimização de Topologia de Reforços em Painéis Planos”, Tese Mestrado em Engenharia Mecânica, Instituto Superior Técnico, Universidade de Lisboa.

[40] Guilherme, Carlos E.M. (2006). “Otimização topológica e cálculo do gradiente de forma para estruturas submetidas à restrição de fadiga”, Tese Doutoramento em Engenharia Mecânica, Universidade Federal do Rio Grande do Sul.

[41] *http://www2.dbd.puc-rio.br/pergamum/tesesabertas/0310953_05_cap_03.pdf*. Acedido em: 29 de Abril de 2014

[42] *http://www.fing.edu.uy/~acanelas/publicaciones_files/P_Large.pdf*. Acedido em: 2 de Maio de 2014

[43] Coleman, T., Branch, M.A. and Grace, A., "Optimization toolbox for use with MATLAB", version 2, The Math Works.

[44] Herskovits, J. and Santos, G. (1998), "Feasible Arc Interior Point Algorithms for Nonlinear Optimization", in Fourth World Congress on Computational Mechanics, Buenos Aires.

[45] *http://www.mathworks.com/help/pdf_doc/optim/optim_tb.pdf*. Acedido em: 16 de Maio de 2014

[46] Cheng K.T. and Olhoff N., (1982), "Regularized formulation for optimum design of axisymmetric plates", International Journal of Solids and Structures 18, 153-169.

[47] Olhoff N., Lurie K.A., Cherkaev A.V., Fedorov A., (1981), "Sliding regimes of anisotropy in optimal design of vibrating axisymmetric plates", International Journal of Solids and Structures 17, 883-931.

[48] H.O. Hatem, Al-Taee, (2008), "Optimum Topological Design Of Stiffened Plates & Shells With Material Variables Under Unsymmetrical Lateral Loads", University of Babylon.

[49] Reddy, J. N. (2004), "Mechanics of Laminated Composite Plates and Shells: Theory and Analysis", Second Edition, CRC Press, USA.

[50] Almas, Miguel F.B.M. (2013), "Lamination Parameter Optimization Of Flat Fibre Reinforced Plates for Vibration Frequency Criteria", Tese Mestrado em Engenharia Mecânica, Instituto Superior Técnico, Universidade de Lisboa.

[51] Venkataraman, P. (2009), "Applied Optimization with MATLAB programming", second edition, John Wiley & Sons, Canada.

Anexo A

A.1 Exemplo de um ficheiro APDL que permite obter a energia, e os graus de liberdade de cada nó, a partir do software ANSYS

```
FINISH
/CLEAR,START
/PREP7
*AFUN,DEG
!EDDBL,1
j=1/4
PARRES,CHANGE,variaveis.txt,,
furo = 2                    !Tamanho do Furo
k=Ex1                       !Placa 1 Esquerda
l=Ex2                       !Placa 2
m=Ex3                       !Placa 3
n=Poissonxy
q = 200E9
y = 200E3

ET,1,SHELL181  ! Placa1
ET,2,SHELL181  ! Placa2
ET,3,SHELL181  ! Placa3
ET,4,SHELL181  ! Áreas que entram sempre
ET,5,SHELL181  ! Para as áreas não escolhidas

R,1,0.01,0.01,0.01,0.01,

MP,EX,1,k,
MP,PRXY,1,n
MP,DENS,1,1200
MP,EX,2,l,
MP,PRXY,2,n
MP,DENS,2,1200
MP,EX,3,m,
MP,PRXY,3,n
MP,DENS,3,1200
MP,EX,4,q,
MP,PRXY,4,n
MP,DENS,4,1200
MP,EX,5,y,
MP,PRXY,5,n
MP,DENS,5,1200

KEYOPT,1,1,0
KEYOPT,1,3,0
KEYOPT,1,8,1
KEYOPT,1,9,0
sectype,1,shell,,
secdata, 0.01,1,,3
secoffset,MID
seccontrol,,,, , , ,

KEYOPT,2,1,0
KEYOPT,2,3,0
KEYOPT,2,8,1
KEYOPT,2,9,0
```

```
sectype,2,shell,,
secdata, 0.01,2,,3
secoffset,MID
seccontrol,,,, , , ,

KEYOPT,3,1,0
KEYOPT,3,3,0
KEYOPT,3,8,1
KEYOPT,3,9,0
sectype,3,shell,,
secdata, 0.01,3,,3
secoffset,MID
seccontrol,,,, , , ,

KEYOPT,4,1,0
KEYOPT,4,3,0
KEYOPT,4,8,1
KEYOPT,4,9,0
sectype,4,shell,,
secdata, 0.01,4,,3
secoffset,MID
seccontrol,,,, , , ,

KEYOPT,5,1,0
KEYOPT,5,3,0
KEYOPT,5,8,1
KEYOPT,5,9,0
sectype,5,shell,,
secdata, 0.01,5,,3
secoffset,MID
seccontrol,,,, , , ,

!!!!!!!!!!!!!!! Keypoints !!!!!!!!!!!!!!!!!!!!!!!!!!!!
!PLACA
k,1,0,0,0
k,2,1,0,0
k,3,2,0,0
k,4,3,0,0
k,5,4,0,0
k,6,4,1,0
k,7,4,2,0
k,8,4,3,0
k,9,4,4,0
k,10,3,4,0
k,11,2,4,0
k,12,1,4,0
k,13,0,4,0
k,14,0,3,0
k,15,0,2,0
k,16,0,1,0
k,17,0,0,furo/2
k,18,1,0,furo/2
k,19,2,0,furo/2
k,20,3,0,furo/2
k,21,4,0,furo/2
k,22,4,1,furo/2
k,23,4,2,furo/2
k,24,4,3,furo/2
k,25,4,4,furo/2
k,26,3,4,furo/2
```

```
k,27,2,4,furo/2
k,28,1,4,furo/2
k,29,0,4,furo/2
k,30,0,3,furo/2
k,31,0,2,furo/2
k,32,0,1,furo/2
!Buraco
k,33,1,1,furo/2
k,34,2,1,furo/2
k,35,3,1,furo/2
k,36,3,2,furo/2
k,37,3,3,furo/2
k,38,2,3,furo/2
k,39,1,3,furo/2
k,40,1,2,furo/2
!Buraco baixo
k,41,1,1,0
k,42,2,1,0
k,43,3,1,0
k,44,3,2,0
k,45,3,3,0
k,46,2,3,0
k,47,1,3,0
k,48,1,2,0

!!!!!!!!!!!!!! Lines !!!!!!!!!!!!!!!!!!!!!!!!!!!
!Placas
L,1,2,j*4
L,2,3,j*4
L,3,4,j*4
L,4,5,j*4
L,5,6,j*4
L,6,7,j*4
L,7,8,j*4
L,8,9,j*4
L,9,10,j*4
L,10,11,j*4
L,11,12,j*4
L,12,13,j*4
L,13,14,j*4
L,14,15,j*4
L,15,16,j*4
L,16,1,j*4      !16
L,17,18,j*4
L,18,19,j*4
L,19,20,j*4
L,20,21,j*4
L,21,22,j*4
L,22,23,j*4
L,23,24,j*4
L,24,25,j*4
L,25,26,j*4
L,26,27,j*4
L,27,28,j*4
L,28,29,j*4
L,29,30,j*4
L,30,31,j*4
L,31,32,j*4
L,32,17,j*4       !31
!Buraco
```

```
L,33,34,j*4
L,34,35,j*4
L,35,36,j*4
L,36,37,j*4
L,37,38,j*4
L,38,39,j*4
L,39,40,j*4
L,40,33,j*4        !39
L,32,33,j*4
L,33,18,j*4
L,34,19,j*4
L,35,20,j*4
L,35,22,j*4
L,36,23,j*4
L,37,24,j*4
L,37,26,j*4
L,38,27,j*4
L,39,28,j*4
L,39,30,j*4
L,40,31,j*4       !51
!Baixo
L,41,42,j*4
L,42,43,j*4
L,43,44,j*4
L,44,45,j*4
L,45,46,j*4
L,46,47,j*4
L,47,48,j*4
L,48,41,j*4
L,16,41,j*4
L,41,2,j*4
L,42,3,j*4
L,43,4,j*4
L,43,6,j*4
L,44,7,j*4
L,45,8,j*4
L,45,10,j*4
L,46,11,j*4
L,47,12,j*4
L,47,14,j*4
L,48,15,j*4
!Apoios verticais
L,33,41,j*4
L,34,42,j*4
L,35,43,j*4
L,36,44,j*4
L,37,45,j*4
L,38,46,j*4
L,39,47,j*4
L,40,48,j*4
L,15,31,j*4
L,3,19,j*4
L,7,23,j*4
L,11,27,j*4

!!!!!!!!!!!!!!! Areas !!!!!!!!!!!!!!!!!!!!!!!!!!!
A,41,42,43,44,45,46,47,48
A,1,2,41,16
A,2,3,42,41
A,3,4,43,42
```

```
A,4,5,6,43
A,6,7,44,43
A,7,8,45,44
A,16,41,48,15
A,15,48,47,14
A,14,47,12,13
A,47,46,11,12
A,46,45,10,11
A,45,8,9,10            !13
A,17,18,33,32
A,18,19,34,33
A,19,20,35,34
A,20,21,22,35
A,22,23,36,35
A,23,24,37,36
A,24,25,26,37
A,26,27,38,37
A,27,28,39,38
A,28,29,30,39
A,30,31,40,39
A,31,32,33,40          !25
!Apoios Verticais com dobragem a 90graus
A,31,40,48,15   !Esquerda
A,36,23,7,44    !Direita
A,27,38,46,11  !Frente
A,34,19,3,42    !Trás               29
!Apoios Verticais sem dobragem (direita)
A,39,40,48,47
A,36,35,43,44
A,38,37,45,46
A,33,34,42,41   !                  33
!Apoios Verticais sem dobragem (esquerda)
A,40,33,41,48
A,37,36,44,45
A,39,38,46,47
A,34,35,43,42      !             37

!!!!!!!!!!!!!! Caso 1 - malha
ASEL,S,AREA,,1,25
AATT,4,,4,0,4
ASEL,S,AREA,,26,27
AATT,3,,3,0,3
ASEL,S,AREA,,30,31
AATT,1,,1,0,1
ASEL,S,AREA,,34,35
AATT,2,,2,0,2
ASEL,S,AREA,,28,29
AATT,5,,5,0,5
ASEL,S,AREA,,32,33
AATT,5,,5,0,5
ASEL,S,AREA,,36,37
AATT,5,,5,0,5
ALLSEL,ALL
amesh,all

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Criar sucessivas repetições da placa !!!!!!!!!!!!!!!!!!!
!AGEN,2 , ALL, , , 4, 0, 0,0, ,
!AGEN,2 , ALL, , , 0, 4, 0,0, ,
```

```
!ALLSEL,ALL
!NUMMRG,NODE

!!!!!!!!!!!!!!!!Constrangimentos para 1 buraco!!!!!!!!!!!!!!!!!!!!!!
LSEL,S,LOC,z,1
LSEL,R,LOC,y,0
DL,ALL,,ALL,0
ALLSEL,ALL
LSEL,S,LOC,z,1
LSEL,R,LOC,y,4
DL,ALL,,ALL,0
ALLSEL,ALL
LSEL,S,LOC,z,0
LSEL,R,LOC,y,0
DL,ALL,,ALL,0
ALLSEL,ALL
LSEL,S,LOC,z,0
LSEL,R,LOC,y,4
DL,ALL,,ALL,0
ALLSEL,ALL
ASEL,S,AREA,,1,13
SFA,ALL,1,PRES,150
ALLSEL,ALL

!!!!!!!!!!!!!!!!!!!!!!!!!!!! Solução !!!!!!!!!!!!!!!!!!!!!!!!!!!!
FINISH
/SOL
/STATUS,SOLU
SOLVE

!!!!!!!!!!!!!!!!!!!!!!!!!!!! Saca Energia !!!!!!!!!!!!!!!!!!!!!!
/POST1
*CREATE,com
*CFOPEN,  'C:\Users\Hugo\Desktop\Tese\Programa  Hugo  Pina_FMINCOM\Programa Hugo Pina _ 3variaveisFMINCOM\Energia','txt',
!Cria uma tabela com a Energia
!Soma todos os valores da tabela para obter a função objectivo final
ETABLE,Energia,SENE
SSUM
*GET,EnergiaST,SSUM,,ITEM,Energia
*VWRITE,EnergiaST
(E30.15)
*CFCLOS
*END
/INPUT,com

!!!!!!!!!!!!!!!!!!!!!!!!!!!! Saca os DOF (Ue) !!!!!!!!!!!!!!!!!!!!!!!!!!!!
cm,nodes,node                    ! Groups geometry items into a component.
                                   ! 'CM, Cname, Entity'
*get,total_node,node,,count       ! total_node = Nº de nós total
*DIM,force,,total_node,7          !creation of an total_node x 4 array
            !*DIM, Par, Type, NºLinhas, NºColunas, KMAX, Var1, Var2, Var3, CSYSID
           !Defines an array parameter and its dimensions.



*do,i,1,total_node               !loop where each element table result and element number is located in each row of the array
                                 !*DO, Par, IVAL, FVAL, INC
                                 !Defines the beginning of a do-loop.
```

```
  cmsel,s,nodes                    !Selects a subset of components and assemblies.
  *get,min_node,node,,num,min      !min_node = Nó mínimo
  *get,force(i,1),node,,num,min
  *get,force(i,2),node,min_node,u,x
  *get,force(i,3),node,min_node,u,y
  *get,force(i,4),node,min_node,u,z
  *get,force(i,5),node,min_node,rot,x
  *get,force(i,6),node,min_node,rot,y
  *get,force(i,7),node,min_node,rot,z
  nsel,s,node,,min_node
  esln
  cmsel,s,nodes
  nsel,u,node,,min_node
  cm,nodes,node
*enddo
alls
/POST1
*CREATE,com
*CFOPEN,'C:\Users\Hugo\Desktop\Tese\Programa Hugo Pina_FMINCOM\Programa Hugo Pina _ 3variaveisFMINCOM\DOF_results','txt'          !Save of the results file
*VWRITE,force(1,1),force(1,2),force(1,3),force(1,4),force(1,5),force(1,6),force(1,7)
(F10.2,2X,E10.3,2X,E10.3,2X,E10.3,2X,E10.3,2X,E10.3,2X,E10.3)
*CFCLOSE
alls
*END
/INPUT,com

!!!!!!!!!!!!!!!!!!!!!!!!!!!! Saca o Nº dos nós de cada elemento !!!!!!!!!!!!!!!!!!!!!!!!!!!!
cm,elements,elem                   ! Groups geometry items into a component.
                                   ! 'CM, Cname, Entity'
*get,total_elem,elem,,count        ! total_elem = Nº de elementos total
*DIM,numerono,,total_elem,5        !creation of an total_node x 7 array
          !*DIM, Par, Type, NºLinhas, NºColunas, KMAX, Var1, Var2, Var3, CSYSID
           !Defines an array parameter and its dimensions.
*do,i,1,total_elem                 !loop where each element table result and element number is located in each row of the array
                   !*DO, Par, IVAL, FVAL, INC
                   !Defines the beginning of a do-loop.
  cmsel,s,elements                 !Selects a subset of components and assemblies.
  *get,min_elem,elem,,num,min      !min_elem = Elemento mínimo
  *get,numerono(i,1),elem,,num,min
  *get,numerono(i,2),elem,min_elem,node,1
  *get,numerono(i,3),elem,min_elem,node,2
  *get,numerono(i,4),elem,min_elem,node,3
  *get,numerono(i,5),elem,min_elem,node,4
  esel,s,elements,,min_elem
  esln
  cmsel,s,elements
  esel,u,elem,,min_elem
  cm,elements,elem
*enddo
alls
/POST1
*CREATE,com
*CFOPEN,'C:\Users\Hugo\Desktop\Tese\Programa Hugo Pina_FMINCOM\Programa Hugo Pina _ 3variaveisFMINCOM\Nos_por_Elemento','txt'          !Tem que ir para a pasta do programa MatLab
*VWRITE,numerono(1,1),numerono(1,2),numerono(1,3),numerono(1,4),numerono(1,5)
(F10.2,2X,F10.2,2X,F10.2,2X,F10.2,2X,F10.2)
*CFCLOSE
```

```
alls
*END
/INPUT,com
```

A.2 Exemplo de um ficheiro APDL que permite obter as matrizes de rigidez de cada elemento, a partir do software ANSYS

```
! O código é idêntico ao apresentado em A.1 até à linha - "!!!! Solução!!!!"
! A partir dessa linha deve escrever-se
!!!!!!!!!!!!!!!!!!!!!!!!!!!! Solução !!!!!!!!!!!!!!!!!!!!!!!!!!!!
/SOLU
ANTYPE,static
wrfull,1
ematwrite,yes
SOLVE
FINISH
/aux2
FILE,file,full
hbmat,MatrizKglobal,txt,,ASCII,STIFF,YES,
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
/output,file,lis,,
FORM,LONG
FILEAUX2,'file','full',' '
DUMP,ALL

*DMAT,BAD_K,D,IMPORT,EMAT,file.emat,STIFF,38      ! onde 38 é o Nº do elemento
DIM = BAD_K_ROWDIM                                ! Nb of ROWS
! COPY ALL THE LOWER TRIANGULAR INTO A VECTOR
*VEC,V,D,ALLOC,(DIM*(DIM+1)/2)
II=1
*DO,J,1,DIM,1
  *DO,I,J,DIM,1
    V(II) = BAD_K(I,J)
    II = II+1
  *ENDDO
*ENDDO
*FREE,BAD_K
! ALLOCATE A NEW DENSE K MATRIX, WE COPY BACK THE VALUES IN THE GOOD WAY

*DMAT,K,D,ALLOC,DIM,DIM
II=1
*DO,J,1,DIM,1
  *DO,I,1,J,1
    K(I,J) = V(II)
    K(J,I) = V(II)
    II=II+1
  *ENDDO
*ENDDO
*FREE,V        ! FREE THE TMP VECTOR
*PRINT,K,K.txt       ! CONTAINS THE CORRECT ELEMTARY K MATRIX
```

A.3 Exemplo da derivada da função de peso

De seguida é apresentado o caso da derivada de w_1, para o caso de três candidatos, em ordem às variáveis de projecto, utilizando o esquema 4 do DMO.

$$\frac{dw_1}{dx_1} = p.x_1^{p-1}.(1-x_2^p)(1-x_3^p)$$

$$\frac{dw_1}{dx_2} = -p.x_1^p.x_2^{p-1}(1-x_3^p)$$

$$\frac{dw_1}{dx_3} = -p.x_1^p.x_3^{p-1}(1-x_2^p)$$

O raciocínio é idêntico para cada uma das funções de peso e esquemas do método DMO restantes.